Speech-based Information Retrieval System with Clarification Dialogue Strategy

Teruhisa Misu **Tatsuya Kawahara**
School of informatics
Kyoto University
Sakyo-ku, Kyoto, Japan
misu@ar.media.kyoto-u.ac.jp

Abstract

This paper addresses a dialogue strategy to clarify and constrain the queries for speech-driven document retrieval systems. In spoken dialogue interfaces, users often make utterances before the query is completely generated in their mind; thus input queries are often vague or fragmental. As a result, usually many items are matched. We propose an efficient dialogue framework, where the system dynamically selects an optimal question based on information gain (IG), which represents reduction of matched items. A set of possible questions is prepared using various knowledge sources. As a bottom-up knowledge source, we extract a list of words that can take a number of objects and potentially causes ambiguity, using a dependency structure analysis of the document texts. This is complemented by top-down knowledge sources of metadata and handcrafted questions. An experimental evaluation showed that the method significantly improved the success rate of retrieval, and all categories of the prepared questions contributed to the improvement.

1 Introduction

The target of spoken dialogue systems is being extended from simple databases such as flight information (Levin et al., 2000; Potamianos et al., 2000) to general documents (Fujii and Itou, 2003) including newspaper articles (Chang et al., 2002; Hori et al., 2003). In such systems, the automatic speech recognition (ASR) result of the user utterance is matched against a set of target documents using the vector space model, and documents with high matching scores are presented to the user.

In this kind of document retrieval systems, user queries must include sufficient information to identify the desired documents. In conventional document query tasks with typed-text input, such as TREC QA Track (NIST and DARPA, 2003), queries are (supposed to be) definite and specific. However, this is not the case when speech input is adopted. The speech interface makes input easier. However, this also means that users can start utterances before queries are thoroughly formed in their mind. Therefore, input queries are often vague or fragmental, and sentences may be ill-formed or ungrammatical. Moreover, important information may be lost due to ASR errors. In such cases, an enormous list of possible relevant documents is usually obtained because there is very limited information that can be used as clues for retrieval. Therefore, it is necessary to narrow down the documents by clarifying the user's intention through a dialogue.

There have been several studies on the follow-up dialogue, and most of these studies assume that the target knowledge base has a well-defined structure. For example, Denecke (Denecke and Waibel, 1997) addressed a method to generate guiding questions based on a tree structure constructed by unifying pre-defined keywords and semantic slots. However, these approaches are not applicable to general docu-

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 1003–1010, Vancouver, October 2005.

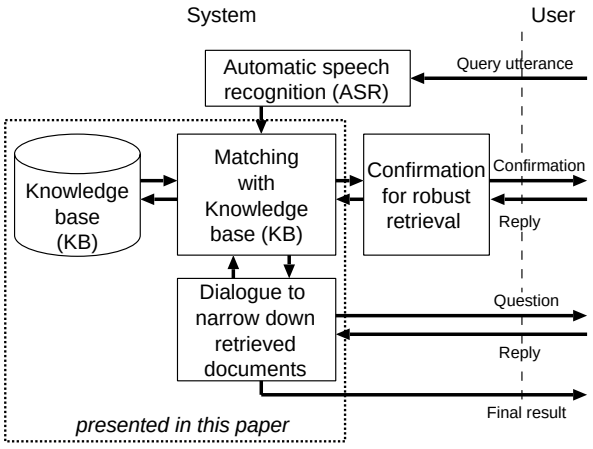

Figure 1: System overview

ment sets without such structures.

In this paper, we propose a dialogue strategy to clarify the user's query and constrain the retrieval for a large-scale text knowledge base, which does not have a structure nor any semantic slots. In the proposed scheme, the system dynamically selects an optimal question, which can reduce the number of matched items most efficiently. As a criterion of efficiency of the questions, information gain (IG) is defined. A set of possible questions is prepared using bottom-up and top-down knowledge sources. As a bottom-up knowledge source, we conduct dependency structure analysis of the document texts, and extract a list of words that can take a number of objects, thus potentially causing ambiguity. This is combined with top-down knowledge sources of metadata and hand-crafted questions. The system then updates the query sentence using the user's reply to the question, so as to generate a confirmation to the user.

