Available online at www.sciencedirect.com
SCIENCE DIRECT
Speech Communication 48 (2006) 1137–1150
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Dialogue strategy to clarify user's queries for document retrieval system with speech interface

Teruhisa Misu *, Tatsuya Kawahara

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

Received 3 June 2005; received in revised form 31 March 2006; accepted 10 April 2006

Abstract

This paper proposes a dialogue strategy for clarifying and constraining queries to document retrieval systems with speech input interfaces. It is indispensable for spoken dialogue systems to interpret user's intention robustly in the presence of speech recognition errors and extraneous expressions characteristic of spontaneous speech. In speech input, moreover, users' queries tend to be vague, and they may need to be clarified through dialogue in order to extract sufficient information to get meaningful retrieval results. In conventional database query tasks, it is easy to cope with these problems by extracting and confirming keywords based on semantic slots. However, it is not straightforward to apply such a methodology to general document retrieval tasks.

In this paper, we first introduce two statistical measures for identifying critical portions to be confirmed. The *relevance score* (RS) represents the matching degree with the document set. The *significance score* (SS) detects portions that affect retrieval results. With these measures, the system can generate confirmations to handle speech recognition errors, prior to and after the retrieval, respectively. Then, we propose a dialogue strategy for generating clarifications to narrow down the retrieved items, especially when many documents are matched because of a vague input query. The optimal clarification question is dynamically selected based on information gain (IG) – the reduction in the number of matched items. A set of possible clarification 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 hand-crafted questions.

Our dialogue strategy is implemented and evaluated against a software support knowledge base of 40 K entries. We demonstrate that our strategy significantly improves the success rate of retrieval.

Keywords: Spoken dialogue system; Information retrieval; Document retrieval; Dialogue strategy

1. Introduction

In the past years, a great number of spoken dialogue systems have been developed. Their typical task domains include airline information (ATIS&DARPA Communicator) (Levin et al., 2000; Potamianos et al., 2000; Rudnicky et al., 2000; Seneff and Polifroni, 2000), train information (RAILTEL) (Bennacef et al., 1996), and weather information (Zue et al., 2000). Although there are several systems that address planning or tutoring dialogues

* Corresponding author.
E-mail address: misu@ar.media.kyoto-u.ac.jp (T. Misu).

0167-6393/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2006.04.001

(Allen et al., 1996; Stent et al., 1999; Rayner et al., 2003), in most cases, the tasks of dialogue systems are database retrieval and transaction processing. In these tasks, the speech understanding process tries to convert automatic speech recognition (ASR) results into semantic representations equivalent to database query (SQL) commands. This means the system designers can define keywords to achieve the task a priori and manually using database field names and their values. For example, in flight information tasks, a set of keywords is defined for origin, destination, date, and flight number entry. Thus, the user queries of these tasks are interpreted by extracting such keywords from the ASR results, and the dialogue process is designed to disambiguate unfixed database slots.

On the other hand, in recent years, the target of spoken dialogue systems has been extended to general document retrieval (Barnett et al., 1997), including manuals and Web pages. These kinds of applications are expected to be useful especially when retrieving information with keyboardless devices such as a PDA, a tablet PC and a car navigation system. For example, Chang et al. (2002) developed a spoken query system that retrieves documents from the Chinese news corpus used in TREC-5 and TREC-6. Fujii and Itou (2003) designed the speech-driven Web retrieval sub-task in NTCIR-3, which is a TREC-style evaluation workshop. Harabagiu et al. (2002), Hori et al. (2003) and Schofield and Zheng (2003) present speech-input open-domain question-answering (ODQA) systems by enhancing typed-input ODQA systems. In such document retrieval tasks, it is not possible to convert ASR results into definite semantic representations. Therefore, the automatic speech recognition (ASR) result of the user query is usually matched against a set of target documents by using a vector space model, which represents documents and queries using a vector of occurrence counts of words or lexical features.

However, simple use of ASR results causes problems in document retrieval systems:

(1) Errors in automatic speech recognition (ASR).
Errors are inevitable in large vocabulary continuous speech recognition. Such errors in the query input consequently cause erroneous retrieval results, but not every recognition error affects information retrieval. Therefore, adequate confirmation is needed to eliminate misunderstandings caused by ASR errors.

(2) Redundancies in spoken language expressions.
In spontaneous speech, user utterances may include extraneous expressions such as disfluencies and irrelevant phrases, e.g. "you know" and "I want to know". This means not every portion of the user utterance is important for information retrieval. On the contrary, the inclusion of irrelevant phrases might degrade the matching with correct entries, thus such phrases should be eliminated or ignored in the matching.

