Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems ⋆

Ryuichiro Higashinaka *
and Katsuhito Sudoh and Mikio Nakano [1]

NTT Comunication Science Laboratories, Nippon Telegraph and Telephone Corporation, 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan

Abstract

This paper proposes a method for the confidence scoring of intention recognition results in spoken dialogue systems. To achieve tasks, a spoken dialogue system has to recognize user intentions. However, because of speech recognition errors and ambiguity in user utterances, it sometimes has difficulty recognizing them correctly. Confidence scoring allows errors to be detected in intention recognition results and has proved useful for dialogue management. Conventional methods use the features obtained from the speech recognition/understanding results for single utterances for confidence scoring. However, this may be insufficient since the intention recognition result is a result of discourse processing. We propose incorporating discourse features for a more accurate confidence scoring of intention recognition results. Experimental results show that incorporating discourse features significantly improves the confidence scoring.

Key words: confidence scoring, speech understanding, discourse understanding, spoken dialogue systems

⋆ This paper is an extended version of our earlier report presented at 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (Higashinaka et al., 2005).

* Corresponding author. Tel.: +81-774-93-5256; fax: +81-774-93-5385.
Email addresses: rh@cslab.kecl.ntt.co.jp (Ryuichiro Higashinaka), sudoh@cslab.kecl.ntt.co.jp (Katsuhito Sudoh), nakano@jp.honda-ri.com (Mikio Nakano).

[1] Currently with Honda Research Institute Japan, 8-1, Honcho, Wako-shi, Saitama 351-0114, Japan.

1 Introduction

Spoken dialogue systems are expected to usher in new possibilities in human-computer interactions in the near future. For a spoken dialogue system to achieve certain tasks while conversing with users, the system has to correctly recognize user intentions. Here, we use the term *user intention* to express the information that the user has in mind and has to convey to the system in order to achieve his/her goal, such as extracting some particular information from the system.

Since users do not always convey their intentions in one utterance and speech recognition errors might occur, the system and the user normally have to exchange several utterances before the system finally recognizes the user's true intention. This paper addresses this interactive intention recognition process, focusing on the types of tasks in which intention recognition results are represented by *frames* or *slot-value pairs* (Bobrow et al., 1977; Goddeau et al., 1996). We assume that the slots are filled with words or concepts, sometimes referred to as slot-fillers, in user utterances, which is common in frame-based applications.

In such interactive intention recognition, after each user utterance, the system updates the intention recognition result, based on which the system performs dialogue management; namely, it decides what response it should make. Recently, *confidence scoring*, a technique for assigning reliability scores to speech recognition results, has been applied to detect errors in intention recognition results and has proved useful for dialogue management (Komatani and Kawahara, 2000; Singh et al., 2002; Dohsaka et al., 2003). If the detection is successful, the system can safely avoid unnecessary confirmations for reliable slots and ask questions about unreliable or unfilled ones preferentially.

In current confidence scoring for intention recognition results, since words/concepts in user utterances fill the slots, the confidence of words/concepts, which is typically calculated using various features obtained from speech recognition results and speech understanding results for single utterances, is used for the confidence of slot values. However, this may be inappropriate because slot values are the results of discourse understanding, not the results of single utterance understanding. Consider a case where a slot is filled with a value that has once been denied or corrected by the user in a dialogue. The confidence of that value is likely to be lower than can be calculated for the word/concept in the utterance.

This paper addresses this problem and proposes incorporating discourse features into the confidence scoring of intention recognition results. In our approach, we introduce a number of discourse-related features (called *discourse*

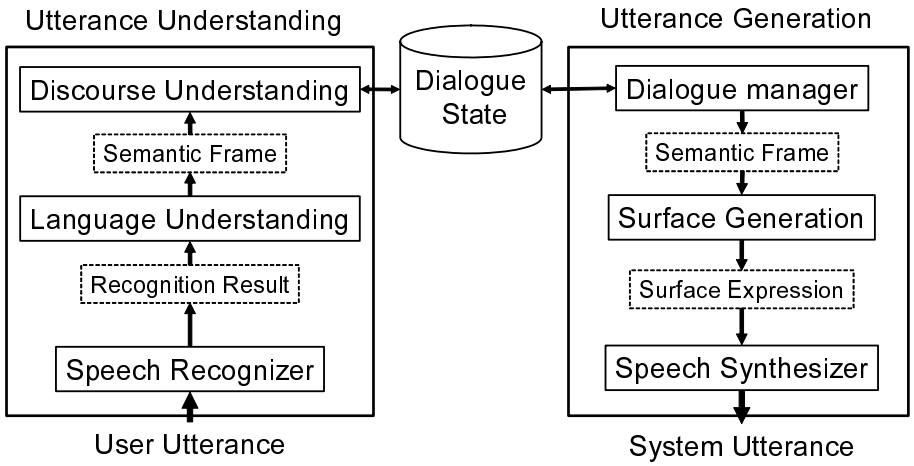

Fig. 1. Architecture of a spoken dialogue system.

features) that characterize the contextual adequacy of slot values in terms of Grice's maxims of cooperativeness, and use them together along with the features obtained from speech recognition results to train *confidence models* that classify slot values as correct or incorrect based on both the context and the speech recognition/understanding results. Since the features are only available for filled slots, we only deal with slots that have values in this paper.

Although this work does not aim at improving discourse understanding of spoken dialogue systems directly, we are hoping to obtain useful ideas for improving our speech understanding component through the process of confidence model training and the analysis of confidence models.

In the next section, we briefly outline the intention recognition process in spoken dialogue systems. In Section 3, we explain the need for the confidence scoring of intention recognition results. In Section 4, we introduce conventional methods and follow that with a detailed description of our proposed method and the discourse features in Section 5. In Section 6, we describe the experiments performed to verify the proposed method. In the last section, we summarize the paper and mention future work.

2 Intention recognition in spoken dialogue systems

The basic architecture of a spoken dialogue system is illustrated in Figure 1. When receiving a user utterance, the system behaves as follows.

(1) The speech recognizer receives a user utterance and outputs a speech recognition result, such as an N-best list and a word graph.
(2) The language understanding component receives the speech recognition

result. Syntactic and semantic analyses are performed to convert it into a meaning representation, often called a semantic frame or sometimes a logical form. A semantic frame is typically composed of a *dialogue act* that identifies the main intent of the user's utterance, augmented with necessary ancillary information often encoded as attribute-value pairs (also called concepts), or encoded by using a predicate calculus terminology in the case of logical forms. Speech recognition and language understanding are collectively referred to as *speech understanding* in this paper.

(3) The discourse understanding component receives the semantic frame, refers to the current dialogue state, and updates the *dialogue state*. A dialogue state is a collection of bits of information that the system internally stores, which includes the intention recognition result, the user utterance history, the system utterance history, and so forth. The intention recognition result is updated to suit the user's true intention, taking all the previous exchanges of utterances into account. The semantic frame corresponding to the user utterance is added to the user utterance history at the same time.

(4) The dialogue manager refers to the updated dialogue state, decides the next utterance, and outputs the next content to be delivered as a semantic frame. The dialogue state is updated at the same time so that it contains the content of system utterances.

(5) The surface generation component receives the semantic frame and produces the surface expression, namely, the next words to be spoken.

(6) The speech synthesizer receives the next words to be spoken and responds to the user by speech.

This paper concerns the intention recognition result in the dialogue state. The intention recognition result is considered to be the most important feature of the dialogue state because it reflects all previous exchanges of utterances between the user and the system.

In this paper, we assume that the intention recognition results are represented simply by frame expressions that consist of slot-value pairs (Bobrow et al., 1977; Goddeau et al., 1996), and that words in speech recognition hypotheses or concepts in language understanding results fill the slots, since filling slots with relevant words/concepts can be considered the most basic way of understanding user utterances and is the practice in many practical applications.

Plan-based systems use plan trees or logical forms to represent the user intention (Allen et al., 2001; Rich et al., 2001). However, considering the complexity of tasks that are currently used in applicable systems and the performance of speech recognizers, frame-based intention recognition is sufficient (Chu-Carroll, 2000; Seneff, 2002).