2 Document retrieval system for large-scale knowledge base

2.1 System overview

We have studied a dialogue framework to overcome the problems in speech-based document retrieval systems. In the framework, the system can handle three types of problems caused by speech input: ASR errors, redundancy in spoken language expression, and vagueness of queries. First, the system realizes robust retrieval against ASR errors and redundancies by detecting and confirming them. Then, the system makes questions to clarify the user's query and narrow down the retrieved documents.

The system flow of these processes is summarized below and also shown in Figure 1.

1. Recognize the user's query utterance.
2. Make confirmation for phrases which may include critical ASR errors.
3. Retrieve from knowledge base (KB).
4. Ask possible questions to the user and narrow down the matched documents.
5. Output the retrieval results.

In this paper, we focus on the latter stage of the proposed framework, and present a clarification dialogue strategy to narrow down documents.

Table 1: Document set (Knowledge Base: KB)

Text collection	# documents	text size (byte)
glossary	4,707	1.4M
FAQ	11,306	12M
DB of support articles	23,323	44M

2.2 Task and back-end retrieval system

Our task involves text retrieval from a large-scale knowledge base. For the target domain, we adopt a software support knowledge base (KB) provided by Microsoft Corporation. The knowledge base consists of the following three kinds: glossary, frequently asked questions (FAQ), and support articles. The specification is listed in Table 1, and there are about 40K documents in total. An example of support article is shown in Figure 2.

Dialog Navigator (Kiyota et al., 2002) has been developed at University of Tokyo as a retrieval system for this KB. The system accepts a typed-text input from users and outputs a result of the retrieval. The system interprets an input sentence by taking syntactic dependency and synonymous expression into consideration for matching it with the KB. The target of the matching is the summaries and detail information in the support articles, and the titles of the Glossary and FAQ. The retrieved result is displayed to the user as the list of documents like Web

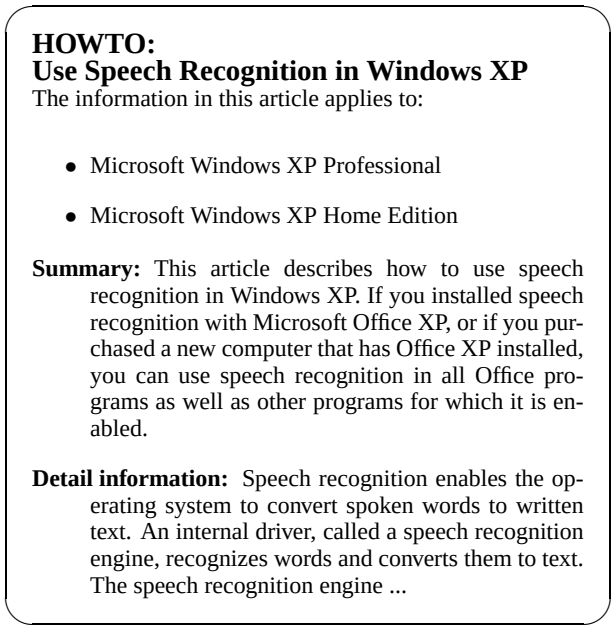

Figure 2: Example of software support article

search engines. Since the user has to read detail information of the retrieved documents by clicking their icons one by one, the number of items in the final result is restricted to about 15.

In this work, we adopt Dialog Navigator as a back-end system and construct a spoken dialogue interface.

