Multi-Domain Spoken Dialogue System with Extensibility and Robustness against Speech Recognition Errors

Kazunori Komatani   Naoyuki Kanda   Mikio Nakano[†]
Kazuhiro Nakadai[†] **Hiroshi Tsujino**[†] **Tetsuya Ogata   Hiroshi G. Okuno**

Kyoto University, Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,ogata,okuno}@i.kyoto-u.ac.jp

† Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama 351-0188, Japan
{nakano,nakadai,tsujino}@jp.honda-ri.com

Abstract

We developed a multi-domain spoken dialogue system that can handle user requests across multiple domains. Such systems need to satisfy two requirements: extensibility and robustness against speech recognition errors. Extensibility is required to allow for the modification and addition of domains independent of other domains. Robustness against speech recognition errors is required because such errors are inevitable in speech recognition. However, the systems should still behave appropriately, even when their inputs are erroneous. Our system was constructed on an extensible architecture and is equipped with a robust and extensible domain selection method. Domain selection was based on three choices: (I) the previous domain, (II) the domain in which the speech recognition result can be accepted with the highest recognition score, and (III) other domains. With the third choice we newly introduced, our system can prevent dialogues from continuously being stuck in an erroneous domain. Our experimental results, obtained with 10 subjects, showed that our method reduced the domain selection errors by 18.3%, compared to a conventional method.

1 Introduction

Many spoken dialogue systems have been developed for various domains, including: flight reservations (Levin et al., 2000; Potamianos and Kuo, 2000; San-Segundo et al., 2000), train travel information (Lamel et al., 1999), and bus information (Komatani et al., 2005b; Raux and Eskenazi, 2004). Since these systems only handle a single domain, users must be aware of the limitations of these domains, which were defined by the system developer. To handle various domains through a single interface, we have developed a multi-domain spoken dialogue system, which is composed of several single-domain systems. The system can handle complicated tasks that contain requests across several domains.

Multi-domain spoken dialogue systems need to satisfy the following two requirements: (1) extensibility and (2) robustness against speech recognition errors. Many such systems have been developed on the basis of a master-slave architecture, which is composed of a single master module and several domain experts handling each domain. This architecture has the advantage that each domain expert can be independently developed, by modifying existing experts or adding new experts into the system. In this architecture, the master module needs to select a domain expert to which response generation and dialogue management for the user's utterance are committed. Hereafter, we will refer to this selecting process **domain selection**.

The second requirement is robustness against speech recognition errors, which are inevitable in systems that use speech recognition. Therefore, these systems must robustly select domains even when the input may be incorrect due to speech recognition errors.

We present an architecture for a multi-domain spoken dialogue system that incorporates a new domain selection method that is both extensible and robust against speech recognition errors. Since our system is based on extensible architecture similar to that developed by O'Neill (O'Neill et al., 2004), we can add and modify the domain

Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, pages 9–17,
Sydney, July 2006. ©2006 Association for Computational Linguistics

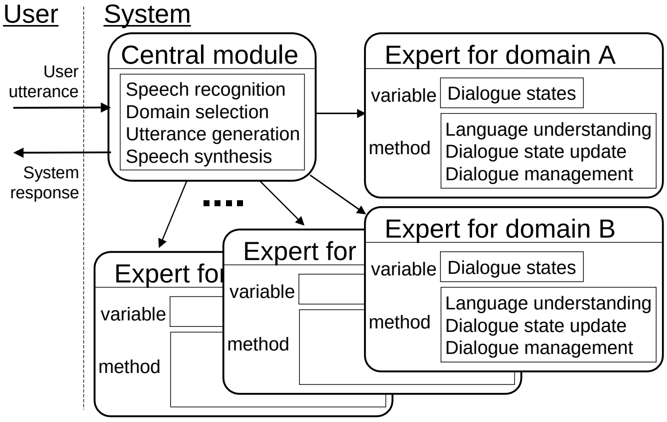

Figure 1: Distributed-type architecture for multi-domain spoken dialogue systems

experts easily. In order to maintain robustness, domain selection takes into consideration various features concerning context and situations of the dialogues. We also designed a new selection framework that satisfies the extensibility issue by abstracting the transitions between the current and next domains. Specifically, our system selects the next domain based on: (I) the previous domain, (Ⅱ) the domain in which the speech recognition result can be accepted with the highest recognition score, and (Ⅲ) other domains. Conventional methods cannot select the correct domain when neither the previous domain nor the speech recognition results for a current utterance are correct. To overcome this drawback, we defined another choice as (Ⅲ) that enables the system to detect an erroneous situation and thus prevent the dialogue from continuing to be incorrect. We modeled this framework as a classification problem using machine learning, and showed it is effective by performing an experimental evaluation of 2,205 utterances collected from 10 subjects.

