DIALOGUE MANAGEMENT USING CONCEPT-LEVEL CONFIDENCE MEASURES OF SPEECH RECOGNITION

Tatsuya Kawahara *Kazunori Komatani* *Shuji Doshita*

School of Informatics, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan

Ryukoku University
Otsu, 520-2194, Japan

ABSTRACT

We present a method to generate effective confirmation and guidance using concept-level confidence measures (CM) derived from speech recognizer output in order to handle speech recognition errors. We define two concept-level CM, which are on content-words and on semantic-attributes, using 10-best outputs of the speech recognizer and parsing with phrase-level grammars. Content-word CM is useful for selecting plausible interpretations. Less confident interpretations are given to confirmation process, and non-confident ones are rejected. The strategy improved the interpretation accuracy by 11.5%. Moreover, the semantic-attribute CM is used to estimate user's intention and generates system-initiative guidances even when successful interpretation is not obtained. We also introduce design and implementations of domain-independent spoken dialogue interfaces for information query.

1. INTRODUCTION

In a spoken dialogue system, it frequently occurs that the system incorrectly recognizes user utterances and the user makes expressions the system has not expected. These problems are essentially inevitable in handling the natural language by computers, even if vocabulary and grammar of the system are tuned. Namely, the system must behave appropriately even when speech recognizer output contains some errors.

Obviously, making confirmation is effective to avoid misunderstandings caused by speech recognition errors. However, when confirmations are made for every utterance, the dialogue will become too redundant and consequently troublesome for users. Previous works have shown that confirmation strategy should be decided according to the frequency of speech recognition errors, using mathematical formula [1] and using computer-to-computer simulation [2]. These works assume fixed performance (averaged speech recognition accuracy) in whole dialogue with any speakers. For flexible dialogue management, however the confirmation strategy must be dynamically changed based on the individual utterances. For instance, we human make confirmation only when we are not confident. Similarly, confidence measures (CM) of speech recognition output should be modeled as a criterion to control dialogue management.

In this paper, we propose two concept-level CM that are on content-word level and on semantic-attribute level for every content word. The system can make efficient confirmation and effective guidance according to the CM. Even when successful interpretation is not obtained on content-word level, the system generates system-initiative guidances based on the semantic-attribute level, which lead the next user's utterance to successful interpretation.

2. DEFINITION OF CONFIDENCE MEASURES (CM)

Confidence Measures (CM) have been studied for utterance verification that verifies speech recognition result as a post-processing [3]. Since an automatic speech recognition is a process finding a sentence hypothesis with the maximum likelihood for an input speech, some measures are needed in order to distinguish a correct recognition result from incorrect ones.

2.1. Definition of CM for Content Word

We use a grammar-based speech recognizer Julian, which was developed in our laboratory. It correctly obtains the N-best candidates and their scores by using A* search algorithm.

Using the scores of these N-best candidates, we calculate content-word CM as below. A score of each sentence output by the recognizer is a log-scaled likelihood. The content words are extracted by parsing with phrase-level grammars that are used in speech recognition process. In this paper, we set $N = 10$ after we examined various values of N as the number of generated candidates.

First, each i-th score is multiplied by a factor $\alpha (\alpha < 1)$. This factor smoothes the difference of N-best scores to get adequately distributed CM. Next, they are transformed from log-scaled value $(\alpha \cdot scaled_i)$ to probability dimension by taking its exponential, and calculate a posteriori proba-

utterance: "*oosakafu no singururyoukin ga 19000 en no yado*"
("Tell me hotels in Osaka-pref. less than 19000 yen for a single room.")

i	Recognition candidates (<g>: filler model)	$score_i$	p_i
1	*oosakafu no singururyoukin ga 19000 en ika no* <g> Osaka-pref.(location) / less than 19000 yen for a single room	-16490	.15
2	*oosakahu no singuruyoukin ga 19000 en ika no yado* Osaka-pref.(location) / less than 19000 yen for a single room	-16493	.13
3	*oosakafu no singururyoukin ga 12000 en ika no* <g> Osaka-pref.(location) / less than 12000 yen for a single room	-16495	.12
4	*oosakafu no singururyoukin ga 18000 en ika no* <g> Osaka-pref.(location) / less than 18000 yen for a single room	-16496	.11
5	*oosakafu no singururyoukin no 12000 en ika no yado* Osaka-pref.(location) / less than 12000 yen for a single room	-16498	.10
6	*oosakafu no singururyoukin ga 14000 en ika no* <g> Osaka-pref.(location) / less than 14000 yen for a single room	-16498	.10
7	*oosakafu no singururyoukin ga 18000 en ika no yado* Osaka-pref.(location) / less than 18000 yen for a single room	-16500	.09
8	*oosakafu no singururyoukin no 16000 en ika no* <g> Osaka-pref.(location) / less than 16000 yen for a single room	-16501	.09
9	*oosakafu no singururyoukin no 14000 en ika no yado* Osaka-pref.(location) / less than 14000 yen for a single room	-16502	.08
10	*oosakashi no singururyoukin no 19000 en ika no* <g> Osaka-city.(location) / less than 19000 yen for a single room	-16518	.04

