SPOKEN DIALOGUE SYSTEM FOR QUERIES ON APPLIANCE MANUALS USING HIERARCHICAL CONFIRMATION STRATEGY

Tatsuya Kawahara *Ryosuke Ito* *Kazunori Komatani*

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

ABSTRACT

We address a dialogue framework for queries on manuals of electric appliances with a speech interface. Users can make queries by unconstrained speech, from which keywords are extracted and matched to the items in the manual. As a result, so many items are usually obtained. Thus, we introduce an effective dialogue strategy which narrows down the items using a tree structure extracted from the manual. Three cost functions are presented and compared to minimize the number of dialogue turns. We have evaluated the system performance on VTR manual query task. The number of average dialogue turns is reduced to 71% using our strategy compared with a conventional method that makes confirmation in turn according to the matching likelihood. Thus, 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, train information and personal schedules. 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 that is described in less rigid format or natural language. 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 can be regarded as the special case.

In this paper, we deal with the problem of finding the appropriate entry in the manuals 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 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 item 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 the selection and want to specify by their own language. The proposed framework allows users to make queries spontaneously and makes use of directory structure in the follow-up dialogue to determine the most appropriate one. Although there are previous studies on optimizing dialogue strategies[1][2][3][4], most of them assume the tasks of filling semantic slots that are definitely and manually defined. For example, Denecke[5] proposed a method to generate guiding questions by making use of a tree structure that is constructed by unifying pre-defined keywords and semantic slots. However, few studies focus on follow-up dialogue of general information retrieval with speech input.

2. SYSTEM OVERVIEW

An overview of the proposed system is illustrated in Figure 1. It consists of following processes.

1. Keyword spotting from user utterances using an ASR (automatic speech recognition) system[6]

   A natural spoken language query is handled with continuous speech recognition and keywords are extracted. A confidence measure CM_i is assigned to each keyword i based on the N-best recognition result[7].

2. Matching with manual entries (=documents)

   The extracted keywords are matched to a set of manual entries. The matching is performed to the initial portion (index and first summary paragraph) of each

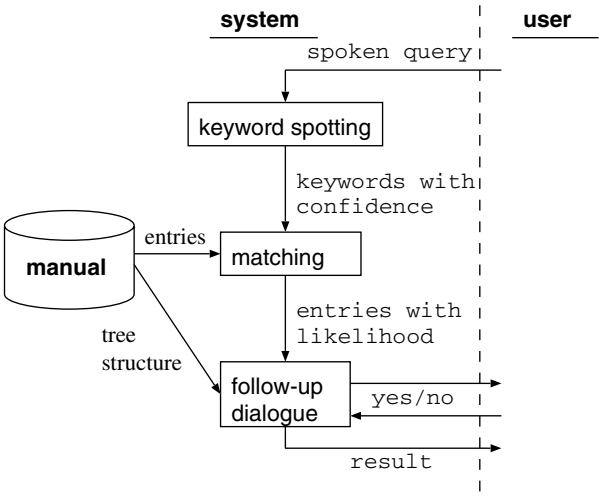

Fig. 1. System overview

manual section. We adopt the conventional matching score function in information retrieval, except that the confidence measure CM_i is incorporated for the speech input. The matching score of an entry j is defined as:

$$L_j = \frac{1}{n_j} \sum_i (CM_i * \log \frac{N}{df_i})$$

Here, df_i is the number of entries that contain keyword i referred as a document frequency and N is the total number of entries. 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 entry j.

3. Dialogue to determine the most appropriate item 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, a dialogue is invoked to narrow down to the intended item. This dialogue is restricted to system-initiated "yes/no" questions (the tree does not have to be binary) in order to avoid further recognition errors in the back-up dialogue, although the proposed framework can be extended to questions among multiple choices.

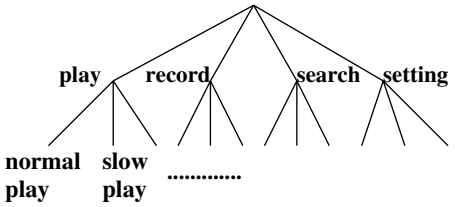

Fig. 2. Example of tree structure of manual (VTR)

3. 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 confidences 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 hierarchical 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 a VTR manual is shown in Figure 2.

3.1. Dialogue Algorithm

The algorithm to find the most appropriate entry through dialogue is described.

For each node of the tree, a 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 entries) is 1.0.
- For a non-leaf node, the sum of the likelihoods 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 in the next subsection 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 confirmed node. If the answer is "no", eliminate all descendants of the denied node.
4. Repeat the process until only one node (or nothing) remains.