2 Architecture used for Multi-Domain Spoken Dialogue Systems

In multi-domain spoken dialogue systems, the system design is more complicated than in single domain systems. When the designed systems are closely related to each other, a modification in a certain domain may affect the whole system. This type of a design makes it difficult to modify existing domains or to add new domains. Therefore, a distributed-type architecture has been previously proposed (Lin et al., 2001), which enables system developers to design each domain independently. In this architecture, the system is composed of two kinds of components: a part that can be designed independently of all other domains, and a part in which relations among domains should be considered. By minimizing the latter component, a system developer can design each domain semi-independently, which enables domains to be easily added or modified. Many existing systems are based on this architecture (Lin et al., 2001; O'Neill et al., 2004; Pakucs, 2003; Nakano et al., 2005).

Thus, we adopted the distributed-type architecture (Nakano et al., 2005). Our system is roughly composed of two parts, as shown in Figure 1: several experts that control dialogues in each domain, and a central module that controls each expert. When a user speaks to the system, the central module drives a speech recognizer, and then passes the result to each domain expert. Each expert, which controls its own domains, executes a language understanding module, updates its dialogue states based on the speech recognition result, and returns the information required for domain selection[1]. Based on the information obtained from the experts, the central module selects an appropriate domain for giving the response. An expert then takes charge of the selected domain and determines the next dialogue act based on its dialogue state. The central module generates a response based on the dialogue act obtained from the expert, and outputs the synthesized speech to the user. Communications between the central module and each expert are realized using method-calls in the central module. Each expert is required to have several methods, such as utterance understanding or response selection, to be considered an expert

[1] Dialogue states in a domain that are not selected during domain selection are returned to their previous states.

in this architecture.

As was previously described, the central module is not concerned with processing the speech recognition results; instead, the central module leaves this task to each expert. Therefore, it is important that the central module selects an expert that is committed to the process of the speech recognition result. Furthermore, information used during domain selection should also be domain independent, because this allows easier domain modification and addition, which is, after all, the main advantage of distributed-type architecture.

3 Extensible and Robust Domain Selection

Domain selection in the central module should also be performed within an extensible framework, and also should be robust against speech recognition errors.

In many conventional methods, domain selection is based on estimating the most likely domains based on the speech recognition results. Since these methods are heavily dependent on the performance of the speech recognizers, they are not robust because the systems will fail when a speech recognizer fails. To behave robustly against speech recognition errors, the success of speech recognition and of domain selection should be treated separately. Furthermore, in some conventional methods, accurate language models are required to construct the domain selection parts before new domains are added to a multi-domain system. This means that they are not extensible.

When selecting a domain, other studies have used the information on the domain in which a previous response was made. Lin et al. (2001) gave preference to the domain selected in the previous turn by adding a certain score as an award when comparing the N-best candidates of the speech recognition for each domain. Lane and Kawahara (2005) also assigned a similar preference in the classification with Support Vector Machine (SVM). A system described in (O'Neill et al., 2004) does not change its domain until its sub-task is completed, which is a constraint similar to keeping dialogue in one domain. Since these methods assume that the previous domain is most likely the correct domain, it is expected that these methods keep a system in the domain despite errors due to speech recognition problems. Thus, should domain selection be erroneous, the damage due to the error is compounded, as the system assumes that the previous domain is always correct. Therefore, we solve this problem by considering features that represent the confidence of the previously selected domain.