CM_w	(word)@(attribute)
0.96	Osaka-pref.@location
0.31	19000yen@single:max
0.22	12000yen@single:max
0.20	18000yen@single:max
0.18	14000yen@single:max
0.09	16000yen@single:max
0.04	Osaka-pref.@location

CM_c	semantic attribute
1.00	single:max
0.50	location

Figure 1: Example of calculating CM

bility for each i-th candidate [4].

$$p_i = \frac{e^{\alpha \cdot scaled_i}}{\sum_{j=1}^{n} e^{\alpha \cdot scaled_j}}$$

If the i-th sentence contains a word w, let $\delta_{w,i} = 1$, and 0 otherwise. A posteriori probability that a word w is contained (p_w) is derived as summation of a posteriori probabilities of sentences that contain the word.

$$p_w = \sum_{i=1}^{n} p_i \cdot \delta_{w,i}$$

We define this p_w as the content-word CM (CM_w). This CM_w is calculated for every content word. Intuitively, words that appear many times in N-best hypotheses get high CM, and frequently substituted ones in N-best hypotheses are judged as unreliable.

In Figure 1, we show an example in CM_w calculation with recognizer outputs (i-th recognized candidates and their a posteriori probabilities) for an utterance "*oosakafu no singururyoukin ga 19000 en ika no yado* (Tell me hotels in Osaka-pref. less than 19000 yen for a single room.)". It is observed that a correct content word 'restaurant as facility' gets a high CM value ($CM_w = 1$). The others, which are incorrectly recognized, get low CM, and shall be rejected.

2.2. CM for Semantic Attribute

A concept category is semantic attribute assigned to content words, and it is identified by parsing with phrase-level grammars that are used in speech recognition process and represented with Finite State Automata (FSA). In our hotel query task, there are seven concept categories such as 'location' and 'facility'.

For this concept category, we also define semantic-attribute CM (CM_c). Here, we introduce $\beta_{c,i}$ representing likelihood that a phrase in i-th sentence belongs to a category c. We define $\beta_{c,i}$ by the summation of idf (inverse document frequency) values of the content words in a phrase. ($idf_j = \log(N/df_j)$, where N is the number of all categories and df_j is number of categories that contain word j.)

$$\beta_{c,i} = \sum_j idf_j (= \sum (\log N/df_j))$$

Then, it is normalized by the expected value for each category, and rewritten as $\beta^*_{c,i}$. If a concept category c is contained in the i-th sentence, let $\delta_{c,i} = 1$, and 0 otherwise. The semantic-attribute CM (CM_c) is defined as below.

$$CM_c = \sum_{i=1}^{n} p_i \cdot \beta^*_{c,i} \cdot \delta_{c,i}$$

This CM_c estimates which category the user refers to and is used to generate effective guidances.

3. DIALOGUE MANAGEMENT USING CONFIDENCE MEASURES

3.1. Making Effective Confirmations

Confidence Measure (CM) is useful in selecting reliable candidates and controlling confirmation strategy. By setting two thresholds $\theta_1, \theta_2 (\theta_1 > \theta_2)$ on content-word CM (CM_w), we adopt the confirmation strategy as follows.

1. $CM_w > \theta_1$ $\rightarrow$ accept the hypothesis
2. $\theta_1 \geq CM_w > \theta_2$ $\rightarrow$ make confirmation to the user "Did you say ...?"
3. $\theta_2 \geq CM_w$ $\rightarrow$ reject the hypothesis

Because CM_w is defined for every content word, judgment among acceptance, confirmation, or rejection is made for every content word when one utterance contains several content words. Only if all content words are rejected, the system will prompt the user to utter again. By accepting apparently correct words and rejecting unreliable candidates, this strategy focuses on only indistinct candidates and avoids redundant confirmations. These thresholds θ_1, θ_2 are optimized considering the false acceptance (FA) and the false rejection (FR) using real data.

