Dialog Navigator : A Spoken Dialog Q-A System based on Large Text Knowledge Base

Yoji Kiyota, Sadao Kurohashi (The University of Tokyo)
{kiyota,kuro}@kc.t.u-tokyo.ac.jp
Teruhisa Misu, Kazunori Komatani, Tatsuya Kawahara (Kyoto University)
{misu,komatani,kawahara}@kuis.kyoto-u.ac.jp
Fuyuko Kido (Microsoft Co., Ltd.)
fkido@microsoft.com

Abstract

This paper describes a spoken dialog Q-A system as a substitution for call centers. The system is capable of making dialogs for both fixing speech recognition errors and for clarifying vague questions, based on only large text knowledge base. We introduce two measures to make dialogs for fixing recognition errors. An experimental evaluation shows the advantages of these measures.

1 Introduction

When we use personal computers, we often encounter troubles. We usually consult large manuals, experts, or call centers to solve such troubles. However, these solutions have problems: it is difficult for beginners to retrieve a proper item in large manuals; experts are not always near us; and call centers are not always available. Furthermore, operation cost of call centers is a big problem for enterprises. Therefore, we proposed a spoken dialog Q-A system which substitute for call centers, based on only large text knowledge base.

If we consult a call center, an operator will help us through a dialog. The substitutable system also needs to make a dialog. First, asking backs for fixing speech recognition errors are needed. Note that too many asking backs make the dialog inefficient. Secondly, asking backs for clarifying users' problems are also needed, because they often do not know their own problems so clearly.

To realize such asking backs, we developed a system as shown in Figure 1. The features of our system are as follows:

- Precise text retrieval.
  The system precisely retrieves texts from large text knowledge base provided by Microsoft Corporation (Table 1), using question types, products, synonymous expressions, and syntactic information. *Dialog cards* which can cope with very vague questions are also retrieved.
- Dialog for fixing speech recognition errors.
  When accepting speech input, recognition errors are inevitable. However, it is not obvious which portions of the utterance the system should confirm by asking back to the user. A great number of spoken dialog systems for particular task domains, such as (Levin et al., 2000), solved this problem by defining slots, but it is not applicable to large text knowledge base. Therefore, we introduce two measures of *confidence in recognition* and *significance for retrieval* to make dialogs for fixing speech recognition errors.
- Dialog for clarifying vague questions.
  When a user asks a vague question such as "An error has occurred", the system navigates him/her to the desired answer, asking him/her back using both *dialog cards* and extraction of

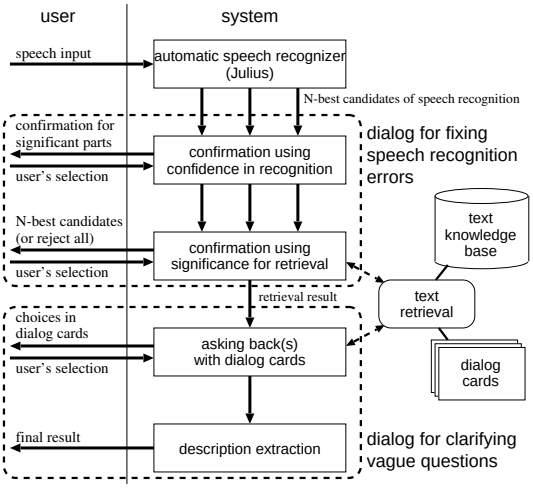


Figure 1: Architecture.

Table 1: Text collections.

text collection	# of texts	# of characters	matching target
Glossary	4,707	700,000	entries
Help texts	11,306	6,000,000	titles
Support KB	23,323	22,000,000	entire texts

summaries that makes differences between retrieved texts more clear.

Our system makes asking backs by showing them on a display, and users respond them by selecting the displayed buttons by mouses.

Initially, we developed the system as a keyboard based Q-A system, and started its service in April 2002 at the web site of Microsoft Corporation. The extension for speech input was done based on the one-year operation. Our system uses Julius (Lee et al., 2001) as a Japanese speech recognizer, and it uses language model acquired from the text knowledge base of Microsoft Corporation.

In this paper, we describe the above three features in Section 2, 3, and 4. After that, we show experimental evaluation, and then conclude this paper.

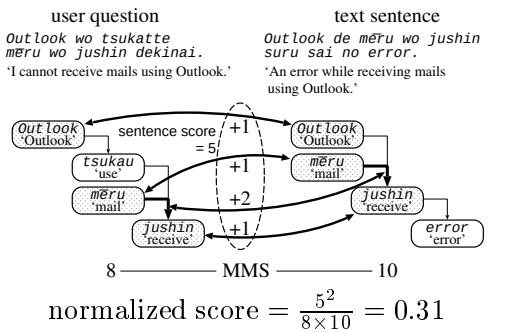
2 Precise Text Retrieval

It is critical for a Q-A system to retrieve relevant texts for a question precisely. In this section, we describe the score calculation method, giving large points to modifier-head relations between *bunsetsu*[1] based on the parse results of KNP (Kurohashi and Nagao, 1994), to improve precision of text retrieval. Our system also uses question types, product names, and synonymous expression dictionary as described in (Kiyota et al., 2002).

