PAPER

Ranking Multiple Dialogue States by Corpus Statistics to Improve Discourse Understanding in Spoken Dialogue Systems*

Ryuichiro HIGASHINAKA[†a)], ***Nonmember and*** **Mikio NAKANO**[††b)], ***Member***

SUMMARY This paper discusses the discourse understanding process in spoken dialogue systems. This process enables a system to understand user utterances from the context of a dialogue. Ambiguity in user utterances caused by multiple speech recognition hypotheses and parsing results sometimes makes it difficult for a system to decide on a single interpretation of a user intention. As a solution, the idea of retaining possible interpretations as multiple dialogue states and resolving the ambiguity using succeeding user utterances has been proposed. Although this approach has proven to improve discourse understanding accuracy, carefully created hand-crafted rules are necessary in order to accurately rank the dialogue states. This paper proposes automatically ranking multiple dialogue states using statistical information obtained from dialogue corpora. The experimental results in the train ticket reservation and weather information service domains show that the statistical information can significantly improve the ranking accuracy of dialogue states as well as the slot accuracy and the concept error rate of the top-ranked dialogue states.

key words: *discourse understanding, multiple dialogue states, corpus statistics, spoken dialogue systems*

1. Introduction

Due to advances in speech recognition and synthesis technologies, spoken dialogue systems have been widely deployed to handle a variety of tasks, such as flight reservations, call routing, and database searches [2]–[4]. To successfully complete the tasks, they need to accurately understand user utterances. Since the tasks are becoming increasingly complex with exchanges requiring more than a few turns, *discourse understanding*, which aims at understanding a user utterance from the context of a dialogue, is becoming important as opposed to *speech understanding*, which aims at understanding a single utterance of a user.

Spoken dialogue systems perform discourse understanding by updating a *dialogue state* using the speech understanding result, or a *dialogue act*. Here, a dialogue state means all the information that the system possesses concerning the dialogue. For example, a dialogue state includes intention recognition results after each user/system utterance, the user/system utterance history, and so forth. Since the dialogue state is used by the discourse understanding component to understand succeeding user utterances and also by the dialogue manager to create system responses, an accurate update of a dialogue state is critical to task success.

Since a speech recognizer usually outputs multiple speech recognition hypotheses and the syntactic and semantic analysis normally produces multiple parses, the discourse understanding component of a system typically receives multiple dialogue acts to update a dialogue state. Many systems use the best dialogue act candidate from the best parse of the best speech recognition hypothesis to update the dialogue state. However, this could lead to inaccuracy because the dialogue act is selected independently of the current dialogue state.

Recent work has considered all the combinations of dialogue acts and the current dialogue state to create multiple dialogue states to be ranked with regard to the context so that the best dialogue state can be selected [5]. In addition, since it is sometimes difficult to decide on a single dialogue state due to ambiguity in user utterances, an approach to keeping multiple dialogue states and resolving the ambiguity using succeeding user utterances has also been proposed [6], [7]. In this way, the correct dialogue state, which was not incidentally selected as the best interpretation in the previous turn, could survive until the next turn. Although this approach has proven to improve discourse understanding accuracy [7], the approach requires hand-crafted rules to accurately rank the dialogue states, which is costly and difficult to maintain and port to other domains. An automatic method would make the development of spoken dialogue systems scalable.

This paper proposes automatically ranking multiple dialogue states using statistical information derived from the corpora of dialogues conducted between a system and users. We hypothesize that a dialogue state that has seen the most likely sequence of dialogue acts and updates is the most probable dialogue state, and thus we use the sequential probability of dialogue act types and dialogue state updates to rank dialogue states. Previous approaches have aimed to automatically rank several conflicting understanding candidates within a dialogue state either by heuristic rules [8] or by using statistical information [9], [10], our approach is different in that we rank dialogue states which represent the system's interpretations of a whole dialogue.

In the next section, we describe the discourse understanding process in spoken dialogue systems. In Sect. 3, we describe previous work, and in Sect. 4, we explain our approach in detail. In Sect. 5, we describe the experiments we

Manuscript received November 26, 2008.
Manuscript revised April 24, 2009.

[†]The author is with NTT Communication Science Laboratories, NTT Corporation, Kyoto-fu, 619–0237 Japan.
[††]The author is with Honda Research Institute Japan Co., Ltd., Wako-shi, 351–0114 Japan.

*This paper is a substantially modified and extended version of our earlier report [1].

a) E-mail: rh@cslab.kecl.ntt.co.jp
b) E-mail: nakano@jp.honda-ri.com

DOI: 10.1587/transinf.E92.D.1771

performed to verify our approach in the train ticket reservation and weather information service domains. In the last section, we summarize the paper and mention future work.

2. Discourse Understanding in Spoken Dialogue Systems

Figure 1 shows the basic architecture of a spoken dialogue system. When receiving a user utterance, the system works as follows.

1. The speech recognizer receives a user utterance and outputs a speech recognition result, such as an N-best list or 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 (a dialogue act). Since a dialogue act is the result of understanding a single utterance, it is also referred to as a speech understanding result. A dialogue act typically comprises a *dialogue act type* that identifies the main intent of the user's utterance and its auxiliary information often encoded as attribute-value pairs. Multiple dialogue acts can be derived for an utterance when there is ambiguity in speech understanding.
3. The discourse understanding component receives the dialogue act(s), refers to the current dialogue state(s), and updates the dialogue state(s). When the system holds multiple dialogue states, they are ranked according to their scores assigned by the component with regards to the context. In this paper, we assume that a dialogue state has the user-intention recognition result as well as the history of intention recognition results, user utterances (speech recognition results and recognized dialogue acts), and system utterances (in surface forms and dialogue acts). The main target of the update is the intention recognition result because it reflects all previous exchanges of utterances between the user and the system. We assume a *frame* or an *E-form* representation for the intention recognition result [12], [13]. Therefore, an update of an intention recognition result means filling, changing, and deleting the slot values of a frame. This update is typically done by a hard-coded process [14] or by hand-crafted rules [15]. In addition, to keep track of user utterances, the dialogue act during processing is added to the history of user utterances upon the update. Although there are plan-based discourse understanding systems [16]–[18], considering the current performance of speech recognizers and the limitations in task domains, we believe frame-based discourse understanding and dialogue management are sufficient for developing systems that can be actually deployed and used by real users [19]–[22].
4. The dialogue manager refers to the updated dialogue state(s), decides the next utterance, and outputs the next content to be spoken as a dialogue act. At the same time, the dialogue manager updates the dialogue states with its dialogue act so that the dialogue states can keep a history of system utterances. When there are multiple dialogue states, the dialogue manager can choose to use only the highest ranked dialogue state or take into account several highly ranked dialogue states to generate its responses, such as 'A or B' type confirmation requests when there are some competing dialogue states.
5. The surface generation component receives the dialogue act and produces the surface expression, namely, the next words to be spoken, possibly augmented with prosodic assignment.
6. The speech synthesizer receives the surface expression and responds to the user by speech.