3 Dialogue strategy to clarify user's vague queries

3.1 Dialogue strategy based on information gain (IG)

In the proposed clarification dialogue strategy, the system asks optimal questions to constrain the given retrieval results and help users find the intended ones. Questions are dynamically generated by selecting from a pool of possible candidates that satisfy the precondition. The information gain (IG) is defined as a criterion for the selection. The IG represents a reduction of entropy, or how many retrieved documents can be eliminated by incorporating additional information (a reply to a question in this case). Its computation is straightforward if the question classifies the document set in a completely disjointed manner. However, the retrieved documents may belong to two or more categories for some questions, or may not belong to any category. For example, some documents in our KB are related with multiple versions of MS-Office, but others may be irrelevant to any of them. Moreover, the matching score of the retrieved documents should be taken into account in this computation. Therefore, we define IG $H(S)$ for a candidate question S by the following equations.

$$H(S) = -\sum_{i=0}^{n} P(i) \cdot \log P(i)$$

$$P(i) = \frac{|C_i|}{\sum_{i=0}^{n} |C_i|}$$

$$|C_i| = \sum_{D_k \in i} CM(D_k)$$

Here, D_k denotes the k-th retrieved document by matching the query to the KB, and $CM(D)$ denotes the matching score of document D. Thus, C_i represents the number of documents classified into category i by candidate question S, which is weighted with the matching score. The documents that are not related to any category are classified as category 0.

The system flow incorporating this strategy is summarized below and also shown in Figure 3.

1. For a query sentence, retrieve from KB.
2. Calculate IG for all possible candidate questions which satisfy precondition.
3. Select the question with the largest IG (larger than a threshold), and ask the question to the user. Otherwise, output the current retrieval result.
4. Update the query sentence using the user's reply to the question.
5. Return to 1.

This procedure is explained in detail in the following sections.

3.2 Question generation based on bottom-up and top-down knowledge sources

We prepare a pool of questions using three methods based on bottom-up knowledge together with top-down knowledge of KB. For a bottom-up knowledge

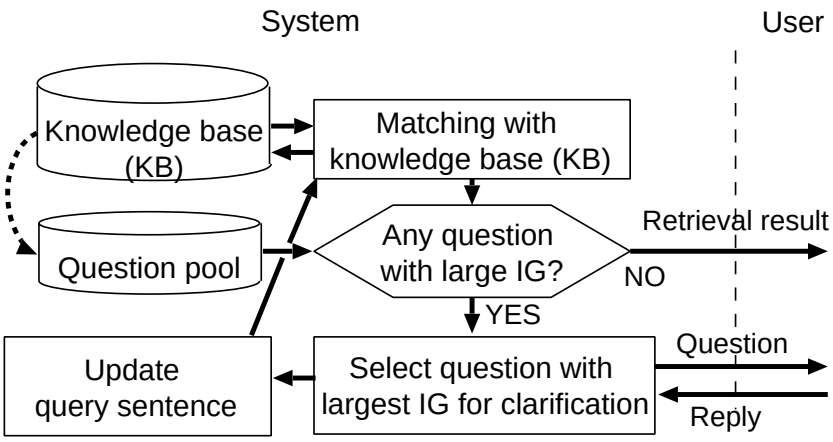

Figure 3: Overview of query clarification

Table 2: Examples of candidate questions (Dependency structure analysis: method 1)

Question	Precondition	Ratio of applicable doc.	IG
What did you delete?	Query sentence includes "delete"	2.15 (%)	7.44
What did you install?	Query sentence includes "install"	3.17 (%)	6.00
What did you insert?	Query sentence includes "insert"	1.12 (%)	7.12
What did you save?	Query sentence includes "save"	1.81 (%)	6.89
What is the file type?	Query sentence includes "file"	0.94 (%)	6.00
What did you setup?	Query sentence includes "setup"	0.69 (%)	6.45

source, we conducted a dependency structure analysis on KB. As for top-down knowledge, we make use of metadata included in KB and human knowledge.