3.2. Generating System-Initiated Guidances

The system-initiated guidances are effective when recognition does not go well. Even when any successful output of content words is not obtained, the system can generate effective guidances based on the semantic attribute with high confidence. For example, if all the 10-best candidates are concerning a name of place but their CM_w values are lower than the threshold (θ_2), any word will be neither accepted nor confirmed. In such a case, rather than rejecting the whole sentence and telling the user "Please say again", it is better to guide the user based on the attribute having high CM_c, such as "Which city is your destination?". This guidance enables the system to narrow down the vocabulary of the next user's utterance and to reduce the recognition difficulty. It will consequently lead next user's utterance to successful interpretation.

4. EXPERIMENTAL EVALUATION

4.1. Task and Data

We evaluate the strategy on the hotel query task. We collected 120 minutes speech data by 24 novice users by using the prototype system with GUI [5]. The data is segmented into 705 utterances with a pause of 1.25 seconds. The vocabulary of the system contains 982 words, and the number of database records is 2040.

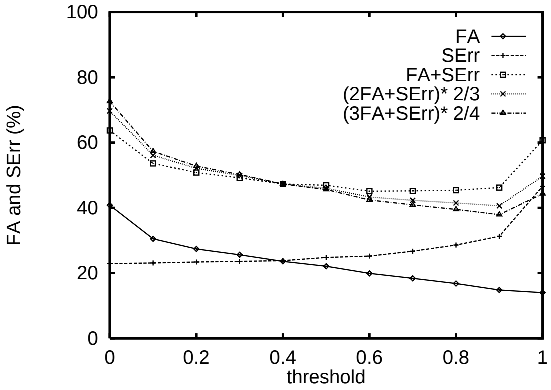

Figure 2: FA+SErr for deciding θ_1

Out of 705 utterances, 124 utterances (17.6%) are beyond the system's capability, namely they are out-of-vocabulary, out-of-grammar, out-of-task, or fragment of utterance. In the following experiments, we evaluate the system performance using all data including these unacceptable utterances in order to evaluate how the system can reject unexpected utterances appropriately as well as recognize regular utterances correctly.

4.2. Optimization of Thresholds

We optimize two threshold values that provide the confirmation strategy using the collected data. We count errors not by the utterance but by the content-word (slot). The number of slots to be filled is 804.

The threshold θ_1 decides between acceptance and confirmation. The value of θ_1 should be determined considering both the ratio of incorrectly accepting recognition errors (False Acceptance; FA) and the ratio of slots that are not filled with correct values (Slot Error; SErr). Namely, FA and SErr are defined as the complements of precision and recall rate of the output, respectively.

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

$$SErr = 1 - \frac{\text{\# of correctly accepted words}}{\text{\# of all correct words}}$$

We weight the FA because accepting an error damages the dialogue worse than rejecting a correct answer. By minimizing this weighted loss function (wFA+SErr), we derive a value of θ_1 as 0.9 (see Figure. 2).

Similarly, the threshold θ_2 decides between confirmation and rejection. The value of θ_2 should be decided considering both the ratio of incorrectly rejecting content words (False Rejection; FR) and the ratio of accepting recognition errors into the confirmation process (condi-

Table 1: Comparison of methods

	FA+SErr	FA	SErr
only 1st candidate	51.5	27.6	23.9
no confirmation	46.1	14.8	31.3
with confirmation	40.0	14.8	25.2

FA: ratio of incorrectly accepting recognition errors
SErr: ratio of slots that are not filled with correct values

tional False Acceptance; cFA).

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

By minimizing FR+cFA, we derive a value of θ_2 as 0.6.

4.3. Comparison with Conventional Methods

In many conventional spoken dialogue systems, only 1-best candidate of a speech recognizer output is used in the subsequent processing. We compare our method with the conventional method that uses only 1-best candidate (Table 1).

In the 'no confirmation' strategy, the hypotheses are classified by a single threshold (θ) into either accepted or rejected. In this case, a threshold value of θ is set to 0.9 that gives minimum FA+SErr. In the 'with confirmation' strategy, we set $\theta_1 = 0.9$ and $\theta_2 = 0.6$. The 'FA+SErr' in Table 1 means FA(θ_1)+SErr(θ_2), on the assumption that the confirmed phrases are correctly accepted or rejected. As shown in Table 1, the interpretation accuracy is improved by 5.4% by the 'no confirmation' strategy compared with the conventional method. And 'with confirmation' strategy, we achieve 11.5% improvement in total. This result proves that our method successfully eliminates recognition errors.

