Efficient Dialogue Strategy to Find Users' Intended Items from Information Query Results

Kazunori Komatani Tatsuya Kawahara Ryosuke Ito Hiroshi G. Okuno
Graduate School of Informatics, Kyoto University
Kyoto 606-8501, Japan
{komatani, kawahara, rito, okuno}@kuis.kyoto-u.ac.jp

Abstract

We address a dialogue framework that narrows down the user's query results obtained by an information retrieval system. The follow-up dialogue to constrain query results is significant especially with the speech interfaces such as telephones because a lot of query results cannot be presented to the user. The proposed dialogue framework generates guiding questions based on an information theoretic criterion to eliminate retrieved candidates by a spontaneous query without assuming a semantic slot structure. We first describe its concept on general information query tasks, and then deal with a query task on the appliance manual where structured task knowledge is available. A *hierarchical confirmation strategy* is proposed by making use of a tree structure of the manual, and then three cost functions for selecting optimal question nodes are compared. Experimental evaluation demonstrates that the proposed system helps users find their intended items more efficiently.

1 Introduction

In the past years, a great number of spoken dialogue systems have been developed. Their typical task domains include airline information (Levin et al., 2000; Potamianos et al., 2000; San-Segundo et al., 2000) and train information (Allen et al., 1996; Bennacef et al., 1996; Sturm et al., 1999; Lamel et al., 1999). Most of them model speech understanding process as converting recognition results into semantic representations equivalent to database query (SQL) commands, and dialogue process as disambiguating their unfixed slots. Usually, the semantic slots are defined a priori and manually. The approach is workable only when data structure of the application is well-organized typically as a relational database (RDB).

Different and more flexible approach is needed for spoken dialogue interfaces to access information described in less rigid format, in particular normal text database. For the purpose, information retrieval (IR) technique is useful to find a list of matching documents from the input query. Typically, keywords are extracted from the query and statistical matching is performed. Call routing task (Chu-Carroll and Carpenter, 1998) can be regarded as the special case.

In IR systems, many candidates are usually obtained as a query result, thus there is a significant problem of how to find the user's intended item among them. Especially, either on the telephone or electrical appliances, there is not a large screen displaying the candidates, and all the query results cannot be presented to a user. So it is desirable for the system to narrow down the query results interactively. Moreover, interactive query is more friendly to novice users rather than requiring them to input a detailed query from the beginning.

In this paper, we address a dialogue strategy to find the user's intended item from the retrieved result, which is initiated by a spontaneous query utterance. In section 2, we describe a method to generate a guiding question that narrows down the query results efficiently, using an example of a restaurant query task. The question is selected based on an information theoretic criterion. In section 3, we present a dialogue management method for a query task on the appliance manual where structured task knowledge is available. We propose a confirmation strategy by making use of a tree structure of the manual, and define three cost functions for selecting question nodes. The method is evaluated by the number of average dialogue turns.

Although there are previous studies on optimizing dialogue strategies (Niimi and Kobayashi, 1996; Levin et al., 1997; Litman et al., 2000), most of them assume the tasks of filling semantic slots that are definitely and manually defined, and few focus on follow-up dialogue of information retrieval. For example, (Denecke, 1997) proposed a method to generate guiding questions by making use of a tree structure constructed by unifying retrieved items based on semantic slots. In this paper, we do not assume any structure of semantic slots. Instead, we make use of distribution of document statistics or a structure of task knowledge. We also investigate cost functions for optimal dialogue control by taking into account of speech recognition errors.

2 Dialogue Strategy in General Information Query Task

Interaction in an information query task can be regarded as a process seeking a common part between the user's request and system knowledge. In order to help users to find their intended items from the system knowledge, the system has to carry out not only interpreting what users say but also showing the relevant portion of the system knowledge to them.

We assume that users freely set and retract query keys based on their preference for information query systems. If many candidates still remain even after specifying all possible his/her preference to the system, users may have difficulty in narrowing down further the query result. Thus, the system should generate efficient guiding questions to help users find their intended items.

In this section, we presume the system knowledge as a pair of an item and a set of keywords (Figure 1). We define keywords as a set of words representing contents of the items, and their categories such as place, food and so on are given. This is similar to indexing words in a conventional information retrieval task. Note that it is not needed that the system knowledge is structured like an RDB.