First, scores of all sentences in each text are calculated as shown in Figure 2. Sentence score is the total points of matching keywords and modifier-head relations. We give 1 point to a matching of a keyword, and 2 points to a matching of a modifier-head relation (these parameters were set experimentally). Then sentence score is normalized by the maximum matching score (MMS) of both sentences as follows (the MMS is the sentence score with itself):

$$\frac{(\text{sentence score})^2}{\left(\begin{array}{c}\text{the MMS of a}\\ \text{user question}\end{array}\right) \times \left(\begin{array}{c}\text{the MMS of a}\\ \text{text sentence}\end{array}\right)}$$

[1] *Bunsetsu* is a commonly used linguistic unit in Japanese, consisting of one or more adjoining content words and zero or more following functional words.



$$\text{normalized score} = \frac{5^2}{8 \times 10} = 0.31$$

Figure 2: Score calculation.

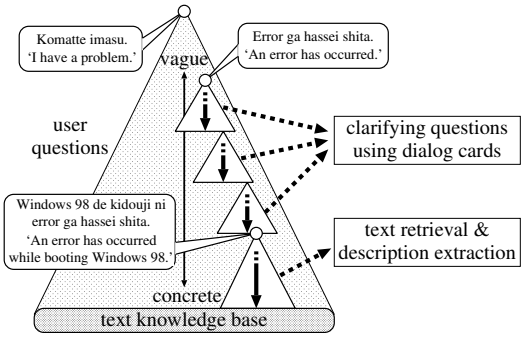


Figure 3: User navigation.

Finally, the sentence that has the largest score in each text is selected as the *representative sentence* of the text. Then, the score of the sentence is regarded as the score of the text.

3 Dialog Strategy for Clarifying Questions

In most cases, users' questions are vague. To cope with such vagueness, our system uses the following two methods: asking backs using *dialog cards* and extraction of summaries that makes difference between retrieved texts more clear (Figure 3).

3.1 Dialog cards

If a question is very vague, it matches many texts, so users have to pay their labor on finding a relevant one. Our system navigates users to the desired answer using *dialog cards* as shown in Figure 3.

We made about three hundred of dialog cards to throw questions back to users. Figure 4 shows two dialog cards. <UQ> (User Question) is followed by a typical vague user question. If a user question matches it, the dialog manager asks the back question after <SYS>, showing choices be-

[Error]
<UQ> *Error ga hassei suru*
'An error occurs'
<SYS> *Error wa itsu hassei shimasuka?*
'When does the error occurs?'
<SELECT>
Windows kidou ji goto [Error/Booting Windows]
'while booting Windows'
in'satsu ji goto [Error/Printing Out]
'while printing out'
application kidou ji goto [Error/Launching Applications]
'while launching applications'
</SELECT>

[Error/Booting Windows]
<UQ> *Windows wo kidou ji ni error ga hassei suru*
'An error occurs while booting Windows'
<SYS> *Anata ga otsukai no Windows wo erande kudasai.*
'Choose your Windows.'
<SELECT>
Windows 95 retrieve *Windows 95 wo kidou ji ni error ga hassei suru*
'An error occurs while booting Windows 95'
Windows 98 retrieve *Windows 98 wo kidou ji ni error ga hassei suru*
'An error occurs while booting Windows 98'
Windows ME retrieve *Windows ME wo kidou ji ni error ga hassei suru*
'An error occurs while booting Windows ME'
</SELECT>

Figure 4: Dialog cards.

tween <SELECT> and </SELECT>. Every choice is followed by goto or retrieve. goto means that the system follow the another dialog cards if this choice is selected. retrieve means that the system retrieve texts using the query specified there.

3.2 Description extraction from retrieved texts

In most cases, the neighborhood of the part that matches the user question describes specific symptoms and conditions of the problem users encounter. Our system extracts such descriptions from the retrieved texts as the summaries of them. The algorithm is described in (Kiyota et al., 2002).

4 Dialog Strategy for Speech Input

It is necessary for a spoken dialog system to determine which portions of the speech input should be confirmed. Moreover, criteria for judging whether it should make confirmation or not are needed, because too many confirmations make the dialog inefficient. Therefore, we introduce two criteria of *confidence in recognition* and *significance for retrieval*.

Our system makes two types of asking backs for fixing recognition errors (Figure 1). First, Julius outputs N-best candidates of speech recognition. Then, the system makes confirmation for significant parts based on *confidence in recognition*. After that, the system retrieves relevant texts in the text knowledge base using each candidate, and makes confirmation based on *significance for retrieval*.

4.1 Confidence in recognition

We define the *confidence in recognition* for each phrase in order to reject partial recognition errors. It is calculated based on word perplexity, which is often used in order to evaluate suitability of language models for test-set sentences. We adopt word perplexity because of the following reasons: incorrectly recognized parts are often unnatural in context, and words that are unnatural in context have high perplexity values.

As Julius uses trigram as its language model, the word perplexity PP is calculated as follows:

$$\log PP = -\frac{1}{n}\sum_{k} \log P(w_k|w_{k-1}, w_{k-2}).$$

PPs are summed up in each *bunsetsu* (phrases). As a result, the system assigned the sum of PPs to each *bunsetsu* as the criterion for *confidence in recognition*.