By making confirmation, the interaction becomes robust, but accordingly the number of whole utterances increases. If all candidates having CM_w under θ_1 are given to confirmation process without setting θ_2, 332 vain confirmation for incorrect contents are generated out of 400 candidates. By setting θ_2, 102 candidates having CM_w between θ_1 and θ_2 are confirmed, and the number of incorrect confirmations is suppressed to 53. Namely, the ratio of correct hypotheses and incorrect ones being confirmed are almost equal. This result shows only indistinct candidates are given to confirmation process whereas unreliable candidates are rejected.

4.4. Effectiveness of Semantic-Attribute CM

In Figure 3, the performance of content-word CM and semantic-attribute CM is shown. Each CM is evaluated by the weighted sum such as '3FA+SErr'. It is observed that semantic-attribute CM is estimated more correctly

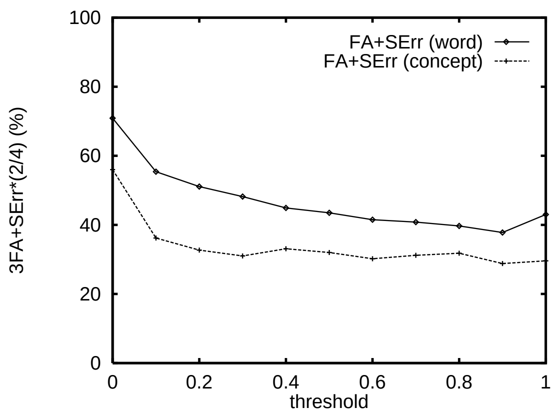

Figure 3: Performance of word CM and category CM

than content-word CM. This fact suggests that semantic-attribute can be estimated correctly even when successful interpretation is not obtained from content-word CM.

In the test data, there are 148 slots[1] that are not obtained correctly by content-word CM. For these slots, we can generate guidance with $CM_c = 1$ in 90% (9/10) accuracy. And by making confirmation for the slots having CM ($1.0 > CM_c \geq 0.5$) like "Are you saying about price?", guidances are generated for 16% (24/148) utterances that had been only rejected in conventional methods.

5. DOMAIN-INDEPENDENT PLATFORM OF SPOKEN DIALOGUE INTERFACES

We have developed a domain-independent platform of spoken dialogue systems for their rapid prototyping. The prototyped system can be used as a baseline for data collection or task analysis.

In order to realize domain-independent features, the platform assumes that the system performs information query using multi-modal interface.

The domain of information query is not limited, but includes trains, flights and hotels. Since the query is generally made of sets of keys and values, the platform automatically generate a lexicon and grammar rules, which mainly consist of keyword values for the query and typical expressions used in the query. They are derived from the target database, i.e. database of trains or hotels. Overview of the platform and system is depicted in Figure 4.

5.1. Use of GUI

Use of multi-modal interface with a display eases the problems of spoken language understanding and dialogue management. One of the major problems in speech understanding is out-of-vocabulary or out-of-grammar expressions.

[1]Out-of-vocabulary and out-of-grammar utterances are included .