Restaurant A
    Chinese noodles, meat dumpling,
    Shinjuku, Kabukicho, Ekoda

Restaurant B
    Chinese noodles, meat dumpling,
    Shinjuku, Kabukicho,

Restaurant C
    Chinese noodles, meat dumpling,
    noodles with boiled-pork-ribs,
    Takadanobaba

Restaurant D
    Chinese noodles, fried garlic, Yebisu

...

Figure 1: An example of system knowledge

Keywords are extracted from a user's utterance, and are matched with the system knowledge. Here, we adopt the following matching function for each item j.

$$L_j = \sum_{i \in K_j} \left(CM_i * \log \frac{N}{df_i} \right)$$

Here, K_j is a set of keywords for item j. CM_i is a confidence measure of speech recognition for keyword i (Komatani and Kawahara, 2000), N is the total number of items, and df_i is the number of items including keyword i. Intuitively, keyword that is recognized with high confidence and does not appear in many items gets higher likelihood L_j by CM_i and df_i, respectively.

Then, we define amount of information that is obtained when the system generates yes/no question and the user answers it. Here, C is a current query condition, A is a condition that is added by the system's question, and $count(x)$ is the number of items that satisfy the condition x. The condition consists of the conjunction of the keywords the user specified. Suppose each item occurs by equal likelihood, the following equation denotes the likelihood $p'(A_{yes})$ that the yes/no question corresponding to the adding condition A will be answered as "yes".

$$p'(A_{yes}) = \frac{count(C \cap A)}{count(C)}$$

We weight on each item j with the likelihood L_j.

$$p(A_{yes}) = \frac{\sum_{j \in \{C \cap A\}} L_j}{\sum_{j \in \{C\}} L_j}$$

The amount of information that is obtained when the user's answer is "yes" is represented as follows.

$$I(A_{yes}) = \log_2 \frac{1}{p(A_{yes})}$$

The following equation gives $H(A)$, the expected value of amount of information that is obtained by generating a question about condition A and getting user's answer ("yes" or "no").

$$H(A) = \sum_{x \in \{yes,no\}} p(A_x) \log_2 \frac{1}{p(A_x)}$$

By calculating $H(A)$ for all conditions A that can be added to the current query condition, the system generates the question that has the maximum value of $H(A)$. The question is generated using the category information of each keyword.

Because the obtained condition A is selected by a viewpoint of narrowing down the current set of items efficiently, the selected condition may be unimportant for the user. In such a case, it is not cooperative to force the user an affirmative or negative reply. Our system does not force the reluctant decision by allowing the user to say "It does not matter anyhow.". Instead, the system presents the second best proposal.

We explain the method with the following example in our restaurant query system in the Tokyo area. When a user says, "Please tell me a restaurant where I can eat Chinese noodle and meat dumpling in Shinjuku area.", three keywords are extracted: "Shinjuku", "Chinese noodle" and "meat dumpling". As a result of the matching using these three keywords, 11 query results are obtained. It is not cooperative to read out all of the 11 query results with a TTS (text-to-speech) system. Here, the expected values of amount of information $H(A)$ are calculated for each condition that corresponds to keywords included in the matched items except for the three keywords, "Shinjuku", "Chinese noodle" and "meat dumpling". Then, we select the keyword "noodles with boiled-pork-ribs" that has the maximum value $H(A)$. By generating a question like "Would you like one which serves noodles with boiled-pork-ribs?", and obtaining a reply from the user, the system adds the new condition and narrows down the candidates efficiently. If the user thinks that the condition "noodles with boiled-pork-ribs" is not important and tells the system so (for example "Either will do."), the system can show the second best proposal, "Would you like one located in Kabukicho area?". Thus, the query result can be narrowed down without forcing the user unnatural yes/no answers.

3 Dialogue Strategy for Query on Appliance Manuals

In this section, we present another efficient solution in the case that the structure or hierarchy of task knowledge is available. The task here is to find the appropriate item in the manual of electric appliances with a spoken dialogue interface. Such an interface will be useful as the recent appliances become complex with many features and so are their manuals. In the appliances such as VTR (Video Tape Recorder) and FAX machines, there is not a large screen to display the list of matched candidates to be selected by the user. Therefore, we address a spoken dialogue strategy to determine the most appropriate one from the list of candidates.

An alternative system design is the use of directory search, as adopted in voice portal systems, where the documents are hierarchically structured and the system prompts users to select one of the menu from the top to the leaf. The method is rigid and not user-friendly since users often have trouble in selection and want to specify by their own expression. The proposed system allows users to make queries spontaneously and makes use of the directory structure in the follow-up dialogue to determine the most appropriate one.