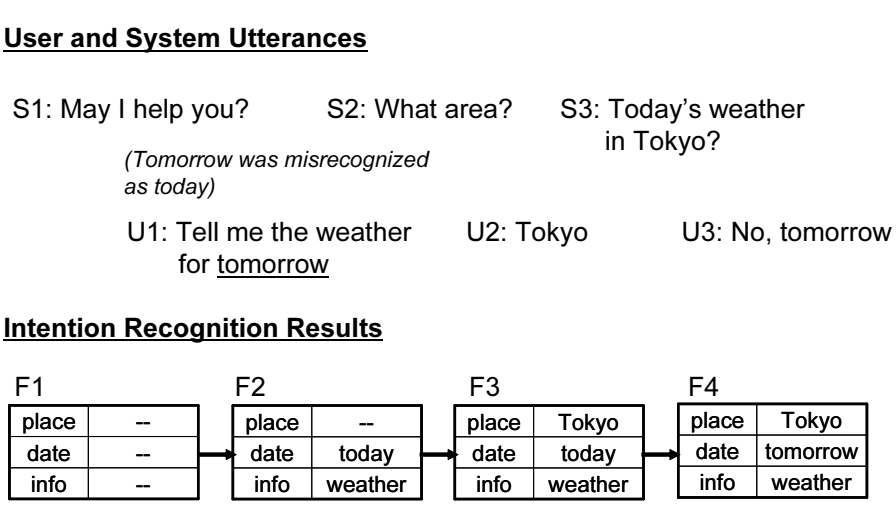

Fig. 2. Updating an intention recognition result. (S, U and F indicate a system utterance, a user utterance, and a frame, respectively.)

We also assume that the system responses are generated using words/concepts in the slots. To formulate, we model the interaction between the user and the system as follows: (1) The user sends words/concepts or sometimes commands to the system to change the slots. Here, commands represent user utterances that do not contain words/concepts, such as acknowledgments and denials. They have certain effects on the slots, such as the deletion of slot values. (2) The system responds to the user using the words/concepts stored in the slots. Then (1) and (2) are repeated until the user is satisfied with the system response. We believe that this model is general enough to encapsulate most of the task-oriented spoken dialogue systems in use today.

Figure 2 shows how the intention recognition result is updated in the course of a dialogue in a weather information system. In the example, "tomorrow" was misrecognized as "today" by the speech recognizer (U1), causing the system to have an incorrect value for the date slot (F2). The misunderstood item was later corrected by the user (U3), who noticed the error in the intention recognition result because of the system's incorrect confirmation request (S3). Through the interactive process with the user, the intention recognition results get closer to the correct user intention (F1-F4).

Since the understanding of user utterances and the production of system utterances are both affected by the content of the intention recognition result, correctly recognizing user intentions is crucial to carrying out the desired tasks.

3 Need for confidence scoring of intention recognition results

There has been a tremendous amount of research in the field of speech understanding in spoken dialogue systems. To cope with speech recognition errors and ungrammatical utterances in unconstrained speech, the keyword spotting method (Foote et al., 1997), which extracts only the words relevant to particular applications, and robust parsing techniques, such as island-driven parsing and partial parsing, which yield semantically important islands of words rather than a full parse, have been extensively studied (Corazza et al., 1991; Seneff, 1992; Baggia and Rullent, 1993).

Similarly, statistical classification techniques have been used to detect relevant pieces of words in utterances (Kuhn and Mori, 1995; Huang et al., 2001; Béchet et al., 2004), and statistical machine translation methods, which regard the problem of speech understanding as translating an utterance into a set of concepts, have also been gaining popularity (Macherey et al., 2001). To enhance classification accuracy, the use of various knowledge sources, such as plan trees and prosodic information, has also been considered (Abdou and Scordilis, 2001).

In the realm of discourse understanding, which works on top of speech understanding, there is also a growing body of research. Filisko (2002) proposed a context resolution server that specializes in reference resolution and ambiguity resolution in speech understanding results. Miyazaki et al. (to appear) and Higashinaka et al. (2003b) both employ a multi-world model, in which multiple discourse understanding results are maintained as an ordered list to enable discourse-level ambiguity to be retained and resolved by succeeding utterances. The difference between the two models is that the former uses hand-crafted rules, and the latter uses statistical information derived from dialogue corpora for the ranking.

Although much work has been done in speech understanding and discourse understanding, it is still acknowledged that speech recognition errors are inevitable, and that speech recognition errors often cause a system to misunderstand the user's intention. In addition, ambiguities in natural language also make it difficult for a systems to correctly understand the user's true intention. Therefore, the dialogue manager has to confront the problem of handling unreliable and ambiguous intention recognition results.

Since the slot values are unreliable, one safe and simple approach for dialogue management is to confirm every item in the slots until all items in them are acknowledged by the user. However, too many confirmations are likely to make dialogues tedious, and when the system reduces the number of confirmations, the system is likely to deliver undesired information based on incorrectly rec-

ognized items. The system needs to find a balance between too many and too few confirmations. For this purpose, the system has to be able to detect exactly what item needs to be confirmed.

In speech recognition research, a technique called *confidence scoring* has been increasingly used to detect errors in speech recognition results. For example, it has been used for utterance verification (Rahim et al., 1997). It also helps transcribers find erroneous words/phrases in the recognized sentences, which speeds up the transcription process (Endo et al., 2002). Recently, this technique has also been applied to detect errors in intention recognition results and has proved useful for dialogue management.

Komatani and Kawahara (2000) and Dohsaka et al. (2003) used the confidence of the intention recognition results to adaptively change dialogue strategies, which enables the system to confirm only the necessary items and avoid unnecessary confirmations. In this way, the task completion time was considerably reduced. The confidence of intention recognition results has also been used in order to better characterize the status of a dialogue state (called a state space) for the automatic learning of optimal dialogue management policies with reinforcement learning techniques (Singh et al., 2002).

Since estimating the reliability of the intention recognition results allows the dialogue manager to have a wider variety of choices as to how to respond to the user and enables the system to characterize the current state of a dialogue more accurately, there is a strong need for the confidence scoring of intention recognition results.

4 Conventional methods

Slots are typically filled with words in speech recognition hypotheses or with concepts in speech understanding results, and the acoustic, linguistic, or sometimes the semantic reliability of the words or concepts has been used for the confidence of the slots. Figure 3 shows an example of the confidence scoring of the slots in Fig. 2, illustrating how the confidence of words $c_1 \dots c_9$ can be associated with the slots.

The simplest way to calculate a confidence score is to use the score that the speech recognizer outputs for words, e.g., the total acoustic and language model score or the word posterior probability (Wessel et al., 2001). When a slot is filled by a concept, the total or mean confidence of the words that form that concept is normally utilized. For example, an utterance "to Tokyo" might form a concept "arrival-city=Tokyo." In this case, the confidence for this concept is calculated taking the summation or the mean of the confidence

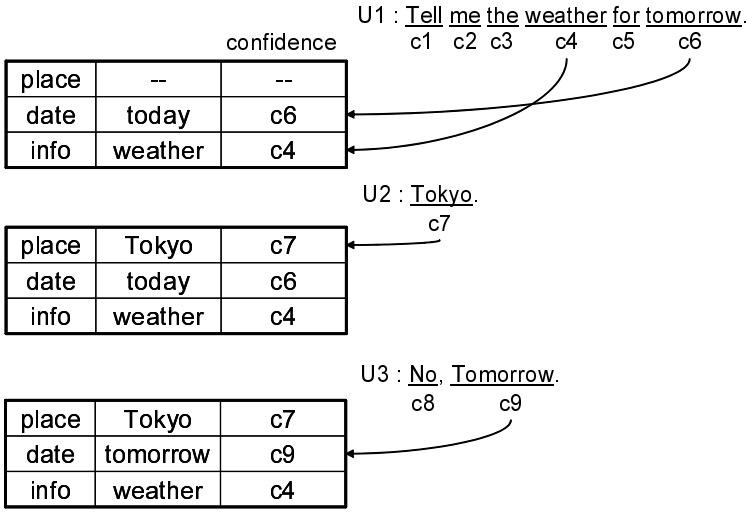

Fig. 3. Conventional methods. Word confidences $c_1 \ldots c_9$ are associated with the slots.

of words "to" and "Tokyo."

To obtain more accurate scores especially for concepts, probabilistic modeling of a sequence of concepts has been proposed. Hacioglu and Ward (2002), using concept N-grams, proposed constructing a concept graph on top of a word graph and calculating confidence of concepts in a fashion similar to calculating the word posterior probability. Lin and Wang (2001) propose a concept-based probabilistic verification model, which also exploits concept N-grams.