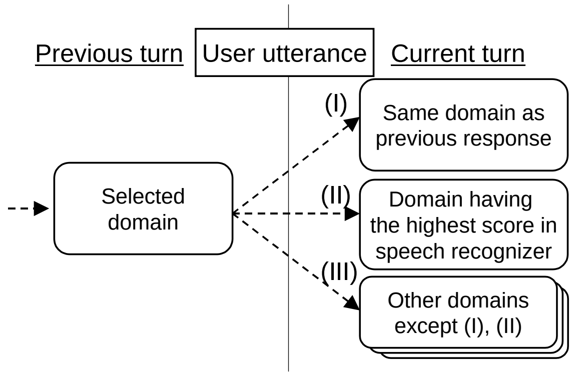
Previous turn
User utterance
Current turn
Selected domain
(I)
Same domain as previous response
(II)
Domain having the highest score in speech recognizer
(III)
Other domains except (I), (II)

Figure 2: Overview of domain selection

We define domain selection as being based on the following 3-class categorization: (I) the previous domain, (Ⅱ) the domain in which the speech recognition results can be accepted with the highest recognition score, which is different from the previous domain, and (Ⅲ) other domains. Figure 2 depicts the three choices. This framework includes the conventional methods as choices (I) and (Ⅱ). Furthermore, it considers the possibility that the current interpretations may be wrong, which is represented as choice (Ⅲ). This framework also has extensibility for adding new domains, since it treats domain selection not by detecting each domain directly, but by defining only a relative relationship between the previous and current domains.

Since our framework separates speech recognition results and domain selection, it can keep dialogues in the correct domain even when speech recognition results are wrong. This situation is represented as choice (I). An example is shown in Figure 3. Here, the user's first utterance (U1) is about the restaurant domain. Although the second utterance (U2) is also about the restaurant domain, an incorrect interpretation for the restaurant domain is obtained because the utterance contains an out-of-vocabulary word and is incorrectly recognized. Although a response for utterance U2 should ideally be in the restaurant domain, the system control shifts to the temple sightseeing information domain, in which an interpretation is obtained based on the speech recognition result. This

U1: **Tell me bars in Kawaramachi area.**
(domain: restaurant)

S1: Searching for bars in Kawaramachi area.
30 items found.

U2: **I want Tamanohikari (name of liquor).**
(domain: restaurant)
Tamanohikari is out-of-vocabulary word, and misrecognized as Tamba-bashi (name of place). (domain: temple)

S2 (bad): Searching spots near Tamba-bashi. 10 items found. (domain: temple)

S2 (good): I do not understand what you said. Do you have any other preferences? (domain: restaurant)

Figure 3: Example in which choice (I) is appropriate in spite of speech recognition error

U1: **Tell me the address of Horin-ji** (temple name)**.**
(domain: temple)

S1: The address of Horin-ji is ...

U2: **Then, what is the fee for Horin-ji?**
(domain: temple)
misrecognized as "the fee of Holiday Inn".
(domain: hotel)

S2: The fee of Holiday Inn is ...

U3: **The fee of Horin-ji.** (domain: temple)
again misrecognized as "the fee of Holiday Inn".
(domain: hotel)

S3 (bad): The fee of Holiday Inn is ...

S3 (good): Are you asking about hotel information?

U4: **No.**

S4: Would you like to return to the temple information service?

Figure 4: Example in which choice (Ⅲ) should be selected

is shown as utterance S2 (bad). In such cases, our framework is capable of behaving appropriately. This is shown as S2 (good), which is made by selecting choice (I). Accepting erroneous recognition results is more harmful than rejecting correct ones for the following reasons: 1) a user needs to solve the misunderstanding as a result of the false acceptance, and 2) an erroneous utterance affects the interpretation of the utterances following it.

Furthermore, we define choice (Ⅲ), which detects the cases where normal dialogue management is not suitable, in which case the central module selects an expert based on either the previous domain or the domain based on the speech recognition results. The situation corresponds to a succession of recognition errors. However, this problem is more difficult to solve than merely detecting a simple succession of the errors because the system needs to distinguish between speech recognition errors and domain selection errors in order to generate appropriate next utterances. Figure 4 shows an example of such a situation. Here, the user's utterances U1 and U2 are about the temple domain, but a speech recognition error occurred in U2, and system control shifts to the hotel domain. The user again says (U3), but this results in the same recognition error. In this case, a domain that should ideally be selected is neither the domain in the previous turn nor the domain determined based on the speech recognition results. If this situation can be detected, the system should be able to generate an appropriate response, like S3 (good), and prevent inappropriate responses based on an incorrect domain determination. It is possible for the system to restart from two utterances before (U1), after asking a confirmatory question (S4) about whether to return to it or not. After that, repetition of similar errors can also be avoided if the system prohibits transition to the hotel domain.