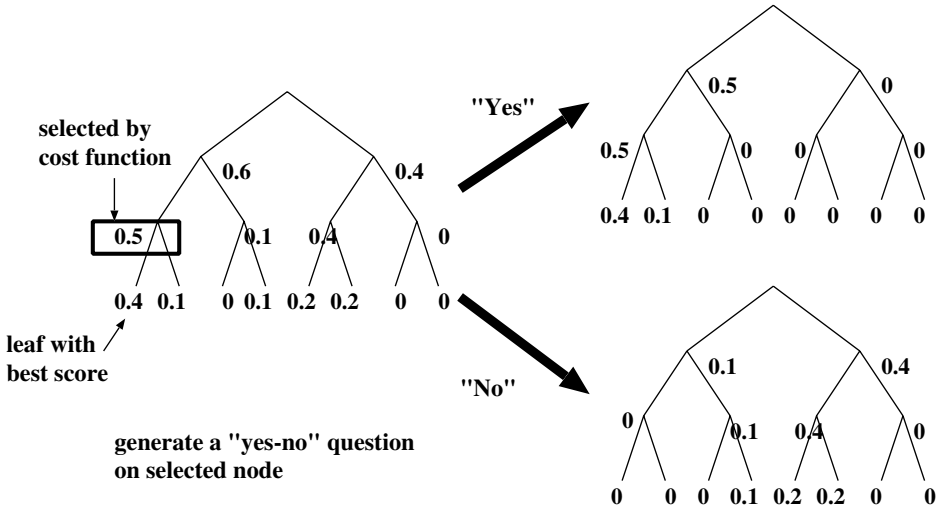

Fig. 3. Dialogue flow using manual structure and cost function

The above processes are illustrated in Figure 3.

The dialogue strategy can be regarded as a directory search for the tree whose leaves are dynamically assigned with statistical matching scores.

3.2. Cost Functions for Generating Questions

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.

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

  Here, $Ques_j$ is the estimated number of following questions needed when the answer is "yes" or "no". It is computed recursively by expanding the sub-trees, and is assigned with 0 when the number of remaining nodes gets one. The function is expected to be accurate but computationally costly.

These are experimentally compared in the next section.

4. EXPERIMENTAL EVALUATION

4.1. Task and System Implementation

The proposed system is implemented for the query task on a VTR (Video Tape Recorder) manual that consists of 111 pages and 47 entries. 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[8]. The language model is based on a finite state grammar derived from the domain-specific corpus, that is the manual text in this task. The acoustic model is a gender-dependent phonetic tied-mixture (PTM) triphone model 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) as well as several spontaneous queries without any scenarios. In total, we have 195 query utterances, of which 157 can 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 entry 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 entry 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 entry.

Table 1. Result with text input

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

Table 2. Result with speech input

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

4.2. Evaluation with Text Input

At first, the system is evaluated with text input, that is transcription of the collected queries. The result is summarized 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 entry 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 significantly 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.

4.3. Evaluation with Speech Input

Next, we made experiments using the spoken queries and the speech recognition system. Summary of the result is given in Table 2. The average number of matched entries is 13.3 and the query success rate is 87%. Some degradation from the case of text input is observed. The average rank of the correct entry 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 functions reaches the correct one in around 3 turns, which is 30% reduction compared with the baseline. The improvement from the baseline is significant, but the difference of performance by the three functions is not statistically significant in this case. 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 the speech interface.

5. CONCLUSIONS

We have addressed a dialogue framework that narrows down user's query results which an information retrieval system outputs. The follow-up dialogue to filter the matched candidates is significant especially with the speech interface, but it is not possible to apply conventional dialogue strategies that assume definite semantic slots to be filled. The proposed dialogue framework generates questions based on an information theoretic criterion to eliminate irrelevant candidates.

In this work, we assume a task on the appliance manuals where structured task knowledge is available. A hierarchical confirmation strategy is proposed by making use of the tree structure of the manual, and three cost functions for selecting question nodes are presented and compared. The experimental results demonstrate the effectiveness of the proposed framework and feasibility of the spoken dialogue interface for manual queries of electric appliances.

Acknowledgments: We would like to thank Prof. H. G. Okuno for his advice on this study.

REFERENCES

[1] Y.Niimi and Y.Kobayashi. A dialog control strategy based on the reliability of speech recognition. In *Proc. ICSLP*, pages 534–537, 1996.

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

[3] D.Litman, M.Kearns, and M.Walker. Automatic optimization of dialogue management. In *Proc. COLING*, pages 502–508, 2000.

[4] K.Wang. A plan-based dialog system with probabilistic inferences. In *Proc. ICSLP*, volume 2, pages 644–647, 2000.

[5] M.Denecke and A.Waibel. Dialogue strategies guiding users to their communicative goals. In *Proc. EUROSPEECH*, 1997.

[6] 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.

[7] K.Komatani and T.Kawahara. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. COLING*, pages 467–473, 2000.

[8] A.Lee, T.Kawahara, and K.Shikano. Julius – an open source real-time large vocabulary recognition engine. In *Proc. EUROSPEECH*, pages 1691–1694, 2001.