There are also approaches that use *confidence models* for confidence scoring. A confidence model is a kind of a classifier that scores or classifies words/concepts based on training data. Although data collection, feature extraction, and labeling procedures have to be performed before the training, the confidence model approach has proved particularly useful when various types of features, such as numeric values and symbolic values, have to be integrated for the scoring.

Hazen et al. (2002) integrate two levels of features in speech recognition hypotheses to train confidence models for words: word-level features that focus only on the reliability of the acoustic samples corresponding to the word, and utterance-level features that concern the appropriateness of the whole utterance in which the word is found. This integration is based on the assumption that if the whole utterance is unreliable, the word contained in that utterance is also likely to be incorrect.

In order to achieve more accurate scoring depending on the context, Pradhan and Ward (2002) proposed creating confidence models for concepts (or semantic frames) using previous system prompts in addition to the features

obtained from the speech recognition results. They adopted this particular approach because they focus on large vocabulary-based system-guided dialogues within the DARPA Communicator project (Pellom et al., 2000), in which user utterances are heavily influenced by previous system prompts.

There is also an approach that utilizes *pragmatic analyses* to score concepts uttered by the user (Ammicht et al., 2001). This makes several basic assumptions about what concepts the user should utter after a system response and uses the assumptions as rules to score the incoming concepts and rescore already recognized concepts. For example, when an already recognized concept seems to have been implicitly confirmed, the confidence of that concept is augmented.

Hirschberg et al. (2004) introduce a number of prosodic features, such as F0, the length of a pause preceding the turn, and the speaking rate, to detect misrecognized user turns in spoken dialogue corpora. Since users tend to change the way they speak when faced with inappropriate system utterances that originate from previous misrecognized utterances, they use the prosodic features of subsequent utterances to detect possible errors in previous user utterances. The problem they are dealing with is different from ours in that they do not evaluate user utterances immediately after speech recognition.

5 Proposed method

Previous methods have been quite successful in providing reasonably good estimates of correct/incorrect for intention recognition results. However, we would like to pose a question: *Is it really appropriate to use the confidence of words/concepts for the confidence of intention recognition results?*

We argue that it may not be appropriate because the confidence of words/concepts is calculated separately from the context; that is, the intention recognition result is the system's understanding result of a discourse, not the result of understanding an independent utterance. There may be some cases where hypothesized words/concepts are not likely to fill the slots, as when the slot values contradict what has been said in a prior part of a dialogue. Ignoring the fact that intention recognition results represent the discourse may lead to inaccurate confidence scoring. Therefore, we propose incorporating discourse information into the training of confidence models.

To enable discourse information to be used in confidence model training, we have to find features to represent a slot value from the discourse point of view. We hypothesize that there is a principle that a valid discourse should satisfy and that any indication of violation of or conformity to this principle can be

used to score a slot value in a discourse. We employ, as such a principle, Grice's maxims of cooperativeness (Grice, 1975). Grice's maxims are described as norms that should be followed in a collaborative conversation. Grice proposed four maxims, namely, maxims of Quantity, Quality, Relation, and Manner. Figure 4 shows the description of the maxims from (Grice, 1975). We created twelve discourse features, each one of them indicating possible violation of or conformity to the maxims. The derivation of the features are described in Section 5.1 in detail. Although there may be other principles or models for discourse, such as discourse plans (Allen et al., 2001; Rich et al., 2001), such high-level discourse principles may not be necessary when considering the speech recognition errors. Therefore, we only consider Grice's maxims in this paper.

Along with the discourse features, we also use acoustic and language model features of the words/concepts filling the slots because they have been proven useful in the literature. Having defined the features, we take the following steps in confidence model training: We (1) collect slot value samples through dialogue experiments with human users, (2) extract the discourse features and the acoustic and language model features for slot values and annotate them as correct/incorrect, and (3) train confidence models for slot values using the collected data.

As a confidence model training technique, we adopt one of the existing techniques (Hazen et al., 2002). For evaluation, we compare the performance of the obtained confidence models with that of the baseline models. The baseline here means models that only use acoustic and language models for the confidence model training. We also compare our models, for reference, with a method that only uses the posterior probability of words that the speech recognizer outputs, since posterior probability is widely used in the community for its simplicity.

Although the use of previous system prompts can be seen as incorporating discourse information into confidence scoring (Pradhan and Ward, 2002), our approach is different in that we deal with the discourse understanding result, not the result of single utterance understanding, and in that our discourse features are represented by numeric values, not symbolic conditions for classifying user utterances. In addition, compared to the tasks in the Communicator project (Pellom et al., 2000), we focus on relatively smaller tasks with less system initiative and handle restricted utterances mainly consisting of user requests. Therefore, the use of previous system prompts is not expected to greatly improve confidence scoring in our case. However, if we had to handle a wider variety of utterances, our approach could be used together with the work of Pradhan and Ward (2002).

We also see (Ammicht et al., 2001) as an attempt to incorporate discourse

(1) **Maxim of Quantity:**
  (a) Make your contribution as informative as is required (for the current purposes of the exchange).
  (b) Do not make your contribution more informative than is required.
(2) **Maxim of Quality:**
  (a) Do not say what you believe to be false.
  (b) Do not say that for which you lack adequate evidence.
(3) **Maxim of Relation:**
  (a) Be relevant.
(4) **Maxim of Manner:**
  (a) Avoid obscurity of expression.
  (b) Avoid ambiguity.
  (c) Be brief (avoid unnecessary prolixity).
  (d) Be orderly.

Fig. 4. Grice's maxims of cooperativeness (Grice, 1975).

information into the confidence scoring. However, they are also not particularly focusing on discourse understanding results but concepts in single utterances and their approach uses heuristic rules for the scoring, directly relating certain discourse phenomena with fixed effects, whereas our approach aims at finding useful features to express discourse information so that the features can be related to confidence scores by confidence model training based on training data.

5.1 Discourse features

Here, we describe how we derive our discourse features. In all, we came up with 12 discourse features: one, seven, and four features in relation to the maxim of quantity, quality, and manner, respectively. Since we consider that the maxim of relation is automatically abided by in task-oriented dialogues–for example, in the weather information domain, the user and the system would not talk about booking flights or train tickets–we only focused on the remaining three maxims.

The discourse features are conceived following our assumption about the interaction between the user and the system; namely, the user sends words/concepts or sometimes commands to the system in order to change the slots, and the system responds to the user using the words/concepts stored in the slots. We argue that as long as the system follows this assumption, our features can be safely extracted. We also assume that the user's true intention does not change during the dialogue. In what follows, we describe in detail each feature

related to the maxims.

5.1.1 Features related to the maxim of quantity

The maxim of quantity suggests that one has to make one's contribution to the conversation as informative as necessary. The mention of a slot value that is the same as the one appearing in the previous system's confirmation request may not, therefore, be desirable. For example, the exchange

System : "Are you interested in the weather in Tokyo?"

User : "The weather in Tokyo"

corresponds to a case violating the maxim of quantity. Although the sequence may be a re-confirmation of the system's confirmation request, in terms of the maxim of quantity it is better for the user to provide more information about his/her intentions. Taking this into account, we conceived the following discourse feature D1:

(D1) **Same keyword pair count:** Throughout the dialogue, count the number of times the system confirms the current slot value and the user mentions the same value in the next utterance. We use this count as the feature. A large value of this feature would mean that there have been a lot of un-informative interactions about a particular slot value, suggesting that the value may be wrong.

5.1.2 Features related to the maxim of quality

The maxim of quality states that one should not say what one believes to be false. This can be interpreted as: the content of all the user utterances should be consistent. Therefore, any contradiction or inappropriateness among the system's recognized user intentions can be used as an indicator of a violation of the maxim of quality.

To describe how the intention recognition results (slot values) are recognized in the course of a single dialogue, we first introduce the idea of the *slot value sequence*, which represents the transition of values of a particular slot. For example, $\{null \rightarrow null \rightarrow Tokyo \rightarrow Tokyo\}$ is a slot value sequence for the place slot in F4 in Fig. 2. Here, the last value Tokyo is the current value whose confidence we aim to estimate, and *null* means that the slot does not have a value. Ideally, if the user is following the maxim, the slot value sequence should consist of just one single value. By characterizing the slot value sequence from different points of views, we conceived the following seven discourse features (D2 through D8):

(D2) **Slot purity:** In the slot value sequence, count the number of times the current value is found and divide that count by the number of non-null values in the sequence. We use this ratio as the feature. For example, when the value of the place slot changes {*Tokyo* → *Osaka* → *Kyoto* → *Osaka*}, then the current value Osaka is found in two of the four values, making the slot purity 1/2. This feature encodes the user's consistency about a certain value. Therefore, a large value of this feature may suggest that the slot value is correct.