3.2.1 Questions based on dependency structure analysis (method 1)

This type of question is intended to clarify the modifier or object of some words, based on dependency structure analysis, when they are uncertain. For instance, the verb "delete" can have various objects such as "application program" or "address book". Therefore, the query can be clarified by identifying such objects if they are missing. However, not all words need to be confirmed because the modifier or object can be identified almost uniquely for some words. For instance, the object of the word "shutdown" is "computer" in most cases in this task domain. It is tedious to identify the object of such words. We therefore determine the words to be confirmed by calculating entropy for modifier-head pairs from the text corpus. The procedure is as follows.

1. Extract all modifier-head pairs from the text of KB and query sentences (typed input) to another retrieval system[1] provided by Microsoft Japan.
2. Calculate entropy $H(m)$ for every word based on probability $P(i)$. This $P(i)$ is calculated with the occurrence count $N(m)$ of word m that appears in the text corpus and the count $N(i,m)$ of word m whose modifier is i.

$$H(m) = -\sum_i P(i) * \log P(i)$$

$$P(i) = \frac{N(i,m)}{N(m)}$$

[1] http://www.microsoft.com/japan/enable/nlsearch/

Table 3: Examples of candidate questions (Metadata: method 2)

Question	Precondition	Ratio of applicable doc.	IG
What is the version of your Windows?	None	30.03 (%)	2.63
What is your application?	None	30.28 (%)	2.31
What is the version of your Word?	Query sentence includes “Word”	3.76 (%)	2.71
What is the version of your Excel?	Query sentence includes “Excel”	4.13 (%)	2.44

Table 4: List of candidate questions (Human knowledge: method 3)

Question	Precondition	Ratio of applicable doc.	IG
When did the symptom occur?	None	15.40 (%)	8.08
Tell me the error message.	Query sentence includes “error”	2.63 (%)	8.61
What do you concretely want to do?	None	6.98 (%)	8.04

As a result, we selected 40 words that have a large value of entropy. Question sentences for these words were generated with a template of “What did you ...?” and unnatural ones were corrected manually. Categories for IG calculation are defined by objects of these words included in matched documents. The system can make question using this method when these words are included in the user’s query. Table 2 lists examples of candidate questions using this method. In this table, ratio of applicable document corresponds to the ratio of documents that include the words selected above, and IG is calculated using applicable documents.

3.2.2 Questions based on metadata included in KB (method 2)

We also prepare candidate questions using the metadata attached to the KB. In general large-scale KBs, metadata is usually attached to manage them efficiently. For example, category information is attached to newspaper articles and books in libraries. In our target KB, a number of documents include metadata of product names to which the document applies. The system can generate question to which the user’s query corresponds using this metadata. However, some documents are related with multiple versions, or may not belong to any category. Therefore, the performance of these questions greatly depends on the characteristics of the metadata.

Fourteen candidate questions are prepared using this method. Example of candidate questions are listed in Table 3. Ratio of applicable document corresponds to the ratio of documents that have metadata of target products.

3.2.3 Questions based on human knowledge (method 3)

Software support is conventionally provided by operators at call centers. We therefore prepare candidate questions based on the human knowledge that has been accumulated there. This time, three kinds of questions are hand-crafted. For instance, the question “When did the symptom occur?” tries to capture key information to identify relevant documents. The categories for IG caluclation are defined using hand-crafted rules by focusing on key-phrases such as “after ...” or “during ...”. Candidate questions are listed in Table 4.

An example dialogue where the system asks questions based on IG is in Figure 4.

3.3 Update of retrieval query sentence

Through the dialogue to clarify the user’s query, the system updates the query sentence using the user’s reply to the question. Our backend information retrieval system does not adopt simple “bag-

S1: What is your problem?
U1: Too garbled to read.
(Retrieval results):

1. Close button and maximize button are garbled.
2. Characters are garbled in Outlook Today.
3. Characters are garbled while inserting Japanese text.
4. VB application is garbled to read.
   . . .

(Calculate IG)
· Candidate question 1:
What is garbled to read? – IG 5.27
· Candidate question 2:
What is the version of your Windows? – IG 1.43
· Candidate question 3:
When did the symptom occur? – IG 2.47
. . .