(3) Vagueness of user's query.
Queries are (supposed to be) definite and specific in conventional document retrieval tasks with typed-text input (NIST and DARPA, 2003). However, this assumption fails when speech input is adopted. A speech interface makes input easier; however, this also means that users can start utterances before their queries are thoroughly formed in their mind. Therefore, input queries are often vague or fragmented, and sentences may be ill-formed or ungrammatical. 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.

To make information retrieval systems robust enough to deal with these problems, we need to design adequate confirmations and follow-up dialogue strategies dedicated to the document retrieval task. This paper addresses dialogue strategies focusing on such confirmations and clarifications.

There have been many studies on confirmation for ASR error correction, and most deal with database query tasks. In such tasks, since keywords are pre-defined from the task specification, the system can focus on them by using confidence measures (Bouwman et al., 1999; Komatani and Kawahara, 2000; Hazen et al., 2000; San-Segundo et al., 2000) to handle possible errors. However, it is not feasible to define such keywords in document retrieval tasks. The redundancy of spoken language expressions also makes it impractical to confirm every portion with low confidence. In this paper, we first propose two statistical measures that are computed for phrase units and are applicable to general information retrieval tasks. One is a relevance score with respect to the target document set, which is computed with a document language model and used for making a confirmation prior to the retrieval. The relevance score is also used to detect redundan-

cies in the query sentence. The other is the significance score in the document matching, which is computed after the retrieval by using *N*-best results and is used for prompting the user for post-selection, if necessary.

Then, we present a dialogue strategy for clarifying the user's query and constraining the retrieval. It addresses the problems of vagueness in the user's query as well as ambiguity caused by ASR errors. Most of the previous studies on these issues assume that the target knowledge base has a well-defined structure. For example, Denecke and Waibel (1997) devised a method to generate guiding questions based on a tree structure constructed by unifying pre-defined keywords and semantic slots. Komatani et al. (2002) also proposed a method to generate optimal clarification questions to identify the desired entry by using the content structure extracted from a manual of electric appliances. Lewis and Fabbrizio (2005) proposed a clarification algorithm to identify the call-type using a tree structure over the target data. However, these approaches are not applicable to general document sets without such structures. In our proposed scheme, the system dynamically selects an optimal clarification question that can reduce the number of matched items most efficiently. A set of possible questions is prepared using bottom-up and top-down knowledge sources. As the bottom-up knowledge source, we conducted a dependency structure analysis of the document texts and extracted a list of words that can take a number of objects, thus potentially causing ambiguity. This is combined with the top-down knowledge sources of metadata and hand-crafted questions. Effectiveness of clarification questions is defined using information gain (IG). After clarification, the system updates the query sentence using the user's reply to the question.

The proposed methods are implemented in a document retrieval system for a software support knowledge base (KB) of 40 K entries. This knowledge base is intended to be an automated help-desk for software support. Experimental evaluations were carried out to evaluate in terms of retrieval success rate and efficiency of dialogue.

The paper is organized as follows. Section 2 describes the system overview and task domain. Section 3 describes the confirmation strategy to handle ASR errors and redundancy in spoken queries and its experimental evaluation. Section 4 presents the dialogue strategy to clarify users' vague queries and its experimental evaluation. Section 5 concludes the paper.

2. Document retrieval system with speech interface

2.1. System overview

We aim to overcome the problems of document retrieval systems taking speech input, which are ASR errors, redundancies in the spoken language expressions, and vagueness of queries. In the proposed scheme, the system realizes robust retrieval against ASR errors and redundancies by detecting and confirming them based on two statistical measures. 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 Fig. 1.

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

2.2. Task and back-end retrieval system

Our task involves document retrieval from a large-scale knowledge base (KB). As the target domain, we adopt a software support KB provided

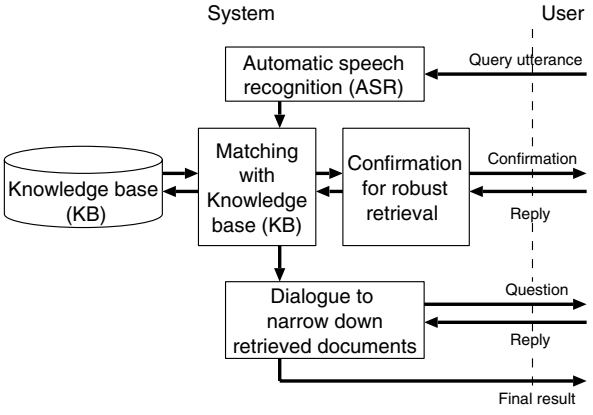

Fig. 1. System flow.

by Microsoft Corporation. The KB consists of a glossary, a frequently asked questions (FAQ) section, and support articles. The glossary and FAQ consist of titles and explanations. The support articles consist of titles, summaries, and detailed information. The specification is summarized in Table 1. There are about 40 K documents in the KB. An example support article is shown in Fig. 2.