4 Domain Selection using Dialogue History

We constructed a classifier that selects the appropriate domains using various features, including dialogue histories. The selected domain candidates are based on: (I) the previous domain, (Ⅱ) the domain in which the speech recognition results can be accepted with the highest recognition score, or (Ⅲ) other domains. Here, we describe the features present in our domain selection method.

In order to not spoil the system's extensibility, an advantage of the distributed-type architecture, the features used in the domain selection should not depend on the specific domains. We categorize the features used into three categories listed below:

- Features representing the confidence with which the previous domain can be considered correct (Table 1)
- Features about a user's speech recognition result (Table 2)

Table 1: Features representing confidence in previous domain

P1:	number of affirmatives after entering the domain
P2:	number of negations after entering the domain
P3:	whether tasks have been completed in the domain (whether to enter "requesting detailed information" in database search task)
P4:	whether the domain appeared before
P5:	number of changed slots after entering the domain
P6:	number of turns after entering the domain
P7:	ratio of changed slots ($= \mathrm{P5}/\mathrm{P6}$)
P8:	ratio of user's negative answers ($= \mathrm{P2}/(\mathrm{P1} + \mathrm{P2})$)
P9:	ratio of user's negative answers in the domain ($= \mathrm{P2}/\mathrm{P6}$)
P10:	states in tasks

Table 2: Features of speech recognition results

R1:	best posteriori probability of the N-best candidates interpreted in the previous domain
R2:	best posteriori probability for the speech recognition result interpreted in the domain, that is the domain with the highest score
R3:	average of word's confidence scores for the best candidate of speech recognition results in the domain, that is, the domain with the highest score
R4:	difference of acoustic scores between candidates selected as (I) and (Ⅱ)
R5:	ratio of averages of words' confidence scores between candidates selected as (I) and (Ⅱ)

Table 3: Features representing situations after domain selection

C1:	dialogue state after the domain selection after selecting previous domain
C2:	whether the interpretation of the user's utterance is negative in previous domain
C3:	number of changed slots after selecting previous domain
C4:	dialogue state after selecting the domain with the highest speech recognition score
C5:	whether the interpretation of the user's utterance is negative in the domain with the highest speech recognition score
C6:	number of changed slots after selecting the domain with the highest speech recognition score
C7:	number of common slots (name of place, here) changed after selecting the domain with the highest speech recognition score
C8:	whether the domain with the highest speech recognition score has appeared before

- Features representing the situation after domain selection (Table 3)

We can take into account the possibility that a current estimated domain might be erroneous, by using features representing the confidence in the previous domain. Each feature from P1 to P9 is defined to represent the determination of whether an estimated domain is reliable or not. Specifically, if there are many affirmative responses from a user or many changes of slot values during interactions in the domain, we regard the current domain as reliable. Conversely, the domain is not reliable if there are many negative answers from a user after entering the domain.

We also adopted the feature P10 to represent the state of the task, because the likelihood that a domain is changed depends on the state of the task. We classified the tasks that we treat into two categories using the following classifications first made by Araki et al. (1999). For a task categorized as a "slot-filling type", we defined the dialogue states as one of the following two types: "*not completed*", if not all of the requisite slots have been filled; and "*completed*", if all of the requisite slots have been filled. For a task categorized as a "database search type", we defined the dialogue states as one of the following two types: "*specifying query conditions*" and "*requesting detailed information*", which were defined in (Komatani et al., 2005a).

The features which represent the user's speech recognition result are listed in Table 2 and correspond to those used in conventional studies. R1 considers the N-best candidates of speech recognition results that can be interpreted in the previous domain. R2 and R3 represent information about a domain with the highest speech recognition score. R4 and R5 represent the comparisons between the above-mentioned two groups.