We preliminarily defined the set of product names as significant phrases[2]. If the sums of PPs for any significant phrases are beyond the threshold (now, we set it 50), the system makes confirmation for these phrases.

4.2 Significance for retrieval

The system calculates *significance for retrieval* using N-best candidates of speech recognition. Because slight speech recognition errors are not harmful for retrieval results, we regard a difference that affects its retrieval result as significant. Namely, when the difference between retrieval results for each recognition candidate is large, we regard that the difference is significant.

Significance for retrieval is defined as a rate of disagreement of five high-scored retrieved texts among N recognition candidates. For example, if there is a substituted part in two recognition candidates, and only one text is commonly retrieved out of five high-scored texts by both candidates, the *significance for retrieval* for the substituted part is 0.8 $(= 1 - 1/5)$.

The system makes confirmation which candidate should be used, if *significance for retrieval* is beyond the threshold (now, we set it 0.5).

[2]We are now developing a method to define the set of significant phrases semi-automatically.

Table 2: Number of successful retrieval for each speaker.

speaker ID	# of utterances	ASR corr.	transcription (1)	speech recognition results (2)	with confidence in recognition (3)	with significance for retrieval (4)	with both measures (5)
A	13	87.8%	10/13	8/13	8/13	10/13	10/13
B	14	55.0%	6/14	3/14	5/14	3/14	5/14
C	13	61.7%	5/13	5/13	5/13	5/13	5/13
D	13	83.1%	11/13	9/13	9/13	9/13	9/13
total	53	68.5%	32/53	25/53	27/53	27/53	29/53

U: OS は ME なのですが、使うには問題ないんですけれども、閉じるや最大化ボタンの表示が、えー、おかしな表示になってるんですけれど、どうにかならないんでしょうか
'I am using (Windows) ME as OS(operating system), then ...'
ASR: OS を IME のですが疲れ問題ないんですけども太字や最大化ボタンの表示がへおかしな表示になってるんですよどういかがでしょか
("ME" was incorrectly recognized as "IME".)
S: The underlined product name seems to be uncorrectly recognized. May I remove it, or leave it?
U: (selected "remove")
S: The following texts are retrieved.
1. [AC97] フォームを最大化したときに [閉じる] ボタンを非表示にする
2. [拡大鏡の設定] で [最大化] ボタンが使用できない
3. [WinMe] TrueType フォント キャッシュが破壊される (a correct answer)

U: Word 2002 で 数式 を入力する方法を教えてください
"Please tell me the way to input formulas in Word 2002."
ASR: 1. Word 2002 で 数字 (numbers) を入力する方法を教えてください
2. Word 2002 で 数式 (formulas) を入力する方法を教えてください
3. Word 2002 で 数値 (values) を入力する方法を教えてください
S: Please select the most correct recognition result from the above candidates.
U: (selected No. 2)
S: The following texts are retrieved.
1. Word で数式を挿入する (a correct answer)
2. Word で現在の日付と時刻を入力する
3. スプレッドシートで数式を入力する

Figure 5: Dialogs for fixing speech recognition errors.
(U: user, S: system, ASR: automatic speech recognition)

5 Experimental Evaluation

We evaluated the system performance experimentally. For the experiments, we had 4 subjects, who were accustomed to using computers. They made utterances by following given 10 scenarios and also made several utterances freely. In total, 53 utterances were recorded. Figure 5 shows two successful dialogs by confirmation using *confidence in recognition* and by that using *significance for retrieval*.

We experimented on the system using the 53 recorded utterances by the following methods:

(1) Using correct transcription of recorded utterance, including fillers.
(2) Using speech recognition results from which only fillers were removed.
(3) Using speech recognition results and making confirmation by *confidence in recognition*.
(4) Using N-best candidates of speech recognition and making confirmation by *significance for retrieval*. Here, $N = 3$.
(5) Using N-best candidates of speech recognition and both measures in (3) and (4).

In these experiments, we assumed that users always correctly answer system's asking backs. We regarded a retrieval as a successful one if a relevant text was contained in ten high-scored retrieval texts.

Table 2 shows the result. It indicates that our confirmation methods for fixing speech recognition errors improve the success rate. Furthermore, the success rate with both measures gets close to that with the transcriptions. Considering that the speech recognition correctness is about 70%, the proposed dialog strategy is effective.

6 Conclusion

We proposed a spoken dialog Q-A system in which asking backs for fixing speech recognition errors and those for clarifying vague questions are integrated. To realize dialog for fixing recognition errors based on large text knowledge base, we introduced two measures of *confidence in recognition* and *significance for retrieval*. The experimental evaluation shows the advantages of these measures.

References

Yoji Kiyota, Sadao Kurohashi, and Fuyuko Kido. 2002. "Dialog Navigator" : A Question Answering System based on Large Text Knowledge Base. In *Proceedings of COLING 2002*, pages 460–466.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4).

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

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 *Proceedings of Int'l Conf. Spoken Language Processing (ICSLP)*.