Dialog Navigator (Kiyota et al., 2002) developed at the University of Tokyo is a retrieval system for this KB. The system accepts typed-text input and outputs a sequence of documents, like a Web search engine would. The system interprets an input sentence by taking syntactic dependency and synonymous expressions into consideration for matching with sentences in the KB. The target of the matching is the summaries and detailed information in the support articles and the titles of the Glossary and FAQ. Since the user has to read the detailed information in the retrieved documents by clicking on their icons one by one, the number of items in the final result is restricted to about 15. This system has been in service at http://www.microsoft.com/japan/navigator/ since April 2002.

We used Dialog Navigator as a back-end system and constructed our own spoken dialogue interface.

Table 1
Specification of target document set (Knowledge Base: KB)

Text collection	# Documents	Text size (byte)
Glossary	4707	1.4M
FAQ	11 306	12M
DB of support articles	23 323	44M

We focus on a dialogue strategy to interpret user utterances robustly, by taking into account the problems that are characteristic of spoken language as previously described.

3. Confirmation strategy for robust retrieval against ASR errors and redundancies in spoken language expressions

Appropriate confirmation is indispensable to eliminate misunderstandings caused by automatic speech recognition (ASR) errors. However, confirming every portion would be tedious, even with a reliable confidence measure, because not every erroneous portion necessarily affects retrieval results. We therefore consider the influence of recognition errors on retrieval and control confirmation accordingly.

Since Dialog Navigator outputs a dozen or so retrieved documents, as in Web search engines, ASR errors included in a query sentence are tolerable as long as the major retrieved documents remain unchanged. Therefore, we make use of the N-best results of ASR for the query, and check if there is a significant difference among the N-best sets of retrieved documents. If there actually is, we confirm the portions that caused the difference. This procedure is regarded as a posterior confirmation. On the other hand, if there is a critical error in the ASR result, such as in a product name, the subsequent retrieval would make no sense. Therefore, we also introduce confirmation prior to the retrieval for critical words.

The system flow including the confirmation is summarized below:

HOWTO:
Use Speech Recognition in Windows XP The information in this article applies to:

- Microsoft Windows XP Professional
- Microsoft Windows XP Home Edition

Summary: This article describes how to use speech recognition in Windows XP. If you installed speech recognition with Microsoft Office XP, or if you purchased a new computer that has Office XP installed, you can use speech recognition in all Office programs as well as other programs for which it is enabled.

Detail information: Speech recognition enables the operating system to convert spoken words to written text. An internal driver, called a speech recognition engine, recognizes words and converts them to text. The speech recognition engine ...

Fig. 2. Example support article.

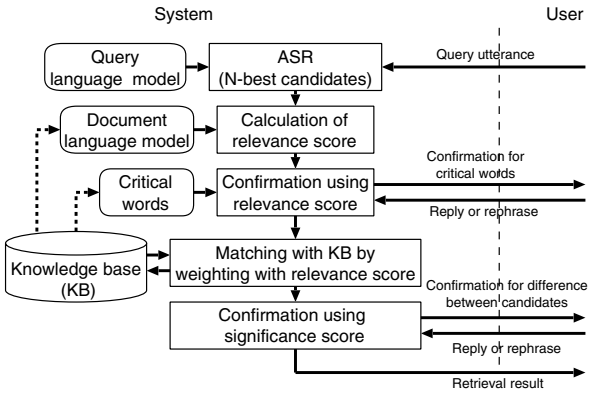

Fig. 3. Flow of confirmation strategy.

(1) Calculate the relevance score for each phrase of the ASR result.
(2) Make confirmations for critical words with low relevance scores.
(3) Make a retrieval from the KB for each of the *N*-best ASR results.
(4) Calculate significance scores, and generate a confirmation based on them.
(5) Output the retrieval results.

The flow is also shown in Fig. 3, and explained in the following sub-sections in detail. The query language model is trained with the target KB documents together with a query sentence corpus so it will be able to cope with spoken language expressions. The document language model is trained with only KB sentences and is used to measure the degree of matching with the document set.