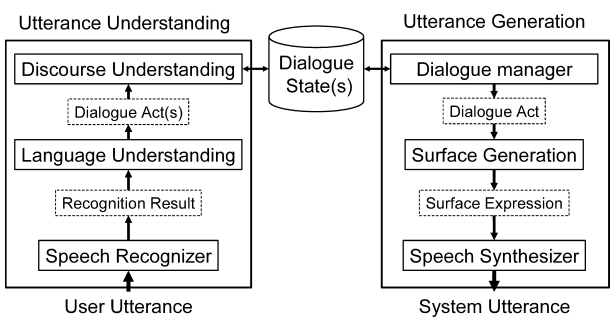

Fig. 1 Architecture of a spoken dialogue system. This figure is a modified version of the diagram we used in [11].

This paper concerns a spoken dialogue system that uses multiple dialogue states for discourse understanding and focuses on the method of ranking the multiple dialogue states. Here, the objective of discourse understanding is to obtain the best ranking of the dialogue states, not to output a single dialogue state. Compared to a system that uses only a single dialogue state, holding multiple dialogue states makes it possible for the system to resolve the ambiguity of previous user utterances with succeeding ones. It should also be noted here that we only deal with simple slot-filling applications in this paper; i.e., the intention of a user does not change during the course of a dialogue and the task can be successfully fulfilled when slots are correctly filled. In the future, we hope to deal with applications, such as query-based searches [23] and tutoring [24], in which user intentions/goals may vary depending on system responses.

Figure 2 illustrates a piece of dialogue in the train ticket reservation domain in which the user says "From Tokyo" (U1) and "From" in the utterance is inaudible and not recognized by the system. This utterance creates two dialogue acts; namely, *(refer-origin place = Tokyo)* (filling the origin slot with "Tokyo") and *(refer-dest place = Tokyo)* (filling the destination slot with "Tokyo"). As a result, two dialogue states (DS1 and DS2) are created from DS0. Note that, in Fig. 2 and also in succeeding figures (Fig. 3 and Fig. 4), we only show the intention recognition result (frame) of a dialogue state for conciseness.

Suppose that, after the system's back-channel (S2), the user says "To Kyoto" (U2), which corresponds to a dialogue

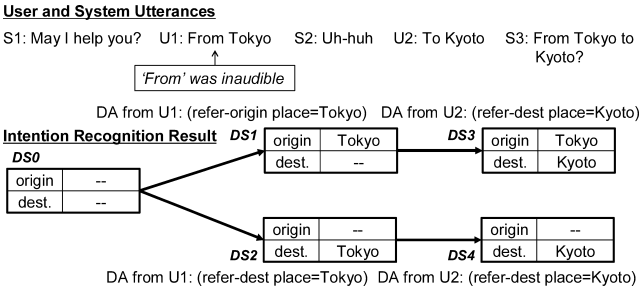

Fig. 2 Example of discourse understanding using multiple dialogue states. (S, U, DA, and DS stand for a system utterance, a user utterance, a dialogue act, and a dialogue state, respectively)

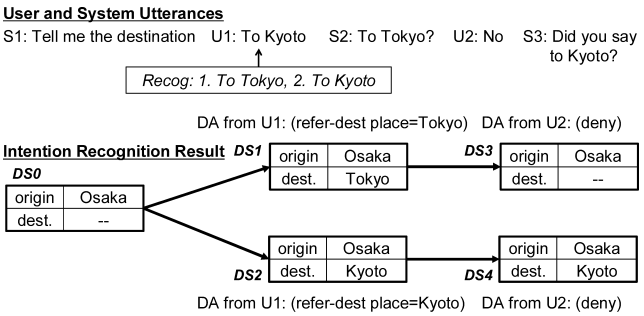

Fig. 3 Example of discourse understanding using multiple dialogue states.

act *(refer-dest place = Kyoto)*. This act creates two new dialogue states (DS3 and DS4) from DS1 and DS2, respectively. A system with a single dialogue state may choose DS2 as the best dialogue state and discard DS1, making it impossible for the system to reach DS3 after the user's next utterance U2. By having both DS1 and DS2, the system can wait for the next user utterance to decide which dialogue state (DS1 or DS2) was actually true after U1. In this example, the system successfully chooses DS3 as the best dialogue state and makes an appropriate confirmation request (S3).

Figure 3 shows another example, in which the speech recognition result outputs two hypotheses ("To Tokyo" and "To Kyoto") for the user utterance "To Kyoto" (U1). These hypotheses create two dialogue acts that result in two dialogue states (DS1 and DS2) from DS0. Suppose that the system chooses DS1 as the best dialogue state and generates a confirmation request (S2), which is denied by the user (U2). By having multiple dialogue states, the system can reconsider that DS2 was actually correct and make an appropriate confirmation request "Did you say Kyoto?" (S3) on the basis of the correct dialogue state (DS4). Note that the value of the destination slot in DS4 (Kyoto) is preserved from DS2 because the exchange of utterances "To Tokyo?" (S2) and "No" (U2) does not negate the fact that the destination is Kyoto.

In both examples, the ambiguity of a user utterance, originating from parsing and speech recognition, is preserved in the form of multiple dialogue states and then correctly disambiguated by appropriately ranking the dialogue states using the succeeding utterances, making it possible for the system to obtain the user's correct intention more accurately and efficiently. Although holding multiple dialogue states has such advantages, the problem is how to achieve appropriate ranking of the dialogue states.

3. Previous Work

Most previous work on spoken dialogue systems has not dealt with ambiguities in discourse understanding results. Although there have been several attempts to use discourse information for disambiguating speech understanding results [25], [26], the approaches do not allow ambiguities that span over multiple utterances. There is also a body of work that aims to automatically estimate the confidence of slot values using discourse information [15], [27], [28]. However, these studies do not consider keeping multiple interpretations (i.e., slot value candidates).

Bohus [10] proposed keeping multiple slot value candidates and ranking them using a confidence scoring function that takes into account various features of a dialogue, including speech recognition confidence scores for the words/concepts filling the slots, as well as discourse-level information, such as whether the confirmation request regarding the slot value has been implicitly or explicitly confirmed by the user. However, the method does not focus on ranking multiple dialogue states and the features used are specifically designed to deal with slot values. We emphasize that ranking slot values and ranking dialogue states are fundamentally different in that a dialogue state represents the system's interpretation of what has happened so far in a dialogue, making the task of ranking multiple dialogue states similar to ranking possible worlds, which has a close connection with the multi-world model [29] applied to processing written discourse. The work described in this paper aims to apply a similar model to understanding a spoken dialogue.

Nakano et al. [6] proposed holding multiple dialogue states to deal with utterances that convey meaning over several speech intervals and with the inability to determine the understanding result at the end of each interval. Multiple dialogue states are used to represent the ambiguity of whether the user has completed his/her utterance as well as the ambiguity in intention recognition results arising from multiple applicable interpretation rules. Dialogue states are scored on the basis of which interpretation rules have been applied and the scoring is based on a system developer's intuition. Miyazaki et al. [7] augmented Nakano et al.'s method to deal with n-best recognition hypotheses and reported improvement in discourse understanding accuracy, and Ammicht et al. [8] used heuristic rules (called *pragmatic analyses*) in order to keep track of several interpretations and rank them by following user input. Each of these approaches holds multiple interpretations (dialogue states) in order to deal with the discourse-level ambiguity in utterance understanding and has shown some success. However, they all rely on

hand-crafted rules.