The features that characterize the situations after domain selection correspond to the information each expert returns to the central module after understanding the speech recognition results. These are listed in Table 3. Features listed from C1 to C3 represent a situation in which the previous domain (choice (I)) is selected. Those listed from C4 to C8 represent a situation in which a domain with the highest recognition score (choice (Ⅱ)) is selected.

Note that these features listed here have survived after feature selection. A feature survives if the performance in the domain classification is degraded when it is removed from a feature set one by one. We had prepared 32 features for the initial set.

Table 4: Specifications of each domain

Name of domain	Class of task	# of vocab. in ASR	# of slots
restaurant	database search	1,562	10
hotel	database search	741	9
temple	database search	1,573	4
weather	slot filling	87	3
bus	slot filling	1,621	3
total	-	7,373	-

5 Experimental Evaluation

5.1 Implementation

We implemented a Japanese multi-domain spoken dialogue system with five domain experts: restaurant, hotel, temple, weather, and bus. Specifications of each expert are listed in Table 4. If there is any overlapping slot between the vocabularies of the domains, our architecture can treat it as a common slot, whose value is shared among the domains when interacting with the user. In our system, place names are treated as a common slot.

We adopted Julian as the grammar-based speech recognizer (Kawahara et al., 2004). The grammar rules for the speech recognizer can be automatically generated from those used in the language understanding modules in each domain. As a phonetic model, we adopted a 3000-states PTM triphone model (Kawahara et al., 2004).

5.2 Collecting Dialogue Data

We collected dialogue data using a baseline system from 10 subjects. First, the subjects used the system by following a sample scenario, to get accustomed to the timing to speak. They, then, used the system by following three scenarios, where at least three domains were mentioned, but neither an actual temple name nor domain was explicitly mentioned. One of the scenarios is shown in Figure 5. Domain selection in the baseline system was performed on the basis of the baseline method that will be mentioned in Section 5.4, in which α was set to 40 after preliminary experiments.

In the experiments, we obtained 2,205 utterances (221 per subject, 74 per dialogue). The accuracy of the speech recognition was 63.3%, which was rather low. This was because the subjects tended to repeat similar utterances even after misrecognition occurred due to out-of-grammar or out-of-vocabulary utterances. Another reason was that the dialogues for subjects with worse speech recognition results got longer, which resulted in an increase in the total number of misrecognition.

> Tomorrow or the day after, you are planning a sightseeing tour of Kyoto. Please find a shrine you want to visit in the Arashiyama area, and determine, after considering the weather, on which day you will visit the shrine. Please, ask for a temperature on the day of travel. Also find out how to go to the shrine, whether you can take a bus from the Kyoto station to there, when the shrine is closing, and what the entrance fee is.

Figure 5: Example of scenarios

5.3 Construction of the Domain Classifier

We used the data containing 2,205 utterances collected using the baseline system, to construct a domain classifier. We used C5.0 (Quinlan, 1993) as a classifier. The features used were described in Section 4. Reference labels were given by hand for each utterance based on the domains the system had selected and transcriptions of the user's utterances, as follows[2].

Label (I): When the correct domain for a user's utterance is the same as the domain in which the previous system's response was made.

Label (Ⅱ): Except for case (I), when the correct domain for a user's utterance is the domain in which a speech recognition result in the N-best candidates with the highest score can be interpreted.

Label (Ⅲ): Domains other than (I) and (Ⅱ).

5.4 Evaluation of Domain Selection

We compared the performance of our domain selection with that of the baseline method described below.

Baseline method: A domain having an interpretation with the highest score in the N-best candidates of the speech recognition was selected, after adding α for the acoustic likelihood of the speech recognizer if the domain was the same as the previous one. We calculated the accuracies of domain selections for various α.

[2] Although only one of the authors assigned the labels, they could be easily assigned without ambiguity, since the labels were automatically defined as previously described. Thus, the annotator only needs to judge whether a user's request was about the same domain as the previous system's response or whether it was about a domain in the speech recognition result.