3.1. Prior confirmation using relevance score (RS)

3.1.1. Definition of relevance score

Our relevance score measures the potential degree of matching with the document set. For this purpose, we introduce a document language model different from one used during ASR. To measure the perplexity of the input utterance, phrase by phrase, the semantic parser KNP[1] is used to segment the query sentence into phrase units called *bunsetsu*.[2]

The perplexity for a phrase including ASR errors usually gets larger because such a word sequence is contextually less frequent. The perplexity for out-of-domain phrases also tends to be large because they appear infrequently in the KB. To compute our relevance score (RS), we transform the perplexity (PP) using the sigmoid function which is widely used to convert distance into a confidence measure.

$$\mathrm{RS} = \frac{1}{1 + \exp(-\alpha * (\log \mathrm{PP} - \beta))}$$

Here, α and β are empirically set to -2.0 and 11.0 using data collected from four subjects, which are different from the data used in our evaluation. Fig. 4 shows an example of calculating the relevance score. In this example, phrases that appear in the beginning and end of the query sentence were incorrectly recognized because they were articulated weakly. The phrase, "*fuyouni natta* (= which is no longer needed)", does not contribute to document retrieval. The perplexity for these portions gets larger as a result, and the relevance score is correspondingly very small. In the beginning phrase, although the words "OS" and "IME" appears frequently in KB, the sequence "OS IME" is less frequent. Therefore, the system can determine the portion to be an ASR error by taking context into consideration.

3.1.2. Confirmation for critical words using relevance score

Critical words should be confirmed before the retrieval, because the retrieval result would be severely damaged if they are not correctly recognized. We define a set of critical words by using $tf \cdot idf$ values, which are derived from the target KB. As a result, we selected 35 words. Examples of these critical words[3] are listed in Table 2.

We use our relevance score to determine whether we should make confirmations for the critical words. If a critical word is contained in a phrase whose relevance score is lower than a threshold θ_1, a confirmation is made. The system presents the recognized query by high-lighting the phrases that may include ASR errors on the display. Users can confirm, discard, or correct the phrase by selecting these choices with a mouse, before passing the phrase to the matching module.

The relevance score (RS) is also used as a weight for phrases during the matching with the KB,

[1] http://www.kc.t.u-tokyo.ac.jp/nl-resource/knp.html.

[2] *Bunsetsu* is defined as a basic unit of Japanese grammar and it consists of a content word (or a sequence of nouns) followed by function words.

[3] These critical words are different from keywords in database query tasks. Our task cannot be achieved by simply extracting these words from the ASR result.

User utterance:
"*Os wa ME desuga fuyouni natta IME 2000 wo sakujo shitai no desu ga dou shitara ii deshou ka?*"
(Please tell me how to delete the IME 2000 which is no longer needed in Windows ME.)

ASR result:
"*Os IME desu ga fuyou ni natta IME 2000 wo sakujo shitai no desuga dou shitai desho ka?*"
[The underlined part was incorrectly recognized.]

Division into phrases:
"*Os IME desu ga / fuyou ni natta / IME 2000 wo / sakujo shitai no desuga / dou shitai desho / ka?*"

Calculation of perplexity (PP) and relevance score (RS):

phrase (its context)	PP	RS
(<S>) *Os IME desu ga* (*fuyou*)	2121.62	0.46
(*ga*) *fuyou ni natta* (*IME*)	21241.56	0.00
(*natta*) *IME 2000 wo* (*sakujo*)	349.24	0.99
(*wo*) *sakujo shitai no desu ga* (*dou*)	10.56	1.00
(*ga*) *dou shitai desho* (*ka*)	2459.01	0.36
(*desho*) *ka* (</S>)	3430.04	0.23

<S>, </S> denote the beginning and end of a sentence.

Fig. 4. Example of calculating perplexity (PP) and relevance score (RS).

Table 2
Examples of critical words confirmed prior to the retrieval

Office	VisualStudio	Word	IME
PowerPoint	InternetExplorer	Excel	setup
Outlook	OutlookExpress	font	backup
registry	printer	mail	form
folder	password	graph	cell

because a phrase with a low relevance score is likely to be an ASR error or a portion that does not contribute to the retrieval, even if it contains a content word.

3.2. Posterior confirmation using significance score (SS)

3.2.1. Definition of significance score

Our significance score is defined using plural retrieval results corresponding to the N-best hypotheses of ASR. Ambiguous portions during the ASR appear as differences between the N-best hypotheses. The significance score represents the degree to which the portion is actually influential to the retrieval by observing the difference of the retrieval results.