The reliance on hand-crafted rules to rank multiple dialogue states is problematic because, when the number of dialogue states becomes large, it becomes difficult to design rules to obtain reasonable ranking results. Although only a small number of dialogue states are considered in the examples in Sect. 2, in a more realistic setting, the system has to consider the much larger number of dialogue states that can be created from N-best recognition hypotheses with N typically much larger than just one or two. Since the number of dialogue states grows exponentially as the dialogue progresses, ranking by hand can easily become intractable. Another problem is that creating rules requires expertise in dialogue system development, which hinders rapid development of systems.

Williams and Young [30] proposed having a probability distribution over dialogue states (user intentions) in order to model the understanding process of a spoken dialogue system as a partially observable Markov decision process (POMDP) and to obtain the best policy for a dialogue manager by reinforcement learning using dialogue simulations. Here, the updating of the distribution is similar to ranking multiple dialogue states. Although they offer a good framework for estimating the distribution over dialogue states from various evidences and observations in a dialogue, their current use of contextual information is limited to the previous user and system dialogue act types; e.g., they do not consider N-grams of dialogue act types or the way a dialogue state is updated. Since our aim is to find out what kind of discourse-level information is useful in ranking dialogue states, we believe our work is complementary to theirs.

4. Approach

We propose automatically ranking multiple dialogue states using statistical information that can be derived from dialogue corpora. Since a dialogue state is a result of (a) a sequence of dialogue acts by the user and system and (b) the updates by them, we hypothesize that a dialogue state that has seen the most likely sequence of dialogue acts and updates is the most probable dialogue state. For this purpose, we derive two kinds of statistical information from a corpus: (1) the N-gram probability of a dialogue act type sequence and (2) the occurrence probability of a dialogue state update pattern. We use these two probabilities to assign scores to the dialogue states for ranking.

Figure 4 shows an example of a dialogue corpus that we need in order to extract the statistical information. The corpus contains speech recognition results for each user utterance, dialogue acts for each user and system utterance, and the transition of dialogue states. In the example, the sequence from hyp-DS0 to hyp-DS3 ('hyp' stands for hypothesis) indicates the transition of the system's intention recognition result when the dialogue took place. The corpus also contains correct dialogue acts and dialogue states ('ref' stands for reference) that can be labeled later by an annotator. Here, a correct dialogue state means the dialogue state

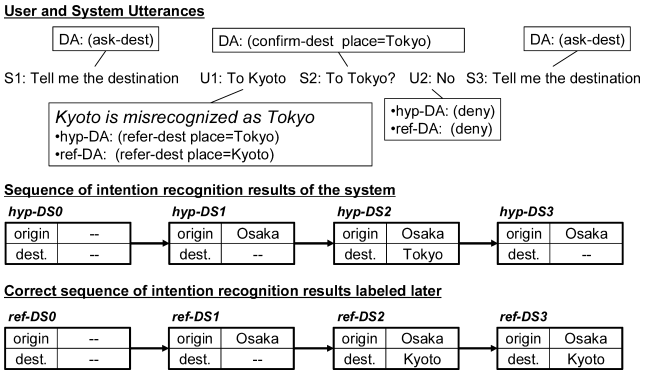

Fig. 4 Example of a dialogue corpus.

that a human overhearing the conversation would think the system should have possessed.

From such a corpus, we can obtain sequences of ref-DAs and ref-DSs, which can be used to calculate probabilities (1) and (2). If a dialogue act type sequence such as *refer-origin refer-dest* is a probable one, DS3 would be chosen over DS4 in Fig. 2, and if a dialogue state update pattern such as from ref-DS2 to ref-DS3 is found likely to occur, the system would be able to correctly choose DS4 as the best dialogue state after U2 in Fig. 3.

4.1 Statistical Information

4.1.1 N-Gram Probability of a Dialogue Act Type Sequence

We employ the N-gram probability for the probability of a dialogue act type sequence. Here, a dialogue act type sequence means a sequence of dialogue act types of both user and system utterances. N-gram probability of dialogue act types has been used to statistically estimate the next dialogue act type in disambiguating speech understanding results [31], [32]. It has also been used in finding problematic dialogues in a tutoring domain by detecting an unlikely sequence [33]. Using the same idea, we collect dialogue act type sequences from the dialogue corpus and create an N-gram language model to calculate the N-gram probability.

4.1.2 Occurrence Probability of a Dialogue State Update Pattern

We use the occurrence probability of a dialogue state update pattern for the probability of a dialogue state update. The simple bigram of dialogue states would not be sufficient due to the complexity of the data that a dialogue state possesses, which can cause data sparseness problems.

We first classify the ways that a dialogue state is updated into 96 classes characterized by seven binary attributes (Fig. 5), and then compute the occurrence probability of each class in a corpus. Note that the number of classes is not 128 (2^7) because attribute 6 is dependent on attribute 5. In the classification, an update after an open prompt

1. Whether slot values asked previously by the system are changed.
2. Whether slot values being confirmed are changed.
3. Whether slot values already confirmed (grounded) are changed.
4. Whether slots that do not have values are filled.
5. Whether slots that have values are overwritten.
6. When 5 is true, whether slot values do not change as a result.
7. Whether the system's previous utterance is an open prompt.

Fig. 5 Seven binary attributes to classify a dialogue state update.

is treated separately by having attribute 7, because such a prompt would lead to an unrestricted user utterance, leading to its own update pattern. Contrary to the N-gram probability of dialogue act types that represents a brief flow of a dialogue, the probability of a dialogue state update represents a more detailed flow of a dialogue, focusing mainly on the intention recognition result.

4.2 Scoring of Dialogue States Using the Statistical Information

Using the two probabilities, we define the score of a dialogue state $S_{t+1}^{i,j}$ as

$$S_{t+1}^{i,j} = S_t^i + \alpha \cdot s_{act}^j + \beta \cdot s_{ngram} + \gamma \cdot s_{update} \quad (1)$$

where S_t^i is the score of the i-th ($i = 1, 2, \ldots m$) dialogue state just before the update (initially set to zero), s_{act}^j the score of the j-th ($j = 1, 2, \ldots l$) dialogue act, s_{ngram} the score concerning the N-gram probability of a dialogue act type sequence, s_{update} the score concerning the occurrence probability of a dialogue state update pattern, and α, β, and γ are the weighting factors. The dialogue act score (s_{act}^j) is introduced to prioritize dialogue states updated by dialogue acts derived from reliable speech recognition and parsing results. For s_{act}^j, speech recognition confidence [15], [28] or acoustic and language model scores of speech recognition results from which the dialogue act originates can be used. The scoring of dialogue states is done only after user utterances. Although the information about system utterances is added to dialogue states after system responses, this does not alter the scores of dialogue states.

Using Eq. (1), $m \times l$ new dialogue states created from l dialogue acts and m dialogue states are scored and ranked. Since the number of dialogue states grows exponentially, we consider that the maximum number of dialogue states has to be set in order to drop low-score dialogue states and thereby perform the operation in real time. This dropping process can be considered as a beam search in view of the entire discourse process; thus, we name the maximum number of dialogue states *the dialogue state beam width*.

5. Experiment

We performed experiments to verify our approach. We first collected dialogue data using two systems in different domains and annotated the dialogues with reference dialogue acts and dialogue states so as to extract the statistical information. We then evaluated the usefulness of the statistical information by performing a dialogue-state-ranking experiment.