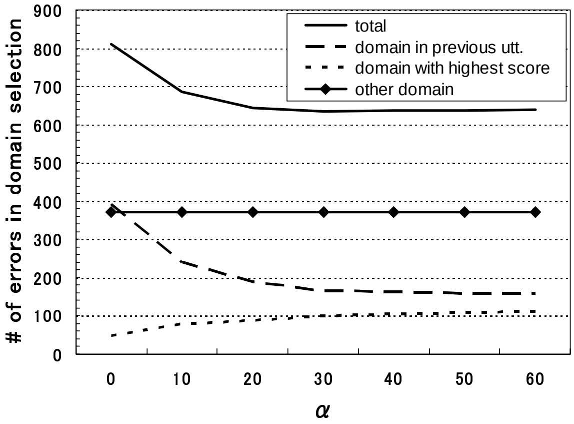

Figure 6: Accuracy of domain selection in the baseline method

Our method: A domain was selected based on our method. The performance was calculated with a 10-fold cross validation, that is, one tenth of the 2,205 utterances were used as test data, and the remainder was used as training data. The process was repeated 10 times, and the average of the accuracies was computed.

Accuracies for domain selection were calculated per utterance. When there were several domains that had the same score after domain selection, one domain was randomly selected among them as an output.

Figure 6 shows the number of errors for domain selection in the baseline method, categorized by their reference labels as α changed. As α increases, so does the system desire to keep the previous domain. A condition where $\alpha = 0$ corresponds to a method in which domains are selected based only on the speech recognition results, which implies that there are no constraints on keeping the current domain. As we can see in Figure 6, the number of errors whose reference labels are "a domain in the previous response (choice (I))" decreases as α gets larger. This is because incorrect domain transitions due to speech recognition errors were suppressed by the constraint to keep the domains. Conversely, we can see an increase in errors whose labels are "a domain with the highest speech recognition score (choice (Ⅱ))". This is because there is too much incentive for keeping the previous domain. The smallest number of errors was 634 when $\alpha = 35$, and the error rate of domain selection was 28.8% (= 634/2205). There were 371 errors whose reference labels were neither "a domain in the previous response" nor "a domain with the highest speech recognition score", which cannot be detected even when α is changed based on conventional frameworks.

We also calculated the classification accuracy of our method. Table 5 shows the results as a confusion matrix. The left hand figure denotes the number of outputs in the baseline method, while the right hand figure denotes the number of outputs in our method. Correct outputs are in the diagonal cells, while the domain selection errors are in the off diagonal cells. Total accuracy increased by 5.3%, from 71.2% to 76.5%, and the number of errors in domain selection was reduced from 634 to 518, so the error reduction rate was 18.3% (= 116/634). There was no output in the baseline method for "other domains (Ⅲ)", which is in the third column, because conventional frameworks have not taken this choice into consideration. Our method was able to detect this kind of error in 157 of 371 utterances, which allows us to prevent further errors from continuing. Moreover, accuracies for (I) and (Ⅱ) did not get worse. Precision for (I) improved from 0.77 to 0.83, and the F-measure for (I) also improved from 0.83 to 0.86. Although recall for (Ⅱ) got worse, its precision improved from 0.52 to 0.62, and consequently the F-measure for (Ⅱ) improved slightly from 0.61 to 0.62. These results show that our method can detect choice (Ⅲ), which was newly introduced, without degrading the existing classification accuracies.

The features that follow played an important role in the decision tree. The features that represent confidence in the previous domain appeared in the upper part of the tree, including "the number of affirmatives after entering the domain (P1)", "the ratio of user's negative answers in the domain (P9)", "the number of turns after entering the domain (P6)", and "the number of changed slots based on the user's utterances after entering the domain (P5)". These were also "whether a domain with the highest score has appeared before (C8)" and "whether an interpretation of a current user's utterance is negative (C2)".