The procedure to calculate the significance score requires detection of different words between the N-best hypotheses. We define the significance score (SS) as the difference between the retrieval results of the n-th and m-th hypotheses based on the cosine distance, which is widely used to measure the distance between sparse vectors.

$$\mathrm{SS}(n,m) = 1 - \frac{|\mathrm{res}(n) \cap \mathrm{res}(m)|^2}{|\mathrm{res}(n)||\mathrm{res}(m)|}$$

Here, res(n) denotes the set of retrieved documents for the n-th query sentence, and $|\mathrm{res}(n)|$ denotes the number of documents in the set. That is, the significance score decreases if the two retrieval results have a large common portion.

3.2.2. Confirmation using significance score

The posterior confirmation is made based on the significance score. If the score is higher than a threshold θ_2, the system makes a confirmation by presenting the difference in the N-best list of ASR to the user. Otherwise, the system simply presents the retrieval result of the first hypothesis without making a confirmation. Here, we set $N = 3$ and

- How can I send a picture by Outlook?
- When executing Windows Update, the connection was lost.
- How can I watch DVDs?
- Um, I can't login to my Windows 2000 PC, as I've forgotten my password.
- I am using Outlook 2002, and I want to back up my e-mail, as there is a lot, some of which is very important.
- I am using Windows Me, Um the Windows shut-down button at upper right has corrupted characters, how can I fix it?

Fig. 5. Example user utterances (translation of Japanese).

the threshold θ_2 for the score to 0.5. In the confirmation phase, if the user selects from the list, the system displays the corresponding retrieval result. If the user judges all query sentences as inappropriate, the system rejects the current results and prompts him/her to utter the query again.

3.3. Experimental evaluation of confirmation strategy

We implemented and evaluated this confirmation strategy as the front-end of Dialog Navigator. The ASR system consists of Julius (Lee et al., 2001) for SAPI.[4] For language model training, we used several corpora: the knowledge base (Table 1), actual query sentences (typed input) put to Dialog Navigator, query sentences (typed input) put to another retrieval system[5] that was provided by Microsoft Japan, and transcripts of simulated spoken dialogue for software support. The total text size was about 6.9M words. A trigram language model was trained with a vocabulary of 18 K words.

We collected test data from 30 subjects who had not used our system before. Each subject was requested to retrieve support information for 14 tasks, which consisted of 11 prepared scenarios (query sentences not given) and three spontaneous queries. Subjects were told that the goal of the system was to carry out call center operations, and they were asked to utter queries as they would to a human operator. Subjects were allowed to utter a query sentence again up to twice per task if a relevant retrieval result was not obtained. As a result, we obtained 651 utterances for 420 tasks in total. Fig. 5 shows example user utterances. The average word accuracy of ASR was 76.8%.

3.3.1. Evaluation of retrieval success rate

First, we evaluated with the success rate of retrieval for the collected speech data. We regarded a retrieval as successful if the retrieval results contained a correct answer to the user's initial query. We compared the following cases:

(1) Transcript: A correct transcription of the user utterance, which was made manually, was used as input to Dialog Navigator.
(2) ASR result (baseline): The first hypothesis of ASR was used as input.
(3) Proposed method: Using the relevance and significance scores, the proposed confirmation strategy was adopted. Appropriate replies were chosen manually.

Table 3 lists the success rates for the three cases. The proposed method achieved a higher rate than the case where the first hypothesis of ASR was used. The improvement of 6.4% achieved by the proposed method was statistically significant ($p < .05$). When we broke down the improvement by incorporating the individual techniques, we identified 22% of the improvement was due to the prior confirmation,

Table 3
Success rates of retrieval

	Success rate (%)
Transcript	79.9
ASR result (baseline)	64.7
Proposed method	71.1

[4] http://julius.sourceforge.jp/sapi/.
[5] http://www.microsoft.com/japan/enable/nlsearch/.

Table 4
Comparison with method using ASR confidence measure (CM)

	# Confirmation (per dialogue)	Success rate (%)
Proposed method	0.34	71.1
CM ($\theta_1 = 0.4$)	0.12	65.6
CM ($\theta_1 = 0.6$)	0.39	66.8
CM ($\theta_1 = 0.8$)	0.74	68.4

64% due to the posterior confirmation, and 14% due to the weighting during the matching using the relevance score. Thus, the proposed confirmation strategy is effective in improving the task achievement.

3.3.2. Evaluation of efficiency of confirmation

We also evaluated in terms of the number of generated confirmations using the dialogue transcripts of the previous evaluation. The proposed method generated 221 confirmations. This means confirmations were generated once every three utterances, on average. The 221 confirmations consisted of 66 prior to the retrieval using the relevance score and 155 posterior to the retrieval using the significance score.