S2: **(Select question with largest IG)**
What is garbled to read?
U2: Characters on window button.
S3: **(Update query sentence)**
Retrieving with "Characters on window button are too garbled to read".

Figure 4: Example dialogue

of-words" model, but conducts a more precise dependency structure analysis for matching; therefore forming an appropriate query sentence is desirable rather than simply adding keywords. Moreover, it is more comprehensible to the user to present the updated query sentence than to show the sequence of ASR results. Here, the update rules of the query sentence are prepared as follows.

1. Questions based on dependency structure analysis
   The user's reply is added immediately before of after the word that is the reply's modifying head. For instance, the reply to the question "What did you delete?" is inserted right after the word "delete" in the query sentence.

2. Questions based on metadata of KB
   Phrases "In {Product name} {version name}" are added to the query sentence.

3. Questions based on human knowledge
   The position where the user's reply is inserted is specified beforehand for each question candidate. For instance, the reply to the question "Tell me the error message." is inserted right after the word "error" in the query sentence.

A dialogue example where the system updates the user's query is shown in Figure 5. In the example, the system makes confirmation "Retrieving with 'When I try to open it in explorer, I cannot open Excel 2002 file' " at the end of the dialogue before presenting the actual retrieval result.

3.4 Experimental evaluation

We implemented and evaluated the proposed method. We collected a test data by 14 subjects who had not used our system Each subject was requested to retrieve support articles for 14 tasks, which consisted of prepared scenarios (query sentences were not given). The subjects were allowed to utter a query again up to twice per task if they thought an adequate retrieval result was not obtained. As a result, we collected 238 utterances for 196 (=14 $\times$ 14) tasks in total. An example of scenario and user utterances are shown in Figure 6. The average word accuracy of ASR was 82.9%. The threshold value in IG that the system makes a question is set to 1.0 ini-

S1:	What is your problem?
U1:	I cannot open the file.
S2:	What is the file type? (method 1)
U2:	Excel file.
(Update query sentence):	"I cannot open Excel file."
S3:	What is the version of your Excel? (method 2)
U3:	My Excel is version 2002.
(Update query sentence):	"I cannot open Excel 2002 file."
S4:	When did the symptom occur? (method 3)
U4:	Tried to open it in explorer.
S5:	Retrieving with "When I try to open it in explorer, I cannot open Excel 2002 file".

Figure 5: Query sentence update using user's reply

- **An example of scenario**
  You are looking for restaurant in Kyoto using WWW. You have found a nice restaurant and tried to print out an image of the map showing the restaurant. However, it is not printed out. (Your browser is IE 6.0)
- **Examples of users' utterance**
  - I want to print an image of map.
  - I can't print out.
  - I failed to print a picture in homepage using IE.
  - Please tell me how to print out an image.

Figure 6: Example of scenario and user utterances

tially, and incremented by 0.3 every time the system generates a question through a dialogue session.

First, we evaluated the success rate of retrieval. We regarded a retrieval as successful when the retrieval result contained a correct document entry for the scenario. We compared the following cases.

1. Transcript: A correct transcript of the user utterance, prepared manually, was used as an input.
2. ASR result (baseline): The ASR result was used as an input.
3. Proposed method (log data): The system generated questions based on the proposed method, and the user replied to them as he/she thought appropriate.

We also evaluated the proposed method by simulation in order to confirm its theoretical effect. Various factors of the entire system might influence the performance in real dialogue which is evaluated by the log data. Specifically, the users might not have answered the questions appropriately, or the replies might not have been correctly recognized. Therefore, we also evaluated with the following condition.

4. Proposed method (simulation): The system generated questions based on the proposed method, and appropriate answers were given manually.

