“Dialog Navigator” : A Question Answering System based on Large Text Knowledge Base

Yoji Kiyota[†], **Sadao Kurohashi**[‡§] and **Fuyuko Kido**[¶]
[†]Graduate School of Informatics, Kyoto University
[‡]Graduate School of Information Science and Technology, The University of Tokyo
[§]PRESTO, Japan Science and Technology Corporation(JST)
[¶]Microsoft Co., Ltd.
{kiyota,kuro}@kc.t.u-tokyo.ac.jp, fkido@microsoft.com

Abstract

This paper describes a dialog based QA system, Dialog Navigator, which can answer questions based on large text knowledge base. In real world QA systems, vagueness of questions is a big problem. Our system can navigates users to the desired answers using the following methods: asking users back with dialog cards, and description extraction of each retrieved text. Another feature of the system is that it retrieves relevant texts precisely, using question types, synonymous expression dictionary, and modifier-head relations in Japanese sentences.

1 Introduction

The best way to study something is to ask an expert. There are often gaps between what he wants to know and the answer, in several aspects, such as concreteness, expression and assumption. An expert can answer the question by overcoming such gaps through the conversation with the questioner.

Huge amount of texts are available on the world wide web these days, and an answer to any question potentially exists somewhere on the web. However, searching the web is far away from the effectiveness of asking an expert. The biggest problem is that machines cannot talk with people to overcome the gaps.

In case of web search engines, an enormous list of possibly relevant texts are returned, annoying users. In case of the QA track of TREC, systems find an answer (passage of 50 or 250 bytes) among texts and return it to the questioner. This seems an attractive ability, but it is assumed that questions are well specific. That is, questions are so clear that the answer can be found by one-step retrieval. However, in most cases a user cannot express his/her question in such a clear way.

We preliminarily examined the question logs of the natural language based text retrieval system[1] serviced by Microsoft Japan, and categorized questions. The examination shows that there are three major types of questions: *symptom*, *how* and *what*. It also shows that about 25% of the user questions are vague. It indicates that dialogs for clarifying questions are strongly required.

This paper describes a dialog based QA system, Dialog Navigator, which can answer questions based on large text knowledge base. The features of our system are as follows:

- Precise text retrieval.
  The system precisely retrieves relevant texts using several methods. First, it selects text collections by three question type. Secondly, it resolves the expression gaps between the question and texts using *synonymous expression dictionary*. Then it can detect plausible texts by giving larger weights to matching of modifier-head relations such as “open $\rightarrow$ file”.
- User navigation.
  When a user asks a vague question, the system navigates him/her to the desired answers, asking him/her back by the following methods: *dialog cards* and *description extraction*. Dialog cards are made for typical vague questions. If the user question matches a card, the system asks him/her back for clarifying the question, according to the card. Description extraction is done after text retrieval. The system extracts the neighborhood of the part which matches the user question, because it is usually important for asking back.

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

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

Recycle Bin Settings Not Retained During Windows 2000 Upgrade (Q240433)

The information in this article applies to:

- Microsoft Windows 2000, Advanced Server
- Microsoft Windows 2000, Professional
- Microsoft Windows 2000, Server

SYMPTOMS
After you upgrade to Windows 2000, from Microsoft Windows NT 4.0, Microsoft Windows 95, or Microsoft Windows 98, your settings for the Recycle Bin may be different than they were before you upgraded.

CAUSE
This issue can occur because the settings for the Recycle Bin are not migrated during the upgrade.

RESOLUTION
To resolve this issue, right-click **Recycle Bin**, click **Properties**, configure the settings you want, and then click **OK**.

MORE INFORMATION
The Recycle Bin settings for Windows 95 and Windows 98 are in the registry. The Recycle Bin settings for Windows NT 4.0 are kept on disk.

Figure 1: Microsoft Support KB (an English version).