3.1 System Overview

An overview of the system is illustrated in Figure 2. It consists of following processes.

1. Keyword spotting from user utterances using an ASR (automatic speech recognition) system (Kawahara et al., 1998)

   A natural spoken language query is accepted and keywords are extracted. A confidence measure CM_i is assigned to each keyword i based on the N-best recognition result (Komatani and Kawahara, 2000).

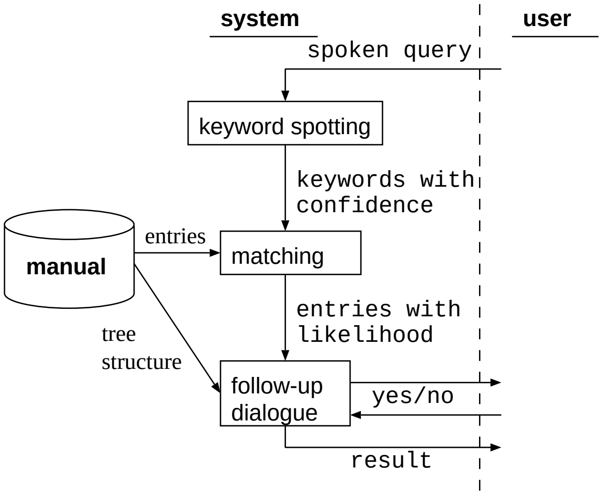

Figure 2: System overview

2. Matching with manual items (documents)

   The extracted keywords are matched with a set of manual items. The matching is performed on the initial portion (index and first summary paragraph) of each manual section. We adopt the following matching score function for an item j. K_j is a set of keywords for item j.

$$L_j = \frac{1}{n_j} \sum_{i \in K_j} (CM_i * \log \frac{N}{df_i})$$

   Here, df_i is the number of items that contain keyword i referred as a document frequency and N is the total number of items. The inverse document frequency (idf) is weighted with a confidence measure CM_i and summed over keywords, then normalized by n_j, the number of keywords in the item j.

3. Generating dialogue to determine the most appropriate one from the list of candidates

   As a result of the matching, many candidates are usually found. They may include irrelevant ones because of speech recognition errors. But it is not practical to read out all of them in order with a TTS (text-to-speech) system. Therefore, dialogue is invoked to narrow down to the intended one. This dialogue is restricted to system-initiated "yes/no" questions in order to avoid further recognition errors and back-up dialogue. The dialogue strategy is explained in the next subsection.

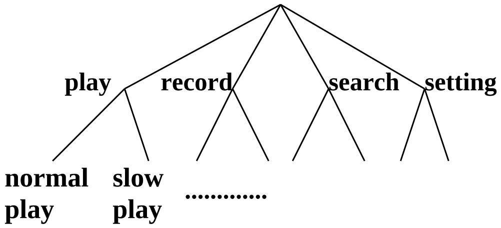

Figure 3: Example of tree structure of manual

3.2 Dialogue Strategy using Structure of Manual

If one of the candidates is more plausible than others with a significant margin, we should make confirmation on it. When there are many candidates with similar confidence and they can be hierarchically grouped into several categories, we had better first identify which category the intended one belongs to. In this work, we make use of the section structure of the manual, i.e. section is the first layer, sub-section is the second-layer, and so on. The tree structure is automatically derived from its table of contents. An example for VTR manual is shown in Figure 3.

For each node of the tree, likelihood L'_j is assigned as follows.

- For a leaf node, the matching score L_j is assigned after normalizing so that the sum over all leaves (manual items) is 1.0.
- For a non-leaf node, the sum of the likelihood of its children nodes is assigned.

Then, a dialogue is generated as follows.

1. Among ancestor nodes of the leaf of the largest likelihood L'_j, pick up the one whose heuristic cost function described below is smallest.
2. Make a "yes/no" question on the node, for example "Do you want to know about ...?". The content of the question is associated with the section title.
3. If the user's answer is "yes", eliminate the nodes other than descendants of the con-