(D3) **Top slot purity:** In the slot value sequence, for all the values that appear, count the number of times each value appears, find the highest count, and divide that count by the number of non-null values in the sequence. We use this ratio as the feature. When the value for the place slot changes {*Tokyo* → *Osaka* → *Kyoto* → *Osaka*}, Tokyo, Osaka, and Kyoto are assigned the values of 1/4, 1/2 (2/4) and 1/4, respectively. The maximum value is Osaka's 1/2; therefore, the top slot purity is 1/2. This feature represents the slot purity of the dominating slot value in the sequence if there is any. If the top slot purity of a slot value is greater than its slot purity, it may be likely that the slot value is wrong.

(D4) **Slot variety:** The number of different values that appear in the slot value sequence. For {*Tokyo* → *Osaka* → *Kyoto* → *Osaka*}, there are three values Tokyo, Osaka, and Kyoto; therefore, the slot variety is 3. This feature encodes the user's inconsistency, and a large value of this feature may suggest that the slot value is wrong.

(D5) **Deny count:** The number of times the current value has been deleted. For example, consider the sequence {*Tokyo* → *null* → *Kyoto* → *Tokyo*}. The current value Tokyo is once denied (set to null) by the user (later set to Kyoto). Therefore, the value is 1. If a certain value is correct, a cooperative user would not delete that value. A large value of this feature may suggest that the slot value is wrong.

(D6) **Overwrite count:** The number of times the current value has been overwritten by other values. For example, consider the sequence {*Tokyo* → *Osaka* → *Kyoto* → *Tokyo*}. The current value Tokyo is overwritten once by Osaka. Therefore, the value is 1. If a certain value is correct, a cooperative user would not overwrite/replace that value. Therefore, a large value of this feature may suggest that the slot value is wrong.

(D7) **Continue count:** Starting backwards from the current value, count the number of times the current value is found in the slot value sequence *successively*. We use this count as the feature. For example, consider the sequence {*null* → *Tokyo* → *Tokyo* → *Tokyo*}. Before the current value Tokyo, there are two Tokyo values. Therefore, the value is 2. Since the slot values have to be successively the same to yield a large value, this feature encodes the user's possible strong consistency about a certain value. Therefore, a large value of this feature may strongly suggest that the slot value is correct.

(D8) **Different value count:** Starting backwards from the current value,

count the number of times the current value is *not* found in the slot value sequence *successively*. We use this count as the feature. For example, consider the sequence $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Tokyo\}$. There are two non-Tokyo values before the current value Tokyo. Therefore, the value is 2. This feature functions exactly opposite to the continue count (D7), as it encodes the user's possible strong inconsistency. A large value of this feature may suggest that the slot value is wrong.

5.1.3 Features related to the maxim of manner

The maxim of manner states that one should avoid unnecessary prolixity as well as ambiguity. Therefore, if there are a large number of same/different words/concepts corresponding to a slot value appearing in user or system utterances, it may be an indication that the slot value is wrong. Note that these features focus on the user's and system's raw utterances or dialogue acts with concepts, not the slot value sequence and that these features encode what the system has observed within a dialogue rather than what the system has understood. Taking this into account, we enumerated the following four features (D9 through D12):

(D9) **Same keyword count in user utterances:** The number of times a concept corresponding to the current value appears in the previous user utterances. For example, when the current value is Tokyo, we count the number of times the word "Tokyo" or the concept "place=Tokyo" appears in the user utterance history.

(D10) **Different keyword count in user utterances:** The number of times concepts not corresponding to the current value appear in the previous user utterances. For example, when the current value is Tokyo, we count the number of times non-Tokyo place names appear in the user utterance history.

(D11) **Same keyword count in system utterances:** The number of times a concept corresponding to the current value appears in the previous system utterances. For example, when the current value is Tokyo, we count the number of times the word "Tokyo" or the concept "place=Tokyo" appears in the system utterance history.

(D12) **Different keyword count in system utterances:** The number of times concepts not corresponding to the current value appear in the previous system utterances. For example, when the current value is Tokyo, we count the number of times non-Tokyo place names appear in the system utterance history.

6 Experiment

6.1 System

We prepared a telephone-based spoken dialogue system in the weather information service domain. The system provides Japan-wide weather information. Users specify a prefecture name or a city name, a date, and an information type (weather, temperature, and precipitation) to obtain the desired information.

The speech recognition engine is Julius (Lee et al., 2001) with its attached acoustic model, and the speech synthesis engine is FinalFluet (Takano et al., 2001). The system has a vocabulary of 1,652 words. The language model is a trigram trained from transcriptions obtained from our previous dialogue data collection in the same domain (Higashinaka et al., 2003a).

The system uses the 1-best speech recognition hypothesis for language understanding. We realized our understanding grammar as a weighted finite state transducer (WFST) in a manner similar to (Potamianos and Kuo, 2000). We first prepared a set of transcribed utterances labeled with dialogue acts and concepts. An utterance is assumed to have a single dialogue act with zero or more concepts. Then, we converted the utterances into a WFST. An utterance corresponds to a path, which has one dialogue act and related concepts on its path. The whole grammar is a union of such paths. The resulting WFST maps a sequence of words into a scored list of dialogue acts augmented with concepts. For example, the user utterance "Tell me the weather for tomorrow" would derive "refer-info-date" as a dialogue act with "info=weather" and "date=tomorrow" as its concepts. Compared to keyword spotting, this can be seen as imposing lexical constraints using surrounding words. The scoring for the WFST was tuned to derive as few dialogue acts and as many concepts as possible from an utterance. Since an utterance may contain several dialogue acts, we made an epsilon transition from the end of the path to the start, enabling the recursion of the dialogue acts. There are 47 dialogue acts in our grammar.

The system maintains three slots for the intention recognition result; namely, the place slot, the date slot, and the information type slot. The intention recognition results are updated by the discourse understanding rules, which update the intention recognition results using the incoming dialogue acts and concepts. The system also holds a *grounding flag* for each slot to indicate if the value of a slot has been acknowledged by the user. For example, when the system confirms by asking "Are you interested in the weather in Tokyo?" and the user says "Yes," then the grounding flags for the information type slot and

the place slot are set to *true*. We call the slots that have been acknowledged by the user the *grounded slots*.

For discourse understanding, we prepared 47 discourse understanding rules. Each rule is responsible for the processing of a particular dialogue act and its related concepts. For example, in the case of the dialogue act "refer-info-date" with concepts "info=weather" and "date=tomorrow," a rule corresponding to "refer-info-date" is invoked, which allocates the concepts to the appropriate slots. Currently, our crude rules put every concept they encounter into the associated slots without consulting the dialogue history. Since only a single value is permitted to fill a slot, previous slot-fillers are always overwritten by the new ones.

There are other rules that deal with dialogue acts that do not have associated concepts, such as acknowledgments and denials. In these cases, corresponding rules are fired to set grounding flags to particular slots or erase particular values from them. Currently, all slots which are associated with the concepts included in the previous system confirmation request are grounded or erased by the succeeding acknowledgment or denial by the user. The system also has several rules that erase the values of particular slots. For example, the user utterance "the place is wrong" yields a dialogue act "erase-place," which erases the value of the place slot. Our grammar allows two slots to be deleted at a time. Users cannot reject some values while simultaneously accepting others. The rules also handle closing remarks such as "good-bye" and the restart commands that initialize all values of the slots.

For response generation, the dialogue manager first determines whether or not the system should utter a back-channel (e.g., "uh-huh"). If the user's previous dialogue act is not of a type explicitly requesting a response from the system, and no more than one slot is filled, the system assumes that the user has not completed his/her request and utters a back-channel. If the system decides not to utter a back-channel, it then checks how many slots have been filled and grounded.

If the system finds slots that are filled but ungrounded, the system confirms these slots in one utterance. For example, when slots for place and information type have been filled with "Tokyo" and "weather" and have not been grounded, the system would utter "Are you interested in the weather in Tokyo?" Similarly, if there is only one slot that is filled and ungrounded, it only confirms that one value. An example of confirmation requests would be "Did you say Tokyo?" The system does not use an implicit confirmation strategy.