Table 1 List of dialogue act types for user utterances in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains.

TRAIN
refer-origin-dest, refer-origin, refer-dest, refer-date, refer-train, refer-train-number, request, end-dialogue, filler, restart, acknowledge, deny
WEATHER
refer-place-info-date, refer-prefecture, refer-city, refer-info, refer-date, deny-prefecture, deny-city, deny-info, deny-date, request, end-dialogue, filler, restart, acknowledge, deny

5.1 Systems

5.1.1 Train Ticket Reservation Domain

We prepared a Japanese spoken dialogue system in the train ticket reservation domain (hereafter the train domain). Using the system, users reserve a train seat by specifying a place of departure, destination, train type, train number, and date. The system works on the phone. The speech recognition engine is Julius [34] with its attached acoustic model trained for telephony, and the speech synthesis engine is FinalFluet [35]. The system has a vocabulary of 193 words. This small size of the vocabulary is due to the limited number of train stations assumed in our scenarios. For the language model, we used a trigram trained from randomly generated texts of acceptable phrases.

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 the same manner as described in [15]. The WFST can decode a sequence of words into a scored list of dialogue acts augmented with concepts. The top-ranked dialogue act is passed on to the discourse understanding component to update the dialogue state. There are 12 dialogue acts in our grammar (see Table 1).

The system simply uses a single dialogue state for discourse understanding because the aim of this system is to collect dialogue data to extract the statistical information. The dialogue state has five slots for the intention recognition result: *origin, destination, train type, train number, and date*. The intention recognition result is updated by manually created discourse understanding rules. We have 15 rules for this domain. For example, there is a rule to process a dialogue act *refer-dest* which fills the destination slot with the place name in the dialogue act [e.g., (refer-dest place = X) $\rightarrow$ (set destination-slot X)]. 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. For each slot, the system also holds a *grounding flag* that indicates if the value of a

slot has been acknowledged by the user.

For response generation, the dialogue manager first determines whether or not the system should utter a back-channel (e.g., "uh-huh"). Note that, in Japanese spoken dialogue, back-channels are frequently observed. If the user's dialogue act is not of a type explicitly requesting a response from the system or a filler and no more than three slots are 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. Similarly, if there is only one slot that is filled and ungrounded, it only confirms that one value. The system does not use an implicit confirmation strategy. If all the slots have been filled and grounded, the system tells the user that it has completed the reservation.

If none of the above conditions hold, which is the case when the user explicitly requests a response with no slots filled or three or fewer slots grounded, the system asks the user to fill the missing slots one at a time in the order of the place of departure, destination, date, train type, and train number. All the responses are generated by templates. There are 19 templates in all, including the ones for greetings and back-channels. The templates have forms such as "Do you want to go to [destination = X] from [origin = Y]?", where X and Y are taken from the destination slot and origin slot, respectively.

5.1.2 Weather Information Service Domain

Another system was developed in the same way in the weather information service domain (hereafter the weather domain). The system is capable of delivering Japan-wide weather information from a weather database updated regularly. The system has a vocabulary of 839 words, covering most principal cities and all prefectures in Japan.

The system uses the 1-best speech recognition hypothesis for language understanding. It uses a WFST constructed from 15 dialogue act definitions for parsing (see Table 1). The system has a single dialogue state and the intention recognition result has three slots: *place, date, and information type* (general weather, probability of precipitation, and warning). The dialogue state is updated by 14 discourse understanding rules. The system uses the same back-channeling and confirmation strategies as the train domain. The system has 17 templates for utterance generation.

5.2 Data Collection

Using the two systems, we collected dialogue data using human subjects. We recruited 15 subjects (9 males and 6 females), and each subject performed 16 dialogues (8 dialogues per system) by calling the systems on the phone. On the basis of scenarios that we prepared in advance, they were instructed to reserve certain train seats or to retrieve weather information. In the train domain, the subjects reserved a single seat in a single session, whereas weather information for multiple places and dates were sought in the weather domain in a single session because some scenarios instructed them to compare the general weather or temperature of several places. Note that the weather system cannot deliver weather information for multiple places or dates at the same time. They must be separately elicited.

We collected 120 dialogues for each domain. We recorded all speech recognition results (10-best hypotheses, although the systems used only 1-best hypotheses for understanding in the data collection), dialogue acts (parsing results of the 10-best speech recognition hypotheses), system's utterances, start and end times of user's utterances, and dialogue states before and after the user utterance. The user's voice and the system's voice were also recorded. We transcribed all user utterances. There are 1,815 and 2,090 utterances in the train and weather domains, respectively.

Dialogues that took more than three minutes were regarded as failures. The task completion rates were 88.33% (106/120) and 78.33% (94/120) in the train and weather domains, respectively. The task success for the weather domain was lower, perhaps because of the complexity of the assignments. The word error rates (WERs) were 42.08% and 48.06%, and the keyword error rates (KERs) were 31.55% and 53.07% in the train and weather domains, respectively. Here, the keywords mean the words that could fill the slots, such as dates and place names, as well as '*hai* (yes)' and '*iie* (no)'. The dialogue act recognition accuracies, which are the rates of utterances that were correctly converted into correct dialogue acts (i.e., dialogue act types and their attribute-value pairs) were 58.35% and 43.25%, and the dialogue act type recognition accuracies were 71.52% and 50.81% in the train and weather domains, respectively.

The speech recognition/understanding accuracy was rather low, probably because the input was telephone speech, the language models were not created from transcriptions of real user utterances but from artificially generated ones, and there were many phonologically similar place names in the lexicons, especially in the weather domain. Although we saw many subjects repeating the same utterance again and again until the system finally understood the user intention, considering the reasonable task success rates and that misrecognition triggers further misrecognition in human-computer dialogues, we consider this speech recognition performance to be tolerable.

5.3 Annotating Reference Dialogue Acts and Dialogue States

On the basis of the transcriptions, reference dialogue acts were annotated by hand for each user utterance in the collected dialogue data. For annotation, we used the dialogue act set defined for the data collection systems. Although dialogue acts were uniquely determined in most cases, there were sometimes utterances that were difficult to annotate using the dialogue act set. In such cases, the most appropriate

Table 2 Examples of dialogue act type sequences and their trigram probabilities in the train ticket reservation domain.

Dialogue Act Type Sequence	probability
refer-origin back-channel refer-origin-dest	0.092
refer-origin back-channel refer-origin	0.258
refer-origin back-channel refer-date	0.016
refer-origin back-channel refer-dest	0.581
refer-origin back-channel refer-train	0.016
refer-origin request confirm-origin	0.494
refer-origin confirm-origin-dest filler	0.105
refer-origin confirm-origin-dest acknowledge	0.106
refer-origin confirm-origin-dest refer-origin-dest	0.106
refer-dest back-channel refer-dest	0.038

dialogue act with regards to the domain was used for annotation. For the utterance "I would like to go to Yokohama in Kanagawa prefecture" in the train domain, since there is no dialogue act that includes an elaboration of a place "in Kanagawa prefecture", we annotated it with *(refer-dest place = Yokohama)*, which is appropriate in terms of this domain. If none of the predefined dialogue acts could be annotated for an utterance (e.g., an out-of-domain utterance), *filler* was assigned.