We compared the proposed method with a conventional method that used a confidence measure of ASR. The confidence measure (CM) was computed as a posteriori probability by using the N-best hypotheses of ASR (Komatani and Kawahara, 2000). In this method, the system generated confirmations only for content words having confidence measures lower than θ_1. The threshold to generate confirmation (θ_1) was set to 0.4, 0.6, or 0.8. In this simulation, the right answer to the confirmation was given manually.

The number of confirmations and retrieval success rates are shown in Table 4. The proposed method achieved a higher success rate with fewer confirmations (less than half) compared with the case of $\theta_1 = 0.8$ for the conventional method. Thus, the proposed confirmation strategy is more efficient, because it considers the influence to the retrieval.

4. Dialogue strategy to clarify user's vague queries

We have started a field trial of the system at our university. In the queries we collected so far, there are a number of vague ones such as "I cannot print." and one-word inputs such as "Mail". Moreover, important information is sometimes lost due to ASR errors, even when the query utterance is specific enough. In such cases, an enormous list of possible relevant documents is usually obtained. Therefore, it is necessary to narrow down the documents by clarifying the user's intention through dialogue. However, it is not possible in general document retrieval tasks to prepare a specific dialogue flow beforehand.

In the proposed framework, the system generates optimal questions by dynamically selecting from a pool of possible candidates. The information gain (IG) is defined as the criterion for the selection.

4.1. Dialogue strategy based on information gain (IG)

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 clearly partitions the document set in a completely disjoint 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 these versions. Moreover, the matching score of the retrieved documents should be taken into account. Therefore, we define IG $H(S)$ for a candidate question S as follows:

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

Here, n denotes the number of categories classified by the candidate question S. The documents that are not related to any category are classified as category 0. $P(i)$ is calculated by normalizing the number of matched documents for each category weighted by their matching scores:

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

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

Here, D_k denotes the k-th retrieved document obtained 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 system flow incorporating this strategy is summarized below and also shown in Fig. 6:

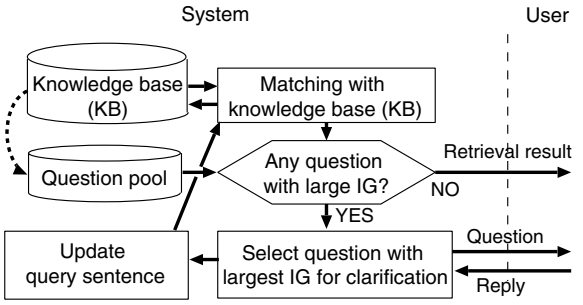

Fig. 6. Overview of query clarification.

(1) For a query sentence, make a retrieval from the KB.
(2) Calculate IG for all possible candidate clarification questions which satisfy a precondition (described below).
(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 by using the user's reply to the question.
(5) Return to (1).

This procedure is explained in detail in the following sub-sections.

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

4.2.1. Questions based on dependency structure analysis (method 1)

This type of questions are intended to clarify the modifiers or objects 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 determined 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 the KB and query sentences (typed input).
(2) Calculate the entropy $H(m)$ for every word based on the probability $P(i)$. $P(i)$ is calculated from 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

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

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

As a result, we selected 40 words that had a large value of entropy. The system makes questions when these words are included in the user's query. Table 5 lists examples of candidate clarification questions using this method. In this table, the percentage of applicable documents corresponds to those including the words selected, and IG is calculated using the applicable documents.

Table 5
Examples of candidate questions (dependency structure analysis: method 1)

Question	Precondition	Percentage 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

4.2.2. Questions based on metadata included in the KB (method 2)

We also prepare candidate questions by using the metadata attached to the KB. In general, metadata is usually attached to large-scale KBs in order 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 consists of product names to which the document applies. Using this metadata, the system can generate questions to identify the product to which the user's query corresponds. However, some documents are related with multiple versions, or may not belong to any category. Therefore, the effectiveness of these questions greatly depends on the characteristics of the metadata.

Fourteen candidate questions were prepared using this method. Table 6 lists examples of these candidate questions. The percentage of applicable documents corresponds to those having the metadata related to the target products.

4.2.3. Questions based on human knowledge (method 3)

Software support is conventionally provided by human operators at call centers. We therefore prepare candidate questions based on the human knowledge that has been accumulated there. In particular, we include three questions written by an expert. For instance, the question "When did the symptom occur?" tries to capture key information to identify relevant documents. The categories for the IG calculation were defined using handcrafted rules by focusing on key-phrases such as "after..." or "during...". Table 7 lists the candidate questions.

Fig. 7 shows an example dialogue where the system asks questions based on IG.