If all the slots have been filled and grounded, the system sends a query to the weather database, retrieves the weather information, formulates it into a

sentence, and utters it to the user. The current version of our system erases and resets all the slots upon delivering the weather information. If none of the above conditions match, which is the case when the user explicitly requests a response with no slots filled or two or fewer slots grounded, the system asks the user to fill the missing slots one at a time in the order of place, information type, and date. An example of the system's utterances is "Tell me the area you are interested in." All the responses are generated by templates. There are 17 templates in all, including the ones for greetings and back-channels. The templates have forms such as "Did you say [place=X]?" and "Are you interested in the [info=X] in [place=Y]?" where X and Y are taken from slot values.

6.2 Data collection

Eighteen subjects used the system over the telephone over a period of six days; three subjects per day. Each subject was given a task sheet listing the information to be requested. Each task demanded the user to ask about just one combination of a place, an information type, and a date. Therefore, if the user succeeds in the task, each dialogue in our collected data should contain one delivery of weather information from the system at the end of the dialogue. The subjects were instructed to complete the tasks one-by-one. Each subject engaged in 16 dialogues, for a total of 288 dialogues collected. Dialogues that took more than three minutes were aborted and regarded as failures. We separated the data into six groupings corresponding to the data for the six experiment dates.

The overall word error rate (WER) was 40.16%. The task completion rate was 95.83% (276/288). Figure 5 illustrates the number of turns required to complete the tasks in each grouping. Three is the minimum number of turns necessary to complete the tasks (a user's request, an acknowledgment of the system's confirmation, and a closing remark), and five out of six groupings had three as their mode value. Overall, the median number of turns is four, and the mode value is three.

The WER may seem high, but considering the nature of human-computer dialogues in which bad speech recognition prolongs dialogues, it is reasonable. We recorded the system and user utterances and the intention recognition results after each user utterance. All user utterances were transcribed.

We briefly ran an analysis of the slot samples we collected and found that most of the errors were caused by speech recognition errors. This is because neither the speech understanding component nor discourse understanding component could override slot choices provided by the speech recognizer's 1-best hypoth-

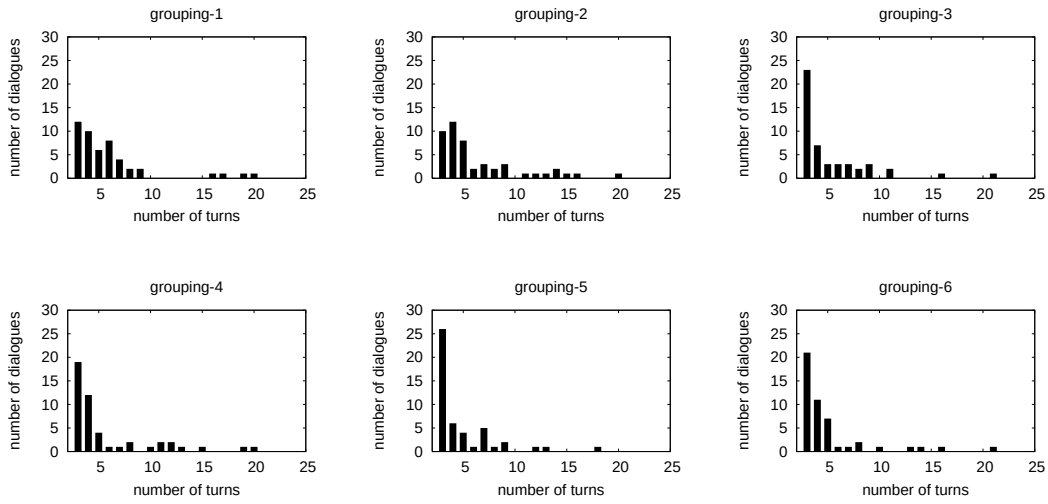

Fig. 5. The number of turns required to complete the tasks in each grouping.

esis. In addition, user utterances contained very little ambiguity needing resolution through language understanding and discourse understanding. For example, the system did not have to choose between place names for arrivals and departures as in the ATIS domain. Other than speech recognition errors, we noticed a small number of cases (15 cases) where our grammar could not output any parse for the input.

6.3 Data screening

Before training confidence models, we screened the data. Since we do not deal with slots that do not have values, we discarded the data for such slots. Then, we removed the data for slots that had a single value in the slot value sequence. The data removed here are of two types: (1) data for slots that had just been filled and (2) data for slots having the same value consecutively all along the dialogue.

The data corresponding to type (1) were removed because we consider that there is little discourse information available for these slot values. The data corresponding to type (2) were removed because we consider it difficult to differentiate (a) the cases in which values do not change because of repeated misrecognitions from (b) those in which the recognizer keeps recognizing the correct values. This is because during the data collection, users frequently repeated the same keywords/phrases for emphases and implicit confirmations. For such data, we recommend using non-discourse features as in conventional methods.

Table 1
Breakdown of the slot value samples for each grouping.

	slots	*null*	*single*	*grounded*	*single & grounded*	*error*	*selected as training sample*
grouping-1	864	267	278	41	131	2	145
grouping-2	927	215	327	33	107	0	245
grouping-3	759	246	281	28	133	0	71
grouping-4	831	262	247	39	158	0	125
grouping-5	696	236	213	29	124	0	94
grouping-6	735	245	248	32	113	0	97
Total	4,812	1,471	1,594	202	766	2	777

In addition, we did not use the data of *grounded* slots, since it is natural to consider that slots that have been grounded are basically correct.

There were 4812 slot value samples in all, and after screening, 777 samples remained (362 positive samples and 415 negative samples).

Table 1 shows the breakdown of the slot value samples for each grouping, where *null*, *single*, *grounded*, and *single & grounded* denote the number of vacant slots, slots having a single value in the slot value sequence, the grounded slots, slots that have a single value in the slot value sequence and are grounded at the same time, respectively. Here, *error* indicates that the samples were not used because acoustic and language model features could not be retrieved for them because of defects in the recorded speech files. The numbers are mutually exclusive in the table.

6.4 Feature extraction and labeling

We extracted the acoustic and language model features and discourse features for all 777 slot value samples. As the acoustic and language model features, we used the same features that Hazen et al. used in (Hazen et al., 2002) (called word-level features in their paper) with some modifications. Modifications had to be made because of the differences in speech recognizers. In addition, since the utterance score in word-level features (W14) is derived from various features of whole utterances (utterance-level features), we combined the word-level features and the utterance-level features to create a single feature vector instead of using the utterance score, making the total number of our acoustic

Table 2
List of word-level features. Labels ⟨*not available*⟩, ⟨*not used*⟩, and ⟨*new*⟩ indicate the modifications we made to the features used in (Hazen et al., 2002).

(W1)	Mean acoustic score
(W2)	Mean acoustic likelihood score ⟨*not available*⟩
(W3)	Minimum acoustic score
(W4)	Maximum acoustic score ⟨*new*⟩
(W5)	Acoustic score standard deviation
(W6)	Mean difference from maximum score
(W7)	Minimum difference from maximum score ⟨*new*⟩
(W8)	Maximum difference from maximum score ⟨*new*⟩
(W9)	Standard deviation of difference from maximum score
(W10)	Mean catch-all score ⟨*not available*⟩
(W11)	Number of acoustic observations
(W12)	N-best purity
(W13)	Number of N-best ⟨*not used*⟩
(W14)	Utterance score ⟨*not used*⟩ (utterance level features were used instead)
(W15)	Mean frame purity [2] ⟨*new*⟩
(W16)	Minimum frame purity ⟨*new*⟩
(W17)	Maximum frame purity ⟨*new*⟩

and language model features 27. We used 10-best speech recognition results for extracting the features.

Tables 2 and 3 show the acoustic and language model features we used with marks showing where the modifications were made. The label ⟨*not available*⟩ means that the feature was used in (Hazen et al., 2002), but not available for our speech recognizers, whereas ⟨*new*⟩ indicates that the feature was available, allowing us to incorporate it to our list of features. The label ⟨*not used*⟩ indicates that the feature was available, but not used as one of our features. The ⟨*not used*⟩ is only given to the number of N-best (W13 and U14) that always had a fixed value of ten in our setting. For a detailed description of the features, see (Hazen et al., 2002).

[2] Frame purity is conceptually the same as the N-best purity, with the focus on phonemes instead of words.