Using the reference dialogue acts, we automatically annotated the reference dialogue states. We made each system used in the data collection update its dialogue state by the reference dialogue acts as input. We recorded the dialogue states after the processing of each reference dialogue act as reference dialogue states. Note that the discourse understanding rules of the systems were designed to correctly update a dialogue state as long as the input is a correct dialogue act. We call the collected dialogue data with these annotations *the corpus*.

5.4 Deriving Statistical Information

5.4.1 Trigram Probability of a Dialogue Act Type Sequence

From the sequences of reference dialogue acts in the corpus, we created an N-gram language model of dialogue act types for each domain using the CMU-Cambridge Toolkit [36]. We chose three as N (trigram) and used Good-Turing discounting. We obtained the trigram probability of a dialogue act type sequence in the train and weather domains.

Table 2 shows examples of dialogue act type sequences and their trigram probabilities in the train domain calculated using the trigram language model. It can be seen from the table that the sequence {*refer-origin back-channel refer-dest*} [P(refer-dest|refer-origin, back-channel)] is much more probable, with the probability of 0.581, than {*refer-dest back-channel refer-dest*} [P(refer-dest|refer-dest, back-channel)], with 0.038. It seems very unlikely that a user would mention the destination again after the system's back-channel, which would prioritize DS3 over DS4 in Fig. 2.

Table 3 The 18 dialogue state update patterns and their occurrence probabilities in the train ticket reservation domain. See Fig. 5 for the details of the binary attributes. Attributes 1-7 are ordered from left to right.

#	Attributes (1–7)	Prob	#	Attributes (1–7)	Prob
1	0 0 0 0 1 1 0	0.3218	10	0 0 0 1 1 1 0	0.0050
2	0 0 0 0 0 0 0	0.2964	11	0 1 0 0 1 1 0	0.0044
3	0 0 0 1 0 0 0	0.1256	12	0 0 0 0 0 0 1	0.0017
4	0 0 0 1 0 0 1	0.0645	13	0 1 0 1 1 1 0	0.0017
5	0 0 1 0 0 0 0	0.0623	14	0 1 0 0 1 0 0	0.0011
6	0 1 0 0 0 0 0	0.0474	15	1 0 0 1 1 1 0	0.0011
7	1 0 0 1 0 0 0	0.0452	16	**0 1 1 0 1 0 0**	0.0006
8	0 1 0 1 0 0 0	0.0138	17	0 0 1 0 1 0 0	0.0006
9	0 1 1 0 0 0 0	0.0066	18	0 0 0 1 1 0 0	0.0006

Table 4 The 23 dialogue state update patterns and their occurrence probabilities in the weather information service domain.

#	Attributes (1–7)	Prob	#	Attributes (1–7)	Prob
1	0 0 0 0 1 1 0	0.3096	13	0 0 0 0 0 0 1	0.0072
2	0 0 0 0 0 0 0	0.2962	14	0 1 1 0 0 0 0	0.0062
3	0 0 0 1 0 0 0	0.0880	15	0 1 0 0 1 0 0	0.0053
4	0 1 0 0 0 0 0	0.0766	16	**0 0 0 0 1 0 0**	0.0038
5	0 0 0 1 0 0 1	0.0679	17	1 0 0 1 1 1 0	0.0033
6	0 0 0 1 1 1 0	0.0244	18	**1 0 0 0 1 0 0**	0.0019
7	0 1 0 0 1 1 0	0.0244	19	0 0 0 1 1 0 0	0.0014
8	0 0 1 0 0 0 0	0.0239	20	**1 0 1 0 0 0 0**	0.0010
9	0 1 0 1 0 0 0	0.0230	21	0 0 1 0 1 0 0	0.0010
10	1 0 0 1 0 0 0	0.0134	22	**0 1 1 0 1 1 0**	0.0005
11	0 1 0 1 1 1 0	0.0124	23	**0 0 1 1 0 0 0**	0.0005
12	**0 0 1 0 1 1 0**	0.0081			

5.4.2 Occurrence Probability of a Dialogue State Update Pattern

From all consecutive pairs of reference dialogue states in the collected data, we obtained the occurrence probability of each dialogue state update pattern using the classification scheme in Fig. 5.

Table 3 shows all the patterns in the corpus in the train domain. The seven binary values in the table indicate the conformity to attributes 1-7 from left to right. The patterns are ordered by the magnitude of occurrence probability. The pattern in **bold font** indicates that it is *not* observed in the weather domain (see Table 4 for comparison).

Out of 96 possible patterns, we observed 18 patterns. It can be seen from the table that there are two dominating patterns: one in which the slot values are overwritten to the same values, and another in which there is no change to the slot values. This leads us to believe that the transition from DS2 to DS4 would be more probable than from DS1 to DS3 in Fig. 3 because the former corresponds to pattern 2 with the probability of 0.2964 (no change in the slots) and the latter matches pattern 6 with the probability of 0.0623 (deleting the value of a slot being confirmed). It is intuitive that the slots were very unlikely to change. This is because, considering the transition of reference dialogue states, once the true intention of the user has been recognized, slot values should not change as long as the intention of the user is consistent, which is the case in our scenario-based dia-

logues.

Table 4 shows the update patterns found in the corpus of the weather domain. We found 23 patterns in all. Overall, the tendency of the observed patterns and their probabilities are similar to that in the train domain. The patterns that are unique in this domain are mostly the ones that conform to attribute 5; i.e., whether slots that have values are overwritten. As mentioned, the user's intention is not supposed to change during a dialogue; however, since the subjects often had to seek information for multiple places/dates in the scenarios of this domain, they sometimes changed their intentions in the middle of a dialogue largely for the purpose of avoiding repeated misrecognition.

5.5 Evaluation

5.5.1 Offline Discourse Understanding

We propose evaluating our approach by *offline discourse understanding*, in which we make the discourse understanding component process sequences of user and system utterances as they are recorded in the corpus. The discourse understanding performance is evaluated by the ranking accuracy of the dialogue states that the component outputs after each user utterance. Although we naturally believe that an online evaluation, in which dialogue experiments are performed by human subjects using a system based on our approach, is preferable, considering that a dialogue state is a system's interpretation of what has happened in a dialogue, we consider it reasonable to evaluate discourse understanding by how accurately the system makes sense of a recorded sequence of user and system utterances.

Suppose that the corpus has a dialogue as illustrated in Fig. 2. In offline discourse understanding, the initial dialogue state (DS0) is updated by the system prompt (S1). Then, the user utterance U1 is taken from the corpus to update the dialogue state using the dialogue acts for U1. Here, the dialogue acts can be those recorded in the corpus, reparsing results of the recorded speech recognition hypotheses, or those newly created by re-understanding the utterance from the recorded voice. If the system derives m dialogue acts for the utterance, the system would create m dialogue states (List-1). After the m dialogue states are updated by the system's back-channeling act (S2), n dialogue acts for U2 update the dialogue states to create $m \times n$ dialogue states (List-2). Finally, we evaluate the ranking accuracy of List-1 and List-2. Note that m and n may be different even if the same number of speech recognition hypotheses are used to derive the dialogue acts because an ambiguous utterance would create more dialogue acts than unambiguous ones do. It should also be noted that, in our current implementation, a dialogue act created from the i-th speech recognition hypothesis is treated separately from one created from the j-th speech recognition hypothesis even if they have the same dialogue act type and attribute-value pairs.