6 Conclusion

We constructed a multi-domain spoken dialogue system using an extensible framework. Domain selection in conventional studies is based on either the domain based on the speech recognition

Table 5: Confusion matrix in domain selection (baseline / our method)

reference label \ output	in previous response (I)	with highest score (Ⅱ)	others (Ⅲ)	# total label (recall)
in previous response (I)	1289 / 1291	162 / 85	0 / 75	1451 (0.89 / 0.89)
with highest score (Ⅱ)	84 / 99	299† / 256†	0 / 28	383 (0.74 / 0.62)
others (Ⅲ)	293 / 172	78 / 42	0 / 157	371 (0 / 0.42)
total (precision)	1666 / 1562 (0.77) / (0.83)	539 / 383 (0.52) / (0.62)	0 / 260 (-) / (0.60)	2205 (0.712 / 0.765)

†: These include 17 errors because of random selection when there were several domains having the same highest scores.

results or the previous domain. However, we noticed that these conventional frameworks cannot cope with situations where neither of these domains is correct. Detection of such situations can prevent dialogues from staying in the incorrect domain, which allows our domain selection method to be robust against speech recognition errors. Furthermore, our domain selection method is also extensible. Our method does not select the domains directly, but, by categorizing them into three classes, it can cope with an increase or decrease in the number of domains. Based on the results of an experimental evaluation using 10 subjects, our method was able to reduce domain selection errors by 18.3% compared to a baseline method. This means our system is robust against speech recognition errors.

There are still some issues that could make our system more robust, and this is included in future work. For example, in this study, we adopted a grammar-based speech recognizer to construct each domain expert easily. However, other speech recognition methods could be used, such as a statistical language model. As well, multiple speech recognizers employing different domain-dependent grammars could be run in parallel. Thus, we need to investigate how to integrate these approaches into our framework, without destroying the extensibility.

References

Masahiro Araki, Kazunori Komatani, Taishi Hirata, and Shuji Doshita. 1999. A dialogue library for task-oriented spoken dialogue systems. In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–7.

Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR engine Julius and japanese model repository. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 3069–3072.

Kazunori Komatani, Naoyuki Kanda, Tetsuya Ogata, and Hiroshi G. Okuno. 2005a. Contextual constraints based on dialogue models in database search task for spoken dialogue systems. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pages 877–880, Sep.

Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2005b. User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.

Lori Lamel, Sophie Rosset, Jean-Luc Gauvain, and Samir Bennacef. 1999. The LIMSI ARISE system for train travel information. In *IEEE Int'l Conf. Acoust., Speech & Signal Processing (ICASSP)*, pages 501–504, Phoenix, AZ.

Ian R. Lane and Tatsuya Kawahara. 2005. Utterance verification incorporating in-domain confidence and discourse coherence measures. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pages 421–424.

E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E. Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker. 2000. The AT&T-DARPA communicator mixed-initiative spoken dialogue system. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*.

Bor-shen Lin, Hsin-min Wang, and Lin-shan Lee. 2001. A distributed agent architecture for intelligent multi-domain spoken dialogue systems. *IEICE Trans. on Information and Systems*, E84-D(9):1217–1230, Sept.

Mikio Nakano, Yuji Hasegawa, Kazuhiro Nakadai, Takahiro Nakamura, Johane Takeuchi, Toyotaka Torii, Hiroshi Tsujino, Naoyuki Kanda, and Hiroshi G. Okuno. 2005. A two-layer model for behavior and dialogue planning in conversational service robots. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1542–1548.

Ian O'Neill, Philip Hanna, Xingkun Liu, and Michael McTear. 2004. Cross domain dialogue modelling: An object-based approach. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*.

Botond Pakucs. 2003. Towards dynamic multi-domain dialogue processing. In *Proc. European*

Conf. Speech Commun. & Tech. (EUROSPEECH), pages 741–744.

Alexandros Potamianos and Hong-Kwang J. Kuo. 2000. Statistical recursive finite state machine parsing for speech understanding. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, volume 3, pages 510–513.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA. http://www.rulequest.com/see5-info.html.

Antoine Raux and Maxine Eskenazi. 2004. Non-native users in the let's go!! spoken dialogue system: Dealing with linguistic mismatch. In *Proc. of HLT/NAACL*.

Ruben San-Segundo, Bryan Pellom, Wayne Ward, and Jose M. Pardo. 2000. Confidence measures for dialogue management in the CU communicator system. In *IEEE Int'l Conf. Acoust., Speech & Signal Processing (ICASSP)*.