Table 3
List of utterance-level features. Labels ⟨*not available*⟩, ⟨*not used*⟩, and ⟨*new*⟩ indicate the modifications we made to the features used in (Hazen et al., 2002).

(U1)	Top-choice total score
(U2)	Top-choice average score
(U3)	Top-choice total N-gram score
(U4)	Top-choice average N-gram score
(U5)	Top-choice total acoustic score
(U6)	Top-choice average acoustic score
(U7)	Total score drop
(U8)	Acoustic score drop
(U9)	Lexical score drop
(U10)	Top-choice average N-best purity
(U11)	Top-choice high N-best purity
(U12)	Average N-best purity
(U13)	High N-best purity
(U14)	Number of N-best hypotheses ⟨*not used*⟩
(U15)	Top-choice number of words

As the discourse features, we used all the discourse features except D6. The feature D6 was excluded by a process of backward-elimination using the F-measure as a criterion. We used the same experimental procedure as described in Section 6.7 to find features that are not contributing to the classification performance. (Refer to Section 6.6 for the derivation of the F-measure.) The exclusion of D6 may be attributable to the inter-dependency among the features. High correlation among features is likely to hinder the training of confidence models, making it difficult to allocate appropriate weights to them.

We first hand-labeled the reference intention recognition results after each user utterance using the transcriptions, and then automatically labeled slot values as correct or incorrect. This process took several hours for our data.

6.5 Confidence model training

We trained six confidence models for intention recognition results, taking every five of the six groupings as training data and making the remaining grouping

the test data for the evaluation. For comparison, we also created, in the same way, six confidence models that only use the acoustic and language model features for training. Hereafter, we call the models trained by acoustic and language model features the *conventional models*, and the models trained by the acoustic and language model features plus the discourse features the *proposed models*.

We adopted the confidence model training method from Hazen et al. (2002). The method produces probabilistic confidence scores as log-likelihood ratios of posterior probabilities, using a weighted linear combination of the confidence feature vectors. The multi-dimensional feature vector $\vec{f}$ is reduced to the raw score r by a linear combination with a projection vector $\vec{p}$ such that

$$r = \vec{p}^T \vec{f} \tag{1}$$

We trained the projection vector $\vec{p}$ in the same manner as Hazen et al. (2002), i.e., by initializing $\vec{p}$ using a Fisher linear discriminant analysis and then updating each element of $\vec{p}$ using a hill-climbing algorithm (Powell, 1964) to minimize the classification errors in the training data.

Using r, probabilistic confidence score c is calculated as follows:

$$c = \log\left(\frac{p(r|correct)P(correct)}{p(r|incorrect)P(incorrect)}\right) - t, \tag{2}$$

where $P(correct)$ and $P(incorrect)$ are *a priori* probabilities of correct and incorrect samples in the training data, and $p(r|correct)$ and $(r|incorrect)$ are posterior probabilities for r for correct and incorrect samples, which were modeled with Gaussian density functions in this experiment. The t is a decision threshold.

Although we employed the simple linear projection model, it may also be possible to use other classification techniques, such as non-linear support vector machines and multi-layered perceptrons. However, since this paper is particularly focused on discourse features and their effect on confidence scoring, we leave investigating the use of different classifiers as future work.

6.6 Evaluation

Table 4 shows the F-measure (harmonic mean of the precision and recall) for the conventional and proposed models when each grouping was used as the test data. The result for the method that uses posterior probability of words corresponding to concepts filling the slots is also shown for reference.

Table 4
F-measure for the method that uses posterior probability, the conventional and proposed models.

Test data	F-measure		
	posterior prob.	conv.	prop.
grouping-1	0.711	**0.809**	0.803
grouping-2	0.706	0.670	**0.821**
grouping-3	**0.704**	0.645	0.689
grouping-4	0.617	0.726	**0.800**
grouping-5	0.747	0.753	**0.833**
grouping-6	0.590	0.600	**0.709**
total	0.685	0.710	**0.791**

The posterior probability was calculated on the N-best list in a similar manner to the N-best posterior probability (Wessel et al., 2001). We used 10 for N, and the scaling factor α was set to 0.03, which was found to be the best in our pilot test with 982 utterances. The utterances here were those randomly selected from the collected data. The decision threshold used for each grouping was determined to achieve minimum classification errors within the training data. Although we acknowledge that increasing N improves the calculation of the posterior probability (Wessel et al., 2001), we considered 10 to be reasonable considering the fact that the calculation has to be performed in real-time in spoken dialogue systems.

The precision, recall, and F-measure are calculated as follows:

$$\text{Precision} = \frac{\text{\# of slots correctly classified as correct}}{\text{\# of slots classified as correct}} \tag{3}$$

$$\text{Recall} = \frac{\text{\# of slots correctly classified as correct}}{\text{\# of correct slots}} \tag{4}$$

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{5}$$

It is clear from Table 4 that the proposed models perform better than the

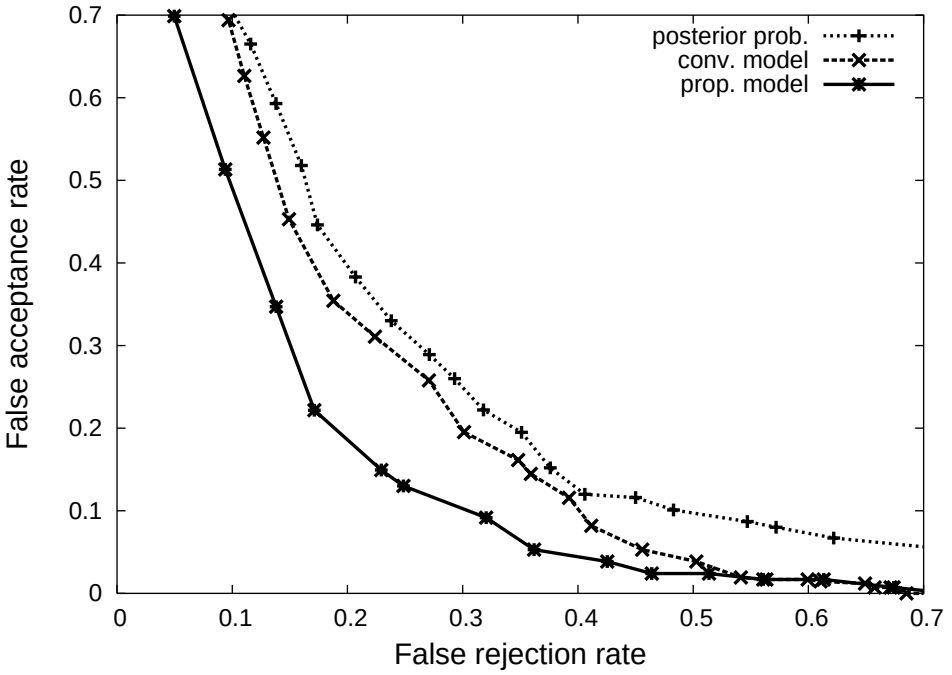

Fig. 6. False acceptance rate (FAR)–false rejection rate (FRR) curves, and for the method that uses posterior probability, the conventional and proposed models.

conventional models overall. The method that uses posterior probability is the worst performing method among the three.

Figure 6 shows the FAR-FRR curves for the three methods. The figure clearly illustrates their difference in classification performance.

The FAR and FRR are calculated as follows:

$$\mathrm{FAR} = \frac{\#\text{ of slots incorrectly classified as correct}}{\#\text{ of incorrect slots}} \tag{6}$$

$$\mathrm{FRR} = \frac{\#\text{ of slots incorrectly classified as incorrect}}{\#\text{ of correct slots}} \tag{7}$$

The FAR is the rate at which the model incorrectly classifies negative samples as positives, and the FRR the rate at which the model incorrectly classifies positives as negatives.

Table 5 shows the matrix of counts of correct and incorrect items for the conventional and proposed models. Among all the samples, there were 83 that only the proposed models classified correctly, and 37 that only the conventional models classified correctly. From a statistical test [McNemar's test (Gillick and

Table 5
Matrix of counts of correct and incorrect items for the conventional (conv.) and proposed (prop.) models.

	prop. correct	prop. incorrect
conv. correct	550	37
conv. incorrect	83	107

Table 6
F-measure for models each trained without D6 and one of the remaining discourse features.