Although offline discourse understanding does not require human subjects, it can still be computationally expensive when the number of utterances to process is large because all utterances in the corpus have to be sequentially processed to create the lists of dialogue states. Even if the dialogue state beam width is set to 100, when the system uses 10-best speech recognition hypotheses for language understanding, as many as 1,000 dialogue states can be created at a time. This computational cost especially hinders experiments with varying parameters. Our idea for coping with this shortcoming is to prepare in advance lists of possible dialogue states after each user utterance using a default set of parameters and to just re-rank the stored lists of dialogue states when we need to evaluate with different parameters. The lists of dialogue states would be different when other sets of parameters are employed because of the accumulative nature of the score of a dialogue state [cf. Eq. (1)]. However, such lists would still be useful for comparing the performance of different discourse understanding methods when we focus on their relative ranking performance; i.e., the method that ranks dialogue states in the most suitable order can be considered to be better than others.

We prepared lists of dialogue states after each user utterance in the corpus. We modified the discourse understanding components of the systems used in the data collection to handle multiple dialogue states and made them sequentially understand the utterances in the corpus. Here, the utterances mean the dialogue acts stored in the corpus. A user utterance is represented by the dialogue acts derived from the 10-best speech recognition hypotheses.

We ranked each list of dialogue states using Eq. (1) with the weighting factors $\alpha = 1$, $\beta = 0$, and $\gamma = 0$, which means that the ranking was purely based on the score of dialogue acts. For s_{act}, we used the common logarithm of the posterior probability of a speech recognition hypothesis from which the dialogue act originates. The posterior probability is estimated from acoustic and language model scores as described in [37]. We stored the top-100 dialogue states after each utterance. In the train domain, we have 1,815 lists of dialogue states corresponding to the number of the utterances. In the weather domain, we have 2,090 such lists.

5.5.2 Ranking Experiment

We re-ranked each list of the stored dialogue states using Eq. (1) with different sets of weighting factors (α, β, and γ). We prepared two baselines. The first baseline (**BL-1**) used the combination [$\alpha = 1, \beta = 0, \gamma = 0$], which corresponds to the case where the ranking relies solely on the speech recognition confidence. Since this configuration is the same as that used in the data collection (i.e., top dialogue states are always derived from top recognition hypotheses), BL-1 can be regarded as emulating our rule-based data collection systems. The only difference is that multiple dialogue states are allowed instead of just one. The second baseline (**BL-2**) used the combination [$\alpha = 1, \beta = 1, \gamma = 0$] with the dialogue state beam width of one, which emulates the understanding based on dialogue act N-grams to determine the single best interpretation at each turn [31], [32].

We assumed that the weighting factors take either 1 or 0 in this experiment because we wanted to investigate how the use of the statistical information affects discourse understanding. We do not consider the case where all the weighting factors are 0 because the ranking is impossible.

For s_{act}, we use the common logarithm of the posterior probability of the speech recognition hypothesis for a dialogue act. For s_{ngram}, we use the common logarithm of the trigram probability for the dialogue act type sequence corresponding to the current user utterance, the previous system utterance, and the user's previous utterance. Since an utterance may correspond to multiple dialogue acts, the sequence would include three or more dialogue act types. For example, the utterance "No, I'd like to go to Tokyo" would correspond to *(deny)* and *(refer-dest place = Tokyo)*. Therefore, the trigram probability is normalized by the number of dialogue acts. For s_{update}, we use the common logarithm of the occurrence probability of the dialogue state update pattern.

5.5.3 Evaluation Criteria

As metrics for evaluation of discourse understanding, we used the following three, each of which evaluates discourse understanding from different perspectives.

Mean Reciprocal Rank (MRR): The MRR is a metric for evaluating ranking performance. It is the mean inverse rank of first correct answers in answer candidate lists. For example, if we have first correct answers in the second and fifth positions, the MRR becomes 0.35 $[(1/2 + 1/5)/2]$. This metric is commonly used in information retrieval and question answering systems research [38], [39], where the ranking of the answer candidates is important. Since we also evaluate ranking, we consider it an appropriate measure. To calculate the MRR, we only focus on the lists where there is a correct dialogue state because we are interested in whether a correct dialogue state can be ranked higher using the statistical information.

Slot Accuracy (ACC): Even though good ranking would mean better discourse understanding, it is also desirable that the slot values of the top-ranked dialogue state be accurate. For example, it is desirable that response generation uses accurate slot values for system confirmations in order to give better feedback to users. Having accurate top-ranked dialogue states at each turn is likely to improve user satisfaction. Therefore, we calculate the slot accuracy of the top-ranked dialogue states. The slot accuracy is calculated by dividing the number of correctly filled slots over the number of slots.

Concept Error Rate (CER): We also calculate the CER of the slot values of the top-ranked dialogue states because the CER is commonly used in speech understanding research. The CER is calculated by dividing the number of incorrect slots by the number of filled slots.

Although Higashinaka et al. [11] proposed creating an evaluation measure for discourse understanding by finding a measure that correlates closely with the performance of a dialogue system, the measure assumes that the system holds a single dialogue state. The best measure they propose is based on the precision of the update of a dialogue state (called update precision), which is difficult to calculate when a system has multiple dialogue states because it is not clear whether the sequence of dialogue states with a different understanding history can be used to calculate the update precision. We leave it as our future work to find an appropriate measure for discourse understanding when we have multiple dialogue states.

5.5.4 Results

For evaluation, we split the corpus into five sets and performed a five-fold cross validation, extracting the statistical information from four sets and evaluating with the remaining set in a round-robin fashion. For the calculation of the MRR, we used 850 and 878 lists that contained correct dialogue states in the train and weather domains, respectively. We found a large number of dialogue state lists that did not contain correct dialogue states because of many speech recognition failures. For example, if all the speech recognition hypotheses contained words/concepts that would fill the slots in a wrong way, all resulting dialogue states would be incorrect. For the slot accuracy and the CER, we used all top-ranked dialogue states in the whole lists of dialogue states. When applying offline discourse understanding to BL-2, for each list of dialogue states, we first filtered the dialogue states that were not derived from the top-ranked dialogue state in the previous list and re-ranked the remaining ones for evaluation.

Table 5 shows the evaluation results for the baselines (BL-1 and BL-2) and the combinations of the weighting factors in the train and weather domains. For the statistical comparison of the MRRs, we performed a sign test that compares the number of times a combination of weighting factors ranks the correct dialogue state higher than the baselines. We used the top-1 MRR (i.e., the MRR of top-ranked dialogue states) to compare BL-2 with the combinations of weighting factors because BL-2 re-ranks a fewer number of dialogue states derived solely from top-ranked dialogue states. It may not be fair to compare the MRRs calculated from the entire list of dialogue states because shorter lists are less likely to possess correct answers. For the slot accuracy and the CER, we calculated the mean of the slot accuracy and the CER for each dialogue and compared the number of dialogues that had the better mean of the slot accuracy or the CER.