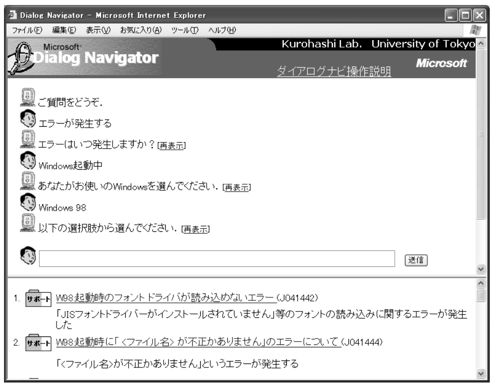
2 The Architecture of Dialog Navigator

Dialog Navigator is a dialog based QA system about Microsoft products. Table 1 shows the text collections that Dialog Navigator uses and their scales. Figure 1 shows an example of Microsoft Support KB (Knowledge Base).

The system consists of the following components:

User Interface: Users access to the system via a WWW browser. On the user interface, dialogs between a user and the system is displayed in the upper frame as shown in Figure 2. After a user inputs a question into the text box, the system asks him/her back by showing choices in the lower frame.

Input Analyzer: The user question is transformed into a dependency structure by a robust parser, KNP (Kurohashi and Nagao, 1994), and question-pattern rules are applied to extract the question type (*symptom* type, *how* type, or *what* type) and the question content. If no rule is applied, *no type* is extracted.


Dialog Navigator - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
Microsoft Dialog Navigator
Kurohashi Lab. University of Tokyo
ダイアログナビ操作説明
Microsoft
送信
サポート

ご質問をどうぞ.
エラーが発生する
エラーはいつ発生しますか？[再表示]
Windows起動中
あなたがお使いのWindowsを選んでください. [再表示]
Windows 98
以下の選択肢から選んでください. [再表示]

1. W98:起動時のフォント ドライバが読み込めないエラー (J041442)
「JISフォントドライバーがインストールされていません」等のフォントの読み込みに関するエラーが発生した
2. W98:起動時に「 <ファイル名> が不正かありません」のエラーについて (J041444)
「<ファイル名>が不正かありません」というエラーが発生する

Figure 2: The user interface.

Japanese is head-final, then the end of a sentence shows its question type. Hence the longest matching of question pattern rules from the end of the question can detect the type in most cases. For example, if the end of a question is *ga dekimasen deshita* 'I couldn't $\cdots$', *symptom* type is extracted.

Text Retriever: It makes the user question match texts, then it returns plausible texts as answer candidates.

Description Extractor: It extracts the description of each answer candidate, eliminating verbose portions of a sentence.

Dialog Manager: When the user inputs a vague question, it asks him/her back by showing choices using *dialog cards*.

3 Text Retriever

It is critical for a QA system to retrieve relevant texts for a question precisely.

The text retriever calculates the score of each text through the following processes. Finally, it selects texts which have high scores as the answer candidates.

3.1 Selection of text collections by question types

As we have mentioned in Section 2, Input analyzer extracts question types (*symptom*, *how*, or *what*) of the user input. The text collections can be also classified into those three types. It

Table 2: Selection of text collections by question types.

text collection		question type *what*	*how*	*symptom*	*no type*
Glossary	*what*	o			
Help texts	*how*	o	o		o
Support KB	*symptom*	o		o	o
	how	o	o		o
	no type	o	o	o	o

seems that Glossary corresponds to *what* type, Help texts to *how* type, and Support KB to *how* or *symptom* type. Hence we can precisely select texts by those types.

The text retriever selects texts according to Table 2. For example, if *how* type is extracted from a user question, Help texts and Support KB (excluding *symptom* type) are selected.

3.2 Expression gap resolution by synonymous expression dictionary

The expression gap between user questions and texts is a big problem. In addition to synonyms, there are a great deal of synonymous phrases such as "boot a PC", "boot a Windows", and "switch on".

We made the *synonymous expression dictionary* which contains such synonymous phrases. The text retriever resolves the expression gap between user questions and texts using the dictionary.

3.2.1 Synonymous expression dictionary

We analyzed question logs of the search engine for Microsoft Support KB, then we made a dictionary of frequent synonymous expressions. Figure 3 shows a few entries of the dictionary.

As shown in Figure 3, the dictionary contains both synonyms and synonymous phrases.