Confidence models	F-measure	Drop in F-measure
prop. (All w/o D6)	0.791	0.000
w/o D6, D1	0.707	**0.084**
w/o D6, D2	0.756	0.035
w/o D6, D3	0.776	0.015
w/o D6, D4	0.750	0.041
w/o D6, D5	0.754	0.036
w/o D6, D7	0.751	0.040
w/o D6, D8	0.763	0.027
w/o D6, D9	0.758	0.032
w/o D6, D10	0.771	0.019
w/o D6, D11	0.765	0.025
w/o D6, D12	0.778	0.013

Cox, 1989)], it was found that the two models have a statistically significant difference in terms of classification performance ($p = 3.99 \cdot 10^{-5}$), which verifies the effectiveness of the discourse features.

6.7 Impact of the discourse features

We investigated how each of the discourse features affects the classification results. Table 6 shows the F-measure for the models, each of which was trained without D6 and one of the remaining discourse features.

Table 7
Weights assigned to each of the discourse features in the six obtained confidence models. Averages and standard deviations of the weights are shown in the last column.

	model-1	model-2	model-3	model-4	model-5	model-6	avg. (sd.)
D1	-2.212	-2.525	-1.711	-1.879	-2.074	-1.659	-2.010 (0.3286)
D2	11.802	5.966	4.809	4.040	5.436	8.130	6.697 (2.8591)
D3	0.145	0.025	-2.304	-2.362	-1.415	-0.844	-1.126 (1.0971)
D4	0.139	-0.267	-0.258	-0.425	-0.265	-0.020	-0.183 (0.2043)
D5	-0.290	0.060	0.500	-0.449	0.500	0.540	0.143 (0.4377)
D7	0.589	0.120	0.055	-0.117	0.051	0.254	0.159 (0.2428)
D8	0.414	0.262	0.275	0.255	0.243	0.227	0.279 (0.0680)
D9	1.266	1.032	1.126	0.899	1.046	1.030	1.066 (0.1219)
D10	-0.156	-0.060	-0.111	-0.143	-0.479	-0.028	-0.163 (0.1622)
D11	-0.083	-0.113	-0.107	-0.028	-0.076	-0.462	-0.145 (0.1581)
D12	-0.018	0.103	-0.021	0.028	0.020	0.033	0.024 (0.0448)

The row indexed by **prop. (All w/o D6)** represents the proposed models and the third column (Drop in F-measure) shows the drop of the F-measure from the proposed models. From the table, one can see that the same keyword pair count (D1) has a relatively large drop value, indicating that it may be more important than other features. On the other hand, the top slot purity (D3) and the different keyword count in system utterances (D12) have small drop values, indicating their possible small contribution to the classification performance.

Our finding that the same key pair count (D1) is important may suggest that Grice's maxim of quantity may be more useful than the others in terms of detecting errors in a dialogue. When we look at the weights of D1 in the confidence models, we find that the values are negative; that is, the larger the same keyword pair count, the lower the confidence. The small drop values of the top slot purity (D3) and the different keyword count in system utterance (D12) suggest that however many times different values occupy slot value sequences or system utterances, the confidence of slot values may not necessarily be affected.

Table 7 shows the weights assigned to each of the discourse features in the

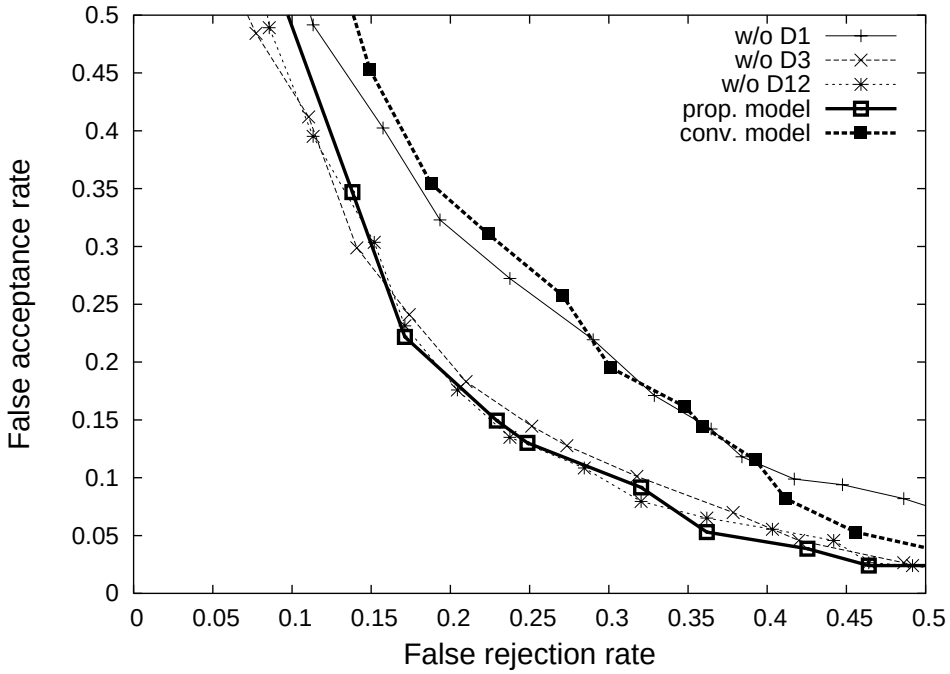

Fig. 7. False acceptance rate (FAR)–false rejection rate (FRR) curves for the proposed and conventional models and for the models that do not use D1, D3, and D12 as discourse features.

six obtained confidence models.[3] Notice that some of the features, such as D1 and D8, have very steady values for their weights compared to other features, suggesting that they play similar roles across all models. Large standard deviations in the weights of some of the features suggest that their effect on confidence scoring is likely to vary depending on the training data.

Weights that are larger than others do not necessarily reflect their importance of their associated features because of the ranges that each feature could take. For example, since the slot purity takes a value between 0 and 1, even with a very large weight, the overall effect of this feature will be limited.

Figure 7 shows the FAR-FRR curves for the models without D1, D3, and D12 along with those for the proposed models and the conventional models. It can be seen clearly that the models without D1 are close to the curve for the conventional models, and the models without D3 and D12 are almost on the curve for the proposed models.

[3] Model-1 to model-6 are the models trained with all data except grouping-1 to grouping-6, respectively.

We analyzed the successful 83 cases and found that there are mainly three patterns when our method succeeds:

(1) **Slots that have a small slot purity and a large slot variety were successfully classified as incorrect.** We found 22 samples matching this pattern. This pattern suggests the rather obvious fact that if there are many different values in the slot value sequence, the slot value becomes dubious, indicating inconsistency in user utterances.

(2) **Slots that have a large slot purity and a small slot variety were successfully classified as correct.** We found 28 samples matching this pattern. This pattern can be seen as the counterpart of the first pattern: the user's being consistent about a certain value adds confidence to that value.

By looking into the dialogue data, we noticed that dialogues in which this successful pattern was found contained the following interaction: (1) the user fills a slot relatively easily with X, (2) the slot is accidentally filled by some other value Y, and (3) the user fills the slot again with X. The conventional method was likely to find X incorrect, whereas the proposed method was likely to take X as correct. In a way, our method is using X's reliable past to boost X's confidence, overcoming the possible low acoustic and linguistic score of X.

(3) **Slots that have a small slot purity, a large slot variety, and a large same keyword count were successfully classified as correct.** We found 22 samples matching this pattern. This can be seen as a special case of the first pattern, where samples the first pattern may classify as incorrect are rescued. Here, the same keyword count is acting as a booster of the confidence.

The pattern was found in dialogues where the following type of user utterances was frequently observed: "X's weather Y," where X and Y are both associated with the same slot and X is correct and Y is wrong (misrecognition). An example would be "Tokyo's weather Kyoto," which corresponds to two dialogue acts and concepts: "refer-place-info place=Tokyo info=weather" and "refer-place place=Kyoto."

Since our discourse understanding component handles dialogue acts sequentially, after this kind of utterance, the slot value can only be Y, which makes X's slot purity very small. The same keyword count complements this small slot purity, suggesting X's potentially large slot purity.

There are 11 other samples that we could not categorize into patterns, partly because they were classified correctly by a combination of the patterns and partly because the weights for particular features were sometimes in an opposite polarity depending on the training data. Although we found two samples

where the same keyword pair count was seemingly acting as a strong indicator of incorrectness, we did not categorize them as a pattern for the lack of samples. It is surprising that the importance of the same keyword pair count was not evident in the successful samples considering the drop in the F-measure when we did not use the feature. Investigating this issue remains as future work.