It can be seen from the table that $[\alpha = 1, \beta = 1, \gamma = 1]$, which uses the statistical information together with the speech recognition confidence, significantly outperforms BL-1 in all evaluation criteria in both domains. The same combination of weighting factors also significantly outperforms BL-2 in the train domain in all measures, al-

Table 5 Evaluation results in Mean Reciprocal Rank (MRR), MRR-1 (top-1 MRR), Slot Accuracy (ACC), and Concept Error Rate (CER), for each combination of the weighting factors in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains. See Sect. 5.5.2 for the baselines BL-1 and BL-2.

	Weights			TRAIN				WEATHER			
	α	β	γ	MRR	MRR-1	ACC	CER	MRR	MRR-1	ACC	CER
(BL-1)	1	0	0	0.676	0.607	0.769	0.405	0.748	0.674	0.686	0.508
(BL-2)	1	1	0	0.664	0.612	0.773	0.386	0.730	0.672	0.691	0.470
	0	0	1	0.683**	0.609	0.765	0.397	0.770**	0.709+	0.695	0.496*
	0	1	0	0.738**	0.658++	0.787*	0.362*/+	0.693	0.634	0.689	0.441*
	0	1	1	0.750**	0.669++	0.792*/+	0.346**/++	0.734*	0.666	0.691*	0.452**
	1	0	1	0.696**	0.623	0.765	0.397	0.771**	0.712+	0.695	0.496
	1	1	0	0.747**	0.666++	0.791**/++	0.359**/++	0.722	0.658	0.697	0.433**
	1	1	1	0.761**	0.685++	0.795**/++	0.341**/++	0.748**	0.682	0.696**	0.447**

** Statistical significance (* p<0.05, ** p<0.01) over BL-1.
++ Statistical significance (+ p<0.05, ++ p<0.01) over BL-2.

though its performance gain is limited in the weather domain with only [$\alpha = 0, \beta = 0, \gamma = 1$] and [$\alpha = 1, \beta = 0, \gamma = 1$] significantly outperforming BL-2 in the top-1 MRR and with the CER tending to improve as we use the statistical information. The fact that the statistical information improves discourse understanding in most cases and that BL-2 is outperformed in ranking by some combinations of weighting factors suggest the effectiveness of our approach of combining the statistical information with multiple dialogue states. Remember that BL-2 uses the dialogue state beam width of one.

Overall, the train domain benefited more from the use of the statistical information. We consider this is due to the ambiguity of dialogue acts that resides in the train domain; i.e., utterances with bare place names would yield multiple dialogue acts referring to places of departures and destinations. On the other hand, the ambiguity of utterances solely come from the multiple speech recognition hypotheses in the weather domain. This is demonstrated by the fact that the use of the trigram probability of a dialogue act type sequence ($\beta = 1$) does not affect the results as much as the probability of a dialogue state update pattern does ($\gamma = 1$) in the weather domain; that is, there seems to be less need to disambiguate dialogue acts. In some cases, understanding based on only the statistical information ($\alpha = 0$) is better than relying only on the speech recognition confidence, probably due to numerous speech recognition errors.

5.5.5 Impact of the Dialogue State Beam Width

We calculated the MRR by placing a limit on the maximum number of dialogue states to hold (dialogue state beam width) just as we set the dialogue state beam width to one for BL-2. By changing the dialogue state beam width, it is possible to examine how the number of dialogue states could affect ranking accuracy.

Figure 6 shows the performance changes in the top-1 MRR with different dialogue state beam widths. We used $\alpha = 1, \beta = 1$, and $\gamma = 1$ as the weighting factors. It can be seen that as the system holds more dialogue states, the top-1 MRR improves. However, the improvement begins to saturate and decreases when the dialogue state beam width is around 50 and 5 for the train and weather domains, respectively, indicating that the sufficient number of dialogue states may depend on the domains and that having too many dialogue states may have an adverse effect. Despite that this result re-confirms that having multiple dialogue states can actually improve discourse understanding and that the system with multiple dialogue states could work in real time because the number of dialogue states to hold could be fewer than 100, which would not impose a computational problem.

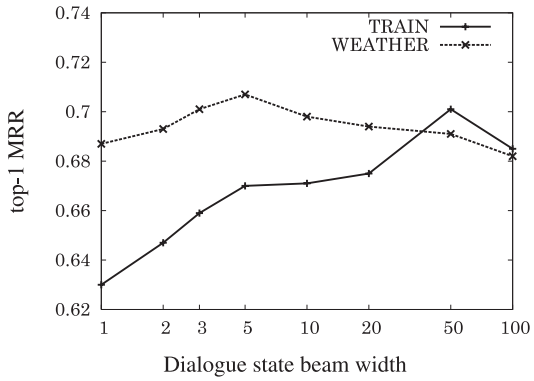

Fig. 6 Performance changes in the Mean Reciprocal Rank (top-1 MRR) with different dialogue state beam widths in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains. The x-axis is on the log scale.

6. Summary and Future Work

We proposed a new discourse understanding method that ranks multiple dialogue states using the statistical information obtained from dialogue corpora. The method uses the combination of (1) the trigram probability of dialogue act types, (2) the occurrence probability of a dialogue state update pattern, and (3) the speech recognition confidence of a dialogue act to score a dialogue state.

Experimental results in the train ticket reservation domain and the weather information service domain show that our approach can significantly improve the ranking of the

dialogue states over two baselines, one based only on the speech recognition confidence and another which holds only the top-ranked multiple dialogue state after the understanding of each utterance using dialogue act type N-grams, although the improvement seems to be limited in the weather domain. Since the discourse understanding performance generally improves as we use (1) and (2) together with (3), we consider that our approach successfully incorporates information suitable for discourse understanding. We also confirmed that it is effective to hold multiple dialogue states for discourse understanding and that the sufficient number of dialogue states to hold could be fewer than 100.

Our contribution lies in showing the possibility of using dialogue corpora to achieve accurate discourse understanding without the use of costly hand-crafted rules and also in our derivation of the dialogue state bigram probability by classifying a dialogue state update by seven attributes. Note that a simple bigram of dialogue states would have been too sparse given the complex data structure of a dialogue state. Although our approach still requires dialogue corpora to derive the statistical information, we believe the expertise required in creating the discourse understanding component can be greatly reduced by our approach.

There remain several issues that we still need to explore. These include the exploration of statistical information other than the probability of a dialogue act type sequence and the occurrence probability of a dialogue state update pattern. We also need to optimize the weighting factors α, β, and γ because we simply used 0 or 1 for the weighting factors as a first step to examine how the use of the statistical information could affect discourse understanding in this paper. It should also be noted that the experiment we performed was an offline evaluation. An online evaluation would be desirable for a more accurate evaluation. More experiments in larger domains would also be necessary to fully verify our approach. Despite these issues, the present results show that our approach is promising.

Acknowledgements

We thank all members of the Knowledge Processing Research Group for helpful discussions and comments. We thank Hiromi Nakaiwa, Naonori Ueda and Shoji Makino for their encouragement and support. Thanks also go to Hideki Isozaki, Kiyoaki Aikawa, Kohji Dohsaka, Noboru Miyazaki, Kentaro Ishizuka, and Matthias Denecke for their helpful comments and suggestions.

References

[1] R. Higashinaka, M. Nakano, and K. Aikawa, "Corpus-based discourse understanding in spoken dialogue systems," Proc. 41st ACL, pp.240–247, 2003.