3.2.2 Synonymous expression database

As shown in Figure 3, the dictionary contains recursive relations. Consider *mēru wo yomu* 'read a mail'. It contains two content words: *mēru* 'mail' and *yomu* 'read'. *mēru* has synonyms such as *meiru* and *e-mail*. Also *yomu* has synonyms such as *yomikomu*.

The system expands such recursive relations as shown in Figure 4. All the synonymous expressions in the dictionary are registered with

[*hassei*]
hassei, *okiru*, *okoru*
'occur', 'occur', 'occur'
[*yomu*]
yomu, *yomikomu*
'read', 'read .. into'
[*mēru*]
mēru, *meiru*, *e-mail*
'mail', 'mail', 'e-mail'
[*mēru wo yomu*]
mēru wo yomu, *mēru wo jushin suru*,
'read a mail', 'receive a mail',
message wo yomu, *message wo jushin suru*,
'read a message', 'receive a message',
[*pasokon wo kidou suru*]
pasokon wo kidou suru, *Windows wo kidou suru*,
'boot a PC', 'boot a Windows',
dengen wo ireru
'switch on'

Figure 3: Synonymous expression dictionary.

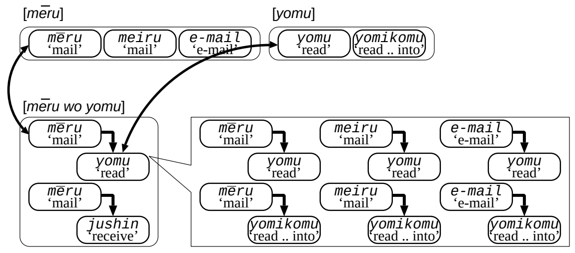


Figure 4: Expansion of recursive relations.

the *synonymous expression database*, including expanded expressions.

3.2.3 Extraction of synonymous expressions

The text retriever makes both the question and texts match the synonymous expression database considering modifier-head relations,

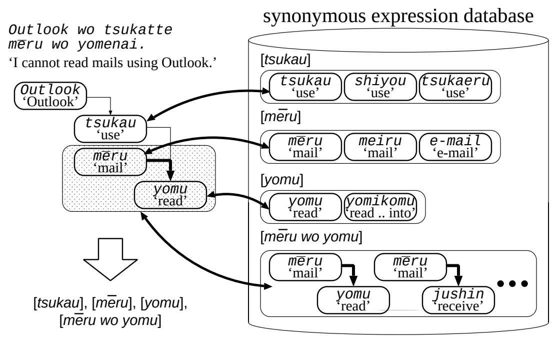


Figure 5: Making a user question match synonymous expression database.

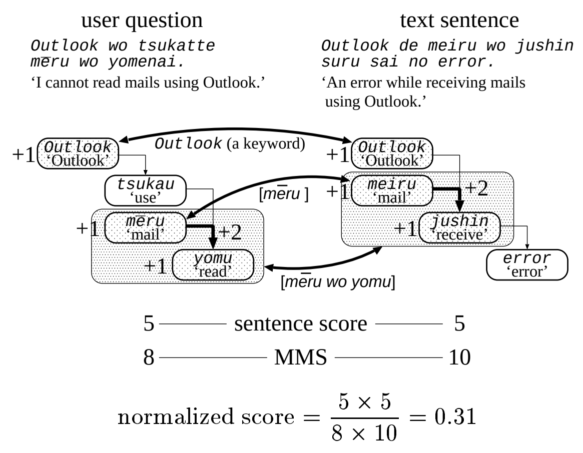

user question
Outlook wo tsukatte mēru wo yomenai.
'I cannot read mails using Outlook.'
text sentence
Outlook de meiru wo jushin suru sai no error.
'An error while receiving mails using Outlook.'
Outlook (a keyword)
+1 Outlook 'Outlook'
tsukau 'use'
+1 mēru 'mail'
+2
+1 yomu 'read'
[mēru]
+1 meiru 'mail'
+2
+1 jushin 'receive'
error 'error'
[mēru wo yomu]
5 sentence score 5
8 MMS 10
normalized score = 5 × 5 / 8 × 10 = 0.31

Figure 6: Score calculation.

then extracts synonymous expressions in them.