7 Summary and Future work

We proposed a confidence scoring method for intention recognition results in spoken dialogue systems. Our method utilizes both discourse-related features and the acoustic and language model features of the speech recognition results to train confidence models for slot values. Experimental results show that the proposed method significantly improves the confidence scoring, indicating the effectiveness of the discourse features.

The results also indicate the usefulness of using Grice's maxims of cooperativeness to detect errors in spoken dialogue interactions. In addition, the analysis of the successful cases have revealed that the confidence model training process was capturing useful patterns to detect errors in the slot values, making the patterns possible decision rules.

As future work, firstly, we plan to perform experiments using different systems in order to verify our approach in different settings, including domains and dialogue strategies. Secondly, we would like to explore other discourse features since the discourse features presented in this paper may not sufficiently characterize the slot values. For example, we are planning to incorporate features that represent relationships and constraints among the slots because slot values tend to have dependencies in certain situations. The use of other classification techniques for confidence model training, including non-linear classification methods, should also be considered in this connection.

Thirdly, we would like to evaluate our method using workable dialogue systems. In this paper, we performed an off-line evaluation, which is based on the assumption that a corpus collected with a certain system is similar to one collected by the improved version of the system. However, in the case of interactive systems, this is not necessarily the case. Therefore, to fully verify the proposed method, an on-line (interactive) evaluation is necessary.

Finally, since Grice's maxims have been found useful for the confidence scoring of intention recognition results, we would also like to investigate the possibility of using Grice's maxims for improving the understanding component in spoken dialogue systems.

Although future work remains, the results of our experiments suggest that our approach is promising. As a final remark, we point out that the discourse features we introduced can be easily obtained as long as the system follows our assumptions about spoken dialogue systems, which facilitates application of our method to other systems.

8 Acknowledgments

We thank Naonori Ueda and Shoji Makino for their encouragement and support. Thanks also go to Kohji Dohsaka, Hajime Tsukada, Atsushi Nakamura, Matthias Denecke, Kentaro Ishizuka and Joseph Polifroni for their helpful comments on the draft version of the paper. We also thank the members of the MIT Spoken Language Systems Group for letting us use their useful confidence-scoring software. Finally, we thank the anonymous reviewers for their valuable comments and suggestions.

References

Abdou, S., Scordilis, M., 2001. Integrating multiple knowledge sources for improved speech understanding. In: Proc. Eurospeech. pp. 1783–1786.

Allen, J., Ferguson, G., Stent, A., 2001. An architecture for more realistic conversational systems. In: Proc. IUI. pp. 1–8.

Ammicht, E., Potamianos, A., Fosler-Lussier, E., 2001. Ambiguity representation and resolution in spoken dialogue systems. In: Proc. Eurospeech. pp. 2217–2220.

Baggia, P., Rullent, C., 1993. Partial parsing as a robust parsing strategy. In: Proc. ICASSP. pp. 123–126.

Béchet, F., Gorin, A. L., Wright, J. H., Hakkani-Tür, D., 2004. Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You? Speech Comm. 42, 207–225.

Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., Winograd, T., 1977. GUS, a frame driven dialog system. Artif. Intel. 8, 155–173.

Chu-Carroll, J., 2000. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In: Proc. 6th Applied NLP. pp. 97–104.

Corazza, A., Mori, R. D., Gretter, R., Satta, G., 1991. Computation of probabilities for an island-driven parser. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (9), 936–950.

Dohsaka, K., Yasuda, N., Aikawa, K., 2003. Efficient spoken dialogue control

depending on the speech recognition rate and system's database. In: Proc. Eurospeech. pp. 657–660.

Endo, T., Ward, N., Terada, M., 2002. Can confidence scores help users post-editing speech recognizer output? In: Proc. ICSLP. pp. 1469–1472.

Filisko, E. A., 2002. A context resolution server for the GALAXY conversational systems. Master's thesis, Massachusetts Institute of Technology.

Foote, J., Young, S., Jones, G., Jones, K. S., 1997. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. Computer Speech and Language 11, 207–224.

Gillick, L., Cox, S., 1989. Some statistical issues in the comparison of speech recognition algorithms. In: Proc. ICASSP. Vol. 1. pp. 532–535.

Goddeau, D., Meng, H., Polifroni, J., Seneff, S., Busayapongchai, S., 1996. A form-based dialogue manager for spoken language applications. In: Proc. ICSLP. pp. 701–704.

Grice, H. P., 1975. Logic and conversation. In: Cole, P., Morgan, J. (Eds.), Syntax and Semantics 3: Speech Acts. New York: Academic Press, pp. 41–58.

Hacioglu, K., Ward, W., 2002. A concept graph based confidence measure. In: Proc. ICASSP. pp. 225–228.

Hazen, T. J., Seneff, S., Polifroni, J., January 2002. Recognition confidence scoring and its use in speech understanding systems. Computer Speech and Language 16, 49–67.

Higashinaka, R., Miyazaki, N., Nakano, M., Aikawa, K., 2003a. Evaluating discourse understanding in spoken dialogue systems. In: Proc. Eurospeech. pp. 1941–1944.

Higashinaka, R., Nakano, M., Aikawa, K., 2003b. Corpus-based discourse understanding in spoken dialogue systems. In: Proc. 41st ACL. pp. 240–247.

Higashinaka, R., Sudoh, K., Nakano, M., 2005. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. In: Proc. ICASSP. Vol. 1. pp. 25–28.

Hirschberg, J., Litman, D., Swerts, M., 2004. Prosodic and other cues to speech recognition failures. Speech Communication 43, 155–175.

Huang, J., Zweig, G., Padmanabhan, M., 2001. Information extraction from voicemail. In: Proc. 39th ACL. pp. 290–297.

Komatani, K., Kawahara, T., 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In: Proc. 18th COLING. Vol. 1. pp. 467–473.

Kuhn, R., Mori, R. D., 1995. The application of semantic classification trees to natural language understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (5), 449–460.

Lee, A., Kawahara, T., Shikano, K., 2001. Julius – an open source real-time large vocabulary recognition engine. In: Proc. Eurospeech. pp. 1691–1694.

Lin, Y.-C., Wang, H.-M., 2001. Probabilistic concept verification for language understanding in spoken dialogue systems. In: Proc. Eurospeech. pp. 1049–1052.

Macherey, K., Och, F. J., Ney, H., 2001. Natural language understanding using statistical machine translation. In: Proc. Eurospeech. pp. 2205–2208.

Miyazaki, N., Nakano, M., Aikawa, K., to appear. Spoken dialogue understanding using an incremental speech understanding method. Systems and Computers in Japan.

Pellom, B., Ward, W., Pradhan, S., 2000. The CU communicator: an architecture for dialogue systems. In: Proc. ICSLP. Vol. 2. pp. 723–726.

Potamianos, A., Kuo, H.-K. J., 2000. Statistical recursive finite state machine parsing for speech understanding. In: Proc. ICSLP. Vol. 3. pp. 510–513.

Powell, M., 1964. An efficient method of finding the minimum of a function of several variables without calculating derivatives. The Computer Journal 7 (2), 155–162.

Pradhan, S. S., Ward, W. H., 2002. Estimating semantic confidence for spoken dialog systems. In: Proc. ICASSP. Vol. I. pp. 233–236.

Rahim, M. G., Lee, C.-H., Juang, B.-H., 1997. Discriminative utterance verification for connected digits recognition. IEEE Transactions on Speech and Audio Processing 5 (3), 266–277.

Rich, C., Sidner, C., Lesh, N., 2001. COLLAGEN: Applying collaborative discourse theory. AI Magazine 22 (4), 15–25.

Seneff, S., 1992. Robust parsing for spoken language systems. In: Proc. ICASSP. pp. 23–26.

Seneff, S., 2002. Response planning and generation in the MERCURY flight reservation system. Computer Speech and Language 16 (3–4), 283–312.

Singh, S., Litman, D., Kearns, M., Walker, M., 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. Journal of Artificial Intelligence Research 16, 105–133.

Takano, S., Tanaka, K., Mizuno, H., Abe, M., Nakajima, S., 2001. A Japanese TTS system based on multi-form units and a speech modification algorithm with harmonics reconstruction. IEEE Transactions on Speech and Audio Processing 9 (1), 3–10.

Wessel, F., Schlüter, R., Macherey, K., Ney, H., 2001. Confidence measures for large vocabulary continuous speech recognition. IEEE Transactions on Speech and Audio Processing 9 (3), 288–298.