Table 5 lists the retrieval success rate and the rank of the correct document in the retrieval result, by these cases. The proposed method achieved a better success rate than when the ASR result was used. An improvement of 12.6% was achieved in the simulation case, and 7.7% by the log data. These figures demonstrate the effectiveness of the proposed approach. The success rate of the retrieval was about 5% higher in the simulation case than the log data. This difference is considered to be caused by following factors.

1. ASR errors in user's uttered replies
   In the proposed strategy, the retrieval sentence is updated using the user's reply to the question regardless of ASR errors. Even when the user notices the ASR errors, he/she cannot correct them. Although it is possible to confirm them using ASR confidence measures, it makes dialogue more complicated. Hence, it was not implemented this time.
2. User's misunderstanding of the system's questions
   Users sometimes misunderstood the system's questions. For instance, to the system question "When did the symptom occur?", some user

Table 5: Success rate and average rank of correct document in retrieval

	Success rate	Rank of correct doc.
Transcript	76.1%	7.20
ASR result (baseline)	70.7%	7.45
Proposed method (log data)	78.4%	4.40
Proposed method (simulation)	83.3%	3.85

Table 6: Comparison of question methods

	Success rate	# generated questions (per dialogue)
ASR result (baseline)	70.7%	—
Dependency structure analysis (method 1)	74.5%	0.38
Metadata (method 2)	75.7%	0.89
Human knowledge (method 3)	74.5%	0.97
All methods (method 1-3)	83.3%	2.24

replied simply "just now" instead of key information for the retrieval. To this problem, it may be necessary to make more specific questions or to display reply examples.

We also evaluated the efficiency of the individual methods. In this experiment, each of the three methods was used to generate questions. The results are in Table 6. The improvement rate by the three methods did not differ very much, and most significant improvement was obtained by using the three methods together. While the questions based on human knowledge are rather general and were used more often, the questions based on the dependency structure analysis are specific, and thus more effective when applicable. Hence, the questions based on the dependency structure analysis (method 1) obtained a relatively high improvement rate per question.

4 Conclusion

We proposed a dialogue strategy to clarify user' queries for document retrieval tasks. Candidate questions are prepared based on the dependency structure analysis of the KB together with KB metadata and human knowledge. The system selects an optimal question based on information gain (IG). Then, the query sentence is updated using the user's reply. An experimental evaluation showed that the proposed method significantly improved the success rate of retrieval, and all categories of the prepared questions contributed to the improvement.

The proposed approach is intended for restricted domains, where all KB documents and several knowledge sources are available, and it is not applicable to open-domain information retrieval such as Web search. We believe, however, that there are many targets of information retrieval in restricted domains, for example, manuals of electric appliances and medical documents for expert systems. The methodology proposed here is not so dependent on the domains, thus applicable to many other tasks of this category.

5 Acknowledgements

The authors are grateful to Prof. Kurohashi and Dr. Kiyota at University of Tokyo and Dr. Komatani at Kyoto University for their helpful advice.

References

E. Chang, F. Seide, H. M. Meng, Z. Chen, Y. Shi, and Y. C. Li. 2002. A system for spoken query information retrieval on mobile devices. *IEEE Trans. on Speech and Audio Processing*, 10(8):531–541.

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

A. Fujii and K. Itou. 2003. Building a test collection for speech-driven Web retrieval. In *Proc. EUROSPEECH*.

C. Hori, T. Hori, H. Isozaki, E. Maeda, S. Katagiri, and S. Furui. 2003. Deriving disambiguous queries in a spoken interactive ODQA system. In *Proc. IEEE-ICASSP*.

Y. Kiyota, S. Kurohashi, and F. Kido. 2002. "Dialog Navigator": A question answering system based on large text knowledge base. In *Proc. COLING*, pages 460–466.

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. ICSLP*.

NIST and DARPA. 2003. The twelfth Text REtrieval Conference (TREC 2003). In *NIST Special Publication SP 500–255*.

A. Potamianos, E. Ammicht, and H.-K. J. Kuo. 2000. Dialogue management in the Bell labs Communicator system. In *Proc. ICSLP*.