[2] A.I. Rudnicky, C. Bennett, A. Black, A. Chotomongcol, K. Lenzo, A. Oh, and R. Singh, "Task and domain specific modelling in the Carnegie Mellon Communicator system," Proc. ICSLP, pp.130–134, 2000.

[3] A.L. Gorin, G. Riccardi, and J.H. Wright, "How may I help you?," Speech Commun., vol.23, pp.113–127, 1997.

[4] K. Komatani, N. Kanda, T. Ogata, and H.G. Okuno, "Contextual constraints based on dialogue models in database search task for spoken dialogue systems," Proc. Eurospeech, pp.877–880, 2005.

[5] C. Wutiwiwatchai and S. Furui, "Belief-based nonlinear rescoring in Thai speech understanding," Proc. Interspeech, pp.2129–2133, 2004.

[6] M. Nakano, N. Miyazaki, J. Hirasawa, K. Dohsaka, and T. Kawabata, "Understanding unsegmented user utterances in real-time spoken dialogue systems," Proc. 37th ACL, pp.200–207, 1999.

[7] N. Miyazaki, M. Nakano, and K. Aikawa, "Spoken dialogue understanding using and incremental speech understanding method," Systems and Computers in Japan, vol.36, no.12, pp.75–84, 2005.

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

[9] D. Bohus and A. Rudnicky, "A K hypotheses + other belief updating model," Proc. AAAI Workshop on Statistical and Empirical Approaches to Spoken Dialogue Systems, 2006.

[10] D. Bohus, Error Awareness and Recovery in Conversational Spoken Language Interfaces, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, 2007.

[11] R. Higashinaka, N. Miyazaki, M. Nakano, and K. Aikawa, "Evaluating discourse understanding in spoken dialogue systems," ACM Trans. Speech Lang. Process., vol.1, pp.1–20, 2004.

[12] D.G. Bobrow, R.M. Kaplan, M. Kay, D.A. Norman, H. Thompson, and T. Winograd, "GUS, a frame driven dialog system," Artif. Intell., vol.8, pp.155–173, 1977.

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

[14] E.A. Filisko, A context resolution server for the galaxy conversational systems, Master's Thesis, Massachusetts Institute of Technology, 2002.

[15] R. Higashinaka, K. Sudoh, and M. Nakano, "Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems," Speech Commun., vol.48, no.3-4, pp.417–436, 2006.

[16] J.F. Allen and C.R. Perrault, "Analyzing intention in utterances," Artif. Intell., vol.15, pp.143–178, 1980.

[17] S. Carberry, Plan Recognition in Natural Language Dialogue, MIT Press, Cambridge, Mass., 1990.

[18] C. Rich, C. Sidner, and N. Lesh, "COLLAGEN: Applying collaborative discourse theory," AI Magazine, vol.22, no.4, pp.15–25, 2001.

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

[20] S. Seneff, "Response planning and generation in the MERCURY flight reservation system," Comput. Speech Lang., vol.16, no.3-4, pp.283–312, 2002.

[21] A. Raux, B. Langner, D. Bohus, A.W. Black, and M. Eskenazi, "Doing research in a deployed spoken dialog system: One year of let's go! public experience," Proc. Interspeech, pp.65–68, 2006.

[22] K. Komatani, T. Kawahara, and H.G. Okuno, "Analyzing temporal transition of real user's behaviors in a spoken dialogue system," Proc. Interspeech, pp.142–145, 2007.

[23] C. Hori, T. Hori, H. Tsukada, H. Isozaki, Y. Sasaki, and E. Maeda, "Spoken interactive ODQA system: SPIQA," Proc. ACL, Interactive Poster and Demonstration Session, pp.153–156, 2003.

[24] D.J. Litman and S. Silliman, "ITSPOKE: An intelligent tutoring spoken dialogue system," Proc. HLT-NAACL, pp.5–8, 2004 (demonstration paper).

[25] M. Purver, F. Ratiu, and L. Cavedon, "Robust interpretation in dialogue by combining confidence scores with contextual features," Proc. Interspeech, pp.1–4, 2006.

[26] N. Fujiwara, T. Itoh, K. Araki, A. Kai, T. Konishi, and Y. Itoh, "Spoken language understanding method using confidence measure and dialogue history," Systems and Computers in Japan, vol.38, no.9,

pp.21–31, 2007.
[27] S.S. Pradhan and W.H. Ward, "Estimating semantic confidence for spoken dialog systems," Proc. ICASSP, pp.233–236, 2002.
[28] T.J. Hazen, S. Seneff, and J. Polifroni, "Recognition confidence scoring and its use in speech understanding systems," Comput. Speech Lang., vol.16, pp.49–67, 2002.
[29] K. Nagao, "Semantic interpretation based on the multi-world model," Proc. 11th IJCAI, pp.1467–1472, 1989.
[30] J.D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," Comput. Speech Lang., vol.21, no.2, pp.393–422, 2007.
[31] M. Nagata and T. Morimoto, "First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance," Speech Commun., vol.15, pp.193–203, 1994.
[32] N. Reithinger and E. Maier, "Utilizing statistical dialogue act processing in Verbmobil," Proc. 33rd ACL, pp.116–121, 1995.
[33] K. Forbes-Riley and D.J. Litman, "Using bigrams to identify relationships between student certainness states and tutor responses in a spoken dialogue corpus," Proc. SIGDIAL, pp.87–96, 2005.
[34] A. Lee, T. Kawahara, and K. Shikano, "Julius – An open source real-time large vocabulary recognition engine," Proc. Eurospeech, pp.1691–1694, 2001.
[35] S. Takano, K. Tanaka, H. Mizuno, M. Abe, and S. Nakajima, "A Japanese TTS system based on multi-form units and a speech modification algorithm with harmonics reconstruction," IEEE Trans. Speech Audio Process., vol.9, no.1, pp.3–10, 2001.
[36] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," Proc. Eurospeech, pp.2707–2710, 1997.
[37] G. Bouwman, J. Sturm, and L. Boves, "Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project," Proc. ICASSP, pp.493–496, 1999.
[38] E.M. Voorhees and D.M. Tice, "Building a question answering test collection," Proc. SIGIR, pp.200–207, 2000.
[39] R. Higashinaka and H. Isozaki, "Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions," ACM Trans. Asian Lang. Inform. Process., vol.7, no.2, Article 6 (29 pages), 2008.

Ryuichiro Higashinaka is a Research Scientist at NTT Communication Science Laboratories, NTT Corporation. He received the B.A. degree in environmental information, the Master of Media and Governance degree, and the ph.D degree from Keio University, Tokyo, in 1999, 2001, and 2008, respectively. He joined NTT in 2001. From 2004 to 2006, he was a visiting researcher at the University of Sheffield, UK. His research interests are in utterance understanding and generation in spoken dialogue systems and question answering systems. He is a member of the Association for Natural Language Processing and IPSJ.

Mikio Nakano is a Senior Researcher at Honda Research Institute Japan Co., Ltd. (HRI-JP). He received his M.S. degree in Coordinated Sciences and Sc.D. degree in Information Science from the University of Tokyo, respectively in 1990 and 1998. From 1990 to 2004, he worked for Nippon Telegraph and Telephone Corporation. In 2004, he joined HRI-JP. His research interests include speech understanding, spoken dialogue systems, and conversational robots. He is a member of ACM, ACL, AAAI, ISCA and other academic societies.