Table 6
Examples of candidate questions (metadata: method 2)

Question	Precondition	Percentage 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 7
List of candidate questions (human knowledge: method 3)

Question	Precondition	Percentage 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
Specifically, what do you want to do?	None	6.98	8.04

4.3. Update of retrieval query sentence

Through the dialogue to clarify the user's query, the system updates the query sentence by using the user's reply to the question. As described in Section 2.2, our back-end information retrieval system does not adopt a simple "bag-of-words" model, but conducts a more precise dependency structure analysis for matching; therefore, forming an appropriate query sentence is more desirable than simply adding keywords. Moreover, it is more comprehensible to present the updated query sentence than to show the sequence of ASR results. The update rules of the query sentence are as follows:

(1) Questions based on dependency structure analysis.
    The user's reply is added immediately before or 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 in the 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.

An example dialogue where the system updates the user's query is shown in Fig. 8. In the example, the system makes a 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.

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".

Fig. 7. Example dialogue.

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".

Fig. 8. Query sentence update using user's reply.

- **An example 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 user 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.

Fig. 9. Example of scenario and user utterances.

4.4. Experimental evaluation

We implemented and evaluated the proposed method. We collected new test data from 14 subjects who had not used our system.[6] Each subject was requested to retrieve support articles for 14 tasks, which consisted of prepared scenarios (query sentences 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 ($=14\times14$) tasks in total. Fig. 9 shows an example of the scenario and user utterances. The average word accuracy of ASR was 82.9%. The threshold value of IG for the system to make a clarification question was set to 1.0 initially, and incremented by 0.3 every time the system generated a question through a dialogue session.

First, we evaluated the success rate of retrieval. We regarded a retrieval as successful when the retrieval result[7] contained a correct document entry for the scenario. We compared the following cases:

(1) Transcript: A correct transcription of the user utterance, which was made manually, was used as input.
(2) ASR result (baseline): The first hypothesis of ASR was used as 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 case.
(4) Proposed method (simulation): The system generated questions based on the proposed method, and appropriate answers were given manually.

Table 8 lists the retrieval success rate and the rank of the correct document in the retrieval result for the four cases. The proposed method achieved a better success rate than when the ASR result was used. The improvement of 12.6% for the simulation was statistically significant ($p<.01$), and 7.7% for the log data was statistically significant ($p<.05$). These figures demonstrate the effectiveness

Table 8
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

[6] The test data and subjects are different from those described in Section 3.3. In Section 3.3, the subjects were asked to utter as if they were talking to a human operator, but for evaluation described in this section, such instructions were not given.

[7] As described in Section 2.2, the retrieved documents were ordered by their matching scores and their number was restricted to about 15, since the user had to read detailed information about the retrieved documents by clicking on their icons.

of the proposed method. The success rate of retrieval in the simulation was about 5% higher than that of the log data. This difference is considered to be caused by the following factors:

(1) ASR errors in the uttered replies.
The retrieval sentence is updated with the user's reply to the question regardless of whether the reply has ASR errors. Even when the user notices the ASR errors, he/she cannot correct them. Although it is possible to confirm them by using ASR confidence measures, this would make the 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 users replied "just now" instead of giving exact information for the retrieval. To solve this problem, it may be necessary to make the questions more specific or to display example replies.

We also evaluated the efficiency of the individual methods by simulation. In this experiment, each of the three methods was adopted to generate questions. The results are in Table 9. The improvement rates of the three methods did not differ very much, and the 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.

Table 9
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

Finally, we evaluated the success rate by using the confirmation strategy proposed by Section 3 together. Here, we conducted a simulated experiment against the test data used in this section, because the same test data should be used to measure the additional gain, and it was not possible to collect data from the same subjects for responding to the clarification dialogue. As a result, we obtained an overall improvement of 14.2% absolute (84.9% success rate) against the baseline case of ASR result. The effect of combination is confirmed, though it is not large.

We assume the small gain by adding the confirmation strategy was due to the difference in the styles of the test data. In Section 3, the subjects were asked to utter as if they were talking to a human. The utterances were long, and thus the system confirmed many critical words that were incorrectly recognized. On the other hand, queries in this section were relatively short and included fewer critical words to be confirmed. The confirmation strategy worked well for long utterances in this test data.