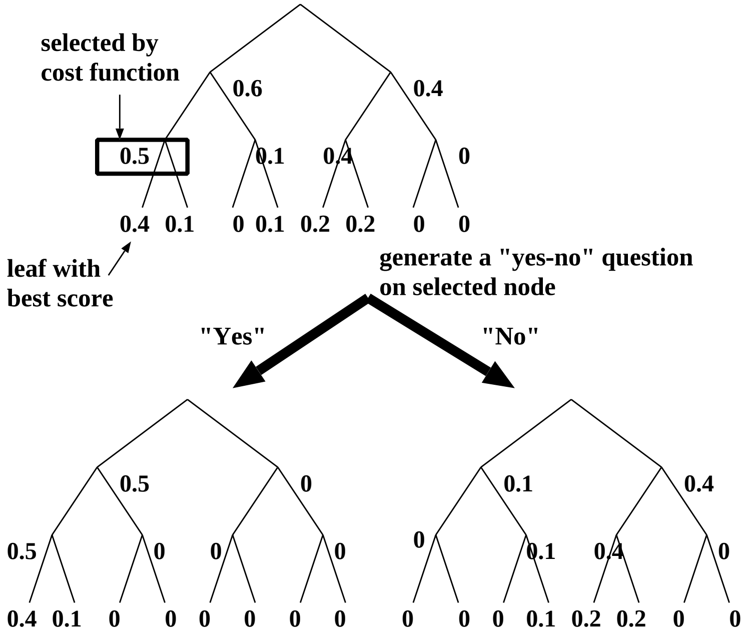

Figure 4: Use of manual structure and cost function for dialogue control

firmed node. If the answer is "no", eliminate all descendants of the denied node.

4. Repeat the process until only one node (or less than a threshold α) remains.

The above processes are illustrated in Figure 4.

We define following three heuristic cost functions in order to realize an efficient dialogue.

- $h_1(j) = |L'_j - 0.5|$

  This makes a question on the most ambiguous node whose likelihood L'_j interpreted as a posteriori probability is close to 0.5.

- $h_2(j) = L'_j * Node_j(yes) + (1 - L'_j) * Node_j(no)$

  Here, $Node_j$ is the number of remaining nodes when the answer is "yes" or "no". This function takes the approximate number of following questions into account.

- $h_3(j) = L'_j * Ques_j(yes) + (1 - L'_j) * Ques_j(no) + 1$

  $Ques_j$ is the estimated number of times of questions needed when the answer is "yes" or "no". It is computed recursively by expanding the sub-tree, and is assigned with 0 when the number of remaining nodes is under a threshold (α). This α means the number of candidates that can be presented to users. Here, we set $\alpha = 3$.

These are experimentally compared in the next subsection.

3.3 Experimental Evaluation

3.3.1 Task and System Implementation

The proposed system is implemented for the query task on a VTR manual that consists of 111 pages and 47 items. The derived tree structure is of three levels. The number of keywords used for matching is 137.

The speech recognition system is based on our large vocabulary continuous speech recognition engine Julius (Lee et al., 2001). The language model is initially based on a finite state grammar and extended to combine statistical models derived from the domain-specific corpus (Komatani et al., 2001), that is the manual text in this task. The acoustic model is a gender-dependent phonetic tied-mixture (PTM) triphone model (Lee et al., 2000) trained with the 40-hour JNAS speech corpus.

For collecting evaluation data, we had 14 subjects and each made 10 queries on given scenarios (query sentences are not given), and several spontaneous queries without any scenarios. In total, we had 195 query utterances, of which

Table 1: Evaluation result with text input

# matched candidates	12.4		
query success rate	93%		
average rank of correct item (# turns by baseline)	3.2		
# turns by	h_1	h_2	h_3
proposed cost functions	2.4	2.5	2.8

Table 3: Evaluation result with speech input

# matched candidates	13.3		
query success rate	87%		
average rank of correct item (# turns by baseline)	4.1		
# turns by	h_1	h_2	h_3
proposed cost functions	2.9	2.9	3.2

157 could be coped with the given manual, thus used as the test-set. Sample queries are "I want to change the recording reservation." and "Can I watch TV while recording another program?"

As for evaluation measures, we first compute the rate of query success where the correct manual item is contained in the candidate list by the initial matching. Then, the system is evaluated by the necessary dialogue turns equivalent to the number of questions before the correct item is identified. It is compared with the baseline case where the candidates are presented to the user in order of the matching score L_j and the number of dialogue turns is equivalent to the rank of the correct item.

3.3.2 Evaluation with Text Input

At first, the system is evaluated with text input, which is transcription of the collected queries. The result is shown in Table 1.

On the average, the matching result consists of 12.4 candidates and contains correct one for 93% of the tractable queries. The average rank of the correct item is 3.2, which means, if we make confirmation in order of the matching score L_j, we need 3.2 turns on the average. With dialogue based on the heuristic cost functions, it can be reduced to 2.4 (h_1), 2.5 (h_2) and 2.8 (h_3), respectively.