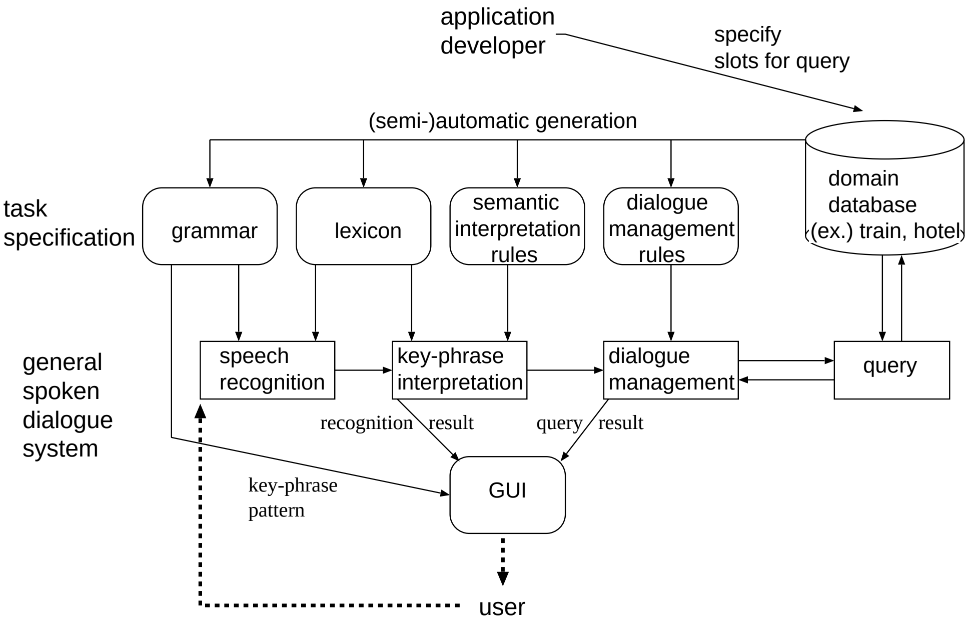

Figure 4: Overview of platform and system

On the other hand, we have observed that users often hesitate to speak to machines simply because they do not know which forms of expressions are acceptable. Our system displays key-phrase patterns as a visual form of the query, which guides the users how to utter and reduces the variation of input utterances. Moreover, the system promptly displays the recognition results in the slot of the visual query form as well as the query results. The feature let the users know the recognition errors and eliminates the necessity of confirmation through spoken dialogue. Instead, the users simply make "undo" commands in case of errors. This feature will avoid the possible crash in dialogue. An outlook of GUI screen is shown in Figure 5.

5.2. Implementation Issues

Several implementation issues must be taken into account to realize these capability.

User utterances can be any combination of key-phrase patterns for task slots and may contain disfluencies. Moreover, there is no data for training statistical language models when designing prototyping systems. To cope with the problem, we have introduced the key-phrase detection and verification strategy which extracts semantically tagged key-phrases directly from unconstrained speech[3].

In many tasks, semantic interpretation of utterances demands pragmatic knowledge sources. For example, a slot value of "tomorrow" must be transformed into an actual date. Therefore, the platform includes several pragmatic concept necessary for typical information query.

Although tasks of information query are generally simple, dialogue management rules are needed to solve ambiguity in interpretation. These rules are also automatically derived by predicting possible cases of ambiguity in the task. They are classified into two. One is the case where the same type of values (e.g. date) can appear in two different slots (e.g. departure and arrival). The other is caused by different word entries of the same pronunciation, which are common in Japanese.

5.3. Application and Evaluation

We have implemented the platform as a toolkit that includes a speech recognizer and GUIs.

First, we have applied and evaluated in the hotel query task. It searches for hotel accommodations that satisfy users' request. A prototype system is set up by specifying what kinds of slots (e.g. location or room rates) are necessary in the query. The lexicon and grammar are semi-automatically derived. The grammar is a set of simple key-phrase patterns. We have confirmed that combination of displaying the key-phrase patterns in GUI and our key-phrase detection approach significantly improves the coverage on user utterances compared with the conventional system that adopts a complex FSA grammar, while reducing time and labor cost in writing and tuning the grammar. We have also confirmed the effect of interactively displaying recognition and query results. In total, it improves the success ratio of the query.

The platform is also applied to an interface of our literature search engine on speech processing [2]. As the bibliography database is updated quite often and contains a

[2] http://winnie.kuis.kyoto-u.ac.jp/bibliography/search_e.html

(a) A real system in Japanese (b) Upper portion translated in English

Figure 5: An outlook of GUI screen

lot of entries, our toolkit which automatically generates the lexicon and finds ambiguity in names of the same pronunciation is useful.

6. CONCLUSION

We have addressed the use of conficence measures of speech recognition results for dialogue management. It leads to generation of effective confirmation and guidance, especially in combination with the estimation of concept-level attributes.

Based on this feature, we have developed a spoken dialogue system and evaluated with the novice users. It is confirmed that the use of confidence measures together with GUI is effective for robust spoken dialogue interfaces.

References

[1] Y. Niimi and Y. Kobayashi, "A dialog control strategy based on the reliability of speech recognition," in *Proc. Int'l Conf. on Spoken Language Processing*, 1996.

[2] T. Watanabe, M. Araki, and S. Doshita, "Evaluating dialogue strategies under communication errors using computer-to-computer simulation," *Trans. of IEICE, Info & Syst.*, vol. E81-D, no. 9, pp. 1025–1033, 1998.

[3] T.Kawahara, C.-H.Lee, and B.-H.Juang. Flexible speech understanding based on combined key-phrase detection and verification. *IEEE Trans. Speech & Audio Process.*, 6(6):558–568, 1998.

[4] G. Bouwman, J. Sturm, and L. Boves, "Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project," in *Proc. ICASSP*, 1999.

[5] T.Kawahara, K.Tanaka, and S.Doshita. Domain-independent platform of spoken dialogue interfaces for information query. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems*, pages 69–72, 1999.

[6] K.Komatani and T.Kawahara. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pp. 467–473, 2000.