5. Conclusion

We have addressed an efficient dialogue strategy for document retrieval tasks, and have approached the major problems caused by speech input: ASR errors, redundancies in spontaneous speech, and the vagueness of the user's query. We first introduced two measures of the relevance score and the significant score, so that the system would generate confirmations to handle recognition errors, prior to and after the retrieval, respectively. An experimental evaluation in the retrieval from a software support knowledge base (KB) showed that the proposed method generates confirmations more efficiently for better task achievement compared with a method using the conventional confidence measure of ASR. We also proposed a dialogue strategy for clarifying vague queries. Candidate questions were prepared based on the dependency structure analysis of the KB together with the KB metadata and human knowledge. The system selected an optimal clarification question based on information gain (IG). The query sentence was then updated using the user's reply. Another experimental evaluation showed that the proposed method significantly improved the success rate of retrieval, and all three types 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.

Acknowledgements

We would like to thank Prof. Kurohashi and Dr. Kiyota at University of Tokyo and Dr. Komatani at Kyoto University for their helpful advice. We are also in debt to Ms. Kido formerly at Microsoft Corporation. We also would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

Allen, J.F., Miller, B.W., Ringger, E.K., Sikorski, T., 1996. A robust system for natural spoken dialogue. In: Proc. of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96). pp. 62–70.

Barnett, J., Anderson, S., Broglio, J., Singh, M., Hudson, R., Kuo, S.W., 1997. Experiments in spoken queries for document retrieval. In: Proc. Eurospeech.

Bennacef, S., Devillers, L., Rosset, S., Lamel, L., 1996. Dialog in the RAILTEL telephone-based system. In: Proc. ICSLP.

Bouwman, G., Sturm, J., Boves, L., 1999. Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project. In: Proc. IEEE-ICASSP.

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

Denecke, M., Waibel, A., 1997. Dialogue strategies guiding users to their communicative goals. In: Proc. Eurospeech.

Fujii, A., Itou, K., 2003. Building a test collection for speech-driven Web retrieval. In: Proc. Eurospeech.

Harabagiu, S., Moldovan, D., Picone, J., 2002. Open-domain voice-activated question answering. In: Proc. COLING. pp. 502–508.

Hazen, T.J., Burianek, T., Polifroni, J., Seneff, S., 2000. Integrating recognition confidence scoring with language understanding and dialogue modeling. In: Proc. ICSLP.

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

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

Komatani, K., Kawahara, T., 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In: Proc. COLING. pp. 467–473.

Komatani, K., Kawahara, T., Ito, R., Okuno, H.G., 2002. Efficient dialogue strategy to find users' intended items from information query results. In: Proc. COLING. pp. 481–487.

Lee, A., Kawahara, T., Shikano, K., 2001. Julius—an open source real-time large vocabulary recognition engine. In: Proc. Eurospeech.

Levin, E., Narayanan, S., Pieraccini, R., Biatov, K., Bocchieri, E., Fabbrizio, G.D., Eckert, W., Lee, S., Pokrovsky, A., Rahim, M., Ruscitti, P., Walker, M., 2000. The AT&T-DARPA Communicator mixed-initiative spoken dialogue system. In: Proc. ICSLP.

Lewis, C., Fabbrizio, G.D., 2005. A clarification algorithm for spoken dialogue system. In: Proc. IEEE-ICASSP.

NIST, DARPA, 2003. The twelfth Text REtrieval Conference (TREC 2003). In: NIST Special Publication SP 500-255.

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

Rayner, M., Hockey, B.A., Hieronymus, J., Dowding, J., Aist, G., Early, S., 2003. An intelligent procedure assistant built using REGULUS 2 and ALTERF. In: Proc. of 42nd Annual Meeting of the ACL.

Rudnicky., A., Bennett, C., Black, A., Chotomongcol, A., Lenzo, K., Oh, Singh, A., 2000. Task and domain specific modelling in the Carnegie Mellon Communicator system. In: Proc. ICSLP, vol. 2.

San-Segundo, R., Pellom, B., Ward, W., Pardo, J., 2000. Confidence measures for dialogue management in the CU Communicator system. In: Proc. IEEE-ICASSP.

Schofield, E., Zheng, Z., 2003. A speech interface for open-domain question-answering. In: The Companion Volume to Proc. 41st Annual Meeting of the Association for Computational Linguistics. pp. 177–180.

Seneff, S., Polifroni, J., 2000. Dialogue management in the Mercury flight reservation system. In: Proc. ANLP-NAACL 2000, Satellite Workshop.

Stent, A., Dowding, J., Gawron, J.M., Bratt, E.O., Moore, R., 1999. The CommandTalk spoken dialogue system. In: Proc. of 37th Annual Meeting of the ACL.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L., 2000. Jupiter: a telephone-based conversational interface for weather information. IEEE Trans. on Speech Audio Process. 8 (1), 85–96.