We have not yet identified the reason why performance by the apparently most accurate function h_3 is not good. We conjuncture that the difference of the cost functions does not matter so much in this framework as long as they are reasonable.

3.3.3 Evaluation with Speech Input

Next, we made experiments using the spoken queries and the speech recognition system. The distribution of recognized keywords and corresponding confidence measures is shown in Table 2. The precision for the keywords with high confidence measures is better, thus the confidence measure works well. Summary of the result is given in Table 3.

The average number of matched items is 13.3 and the success rate is 87%. Some degradation from the case of text input is observed. The average rank of the correct item is 4.1. For reference, if we do not use the confidence measure CM_i, the figure is 4.4, which verifies the effect of the confidence measure. The proposed dialogue strategy with either heuristic function reaches the correct one in around 3 turns, which is 30% reduction compared with the baseline.

It should be noticed that, although the initial matching accuracy is lowered with the speech input, the improvement by the proposed strategy is larger and the number of dialogue turns is close to the text-input case. The result confirms that the proposed framework is effective in speech interface.

4 Conclusion

We present a method to generate guiding utterances for narrowing down users' query results obtained by an information retrieval system. By selecting the most efficient item, the dialogue is restricted to system-initiated "yes/no" questions. We have evaluated our method with a query task on the appliance manual where structured task knowledge is available. The number of average dialogue turns is reduced by about 30% compared with a baseline method in which the candidates are confirmed according to their matching scores. This result demonstrates that the proposed system helps users find their intended items more efficiently.

References

J.F. Allen, B.W. Miller, E.K. Ringger, and T. Sikorski. 1996. A robust system for natural spoken dialogue. In *Proc. of the 34th An-*

Table 2: The precision of keywords and their confidence measures

confidence measure of keyword	1	1 - 0.9	0.9 - 0.8	0.8 - 0.7	0.7 -	total
# correctly recognized words	279	15	10	18	16	338
# incorrectly recognized words	63	17	20	49	60	209
precision	82%	47%	33%	27%	21%	62%

nual Meeting of the Association for Computational Linguistics (ACL-96), pages 62–70.

S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. 1996. Dialog in the RAILTEL telephone-based system. In *Proc. Int'l Conf. on Spoken Language Processing.*

Jennifer Chu-Carroll and Bob Carpenter. 1998. Dialogue management in vector-based call routing. In *Proc. of COLING-ACL98*, pages 256–262.

Matthias Denecke. 1997. An information-based approach for guiding multi-modal human-computer-interaction. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97).*

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

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

K. Komatani, K. Tanaka, H. Kashima, and T. Kawahara. 2001. Domain-independent spoken dialogue platform using key-phrase spotting based on combined language model. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pages 1319–1322.

L.F. Lamel, S. Rosset, J-L.S. Gauvain, and S.K. Bennacef. 1999. The LIMSI ARISE system for train travel information. In *Proc. of Int'l Conf. on Acustics, Speech and Signal Processing (ICASSP).*

A. Lee, T. Kawahara, K. Takeda, and K. Shikano. 2000. A new phonetic tied-mixture model for efficient decoding. In *Proc. of Int'l Conf. on Acustics, Speech and Signal Processing (ICASSP)*, pages 1269–1272.

A. Lee, T. Kawahara, and K. Shikano. 2001. Julius – an open source real-time large vocabulary recognition engine. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pages 1691–1694.

E. Levin, R. Pieraccini, and W. Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–79.

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. on Spoken Language Processing.*

Diane J. Litman, Michael S. Kearns, Satinder Singh, and Marilyn A. Walker. 2000. Automatic optimization of dialogue management. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pages 502–508.

Y. Niimi and Y. Kobayashi. 1996. A dialog control strategy based on the reliability of speech recognition. In *Proc. Int'l Conf. on Spoken Language Processing.*

Alexandros Potamianos, Egbert Ammicht, and Hong-Kwang J. Kuo. 2000. Dialogue management in the bell labs communicator system. In *Proc. Int'l Conf. on Spoken Language Processing.*

R. San-Segundo, B. Pellom, W. Ward, and J. Pardo. 2000. Confidence measures for dialogue management in the CU communicator system. In *Proc. of Int'l Conf. on Acustics, Speech and Signal Processing (ICASSP).*

J. Sturm, E. Os, and L. Boves. 1999. Issues in spoken dialogue systems: Experiences with the Dutch ARISE system. In *Proc. ESCA workshop on Interactive Dialogue in Multi-Modal Systems.*