Figure 5 shows an example of a user question matching with the database. In this case, four synonymous expressions match: *tsukau* 'use', *mēru* 'mail', *yomu* 'read', and *mēru (wo) yomu* 'read a mail'.

The system extracts entries of matched expressions. In Figure 5, four entries [*tsukau*] [*mēru*] [*yomu*] [*mēru wo yomu*] are extracted. If a entry is extracted from both a text and a user question, they match.

3.3 Indexing

To retrieve texts fast, we create an index of keywords and entries of synonymous expressions in advance.

The text retriever extracts keywords and entries of synonymous expressions from the user question, then looks up them in the inverted index. It selects texts which contain at least one keyword or synonymous expression.

3.4 Score calculation

The score of each text is calculated as similarity between the question and a sentence in the text. We give large points to modifier-head relations to improve precision of text retrieval.

First, scores of all sentences in a text are calculated as shown in Figure 6. Sentence score is the total points of matching keywords and modifier-head relations both on the question and on the sentence. We give 1 point to a matching of a keyword, and 2 points to a matching of a modifier-head relation. Those

Table 3: Parameters for selecting candidates for a reply.

Text collection	n	t
Glossary	2	0.8
Help texts	5	0.3
Support KB	10	0.1
<UQ> of Dialog cards (Subsection 4.2)	1	0.8

which match as synonymous expressions are also given same points. Then each sentence score is normalized by the maximum matching score (MMS) as follows (the MMS is the sentence score with the same sentence):

$$\frac{\begin{pmatrix}\text{sentence score}\\\text{on a user question}\end{pmatrix} \times \begin{pmatrix}\text{sentence score}\\\text{on a text sentence}\end{pmatrix}}{\begin{pmatrix}\text{the MMS of a}\\\text{user question}\end{pmatrix} \times \begin{pmatrix}\text{the MMS of a}\\\text{text sentence}\end{pmatrix}}$$

Finally, the sentence which has the largest score in a text is selected as the **representative sentence** of the text. And the score of the sentence is regarded as the score of the text.

3.5 Candidate selection for a reply

The text retriever selects several texts as the candidates for a reply based on scores of texts in each text collection.

It sorts texts in order of their scores, then selects top n texts as the candidates. However, texts which have scores less than t are excluded. Parameters n, t are set as shown in Table 3.

If the candidates are selected from more than one collection, the system shows them to the user in order of Glossary, Help texts, and Support KB.

4 User Navigation

In most cases, there are several kinds of gaps between the user question and the answer. Consider a vague question "An error has occurred". It matches many texts. If the system shows all the texts at once, it is hard for a user to find a relevant one for the problem he/she is coming up with. There is a great demand for the ability to navigate him/her to the desired answer, asking him/her back "When did the error occur?", "Which version of Windows are you using?", or "What kind of error message was displayed?".

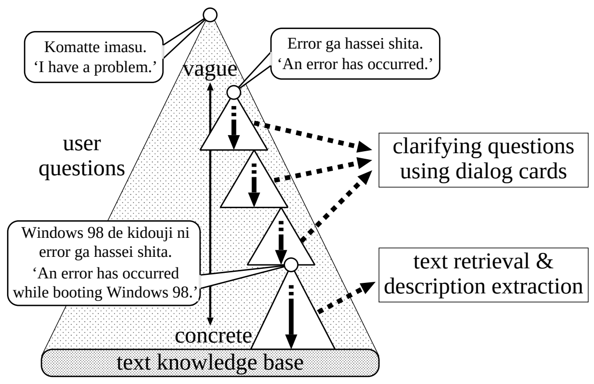

Komatte imasu.
'I have a problem.'
Error ga hassei shita.
'An error has occurred.'
vague
user questions
clarifying questions using dialog cards
Windows 98 de kidouji ni error ga hassei shita.
'An error has occurred while booting Windows 98.'
text retrieval & description extraction
concrete
text knowledge base

Figure 7: User Navigation.

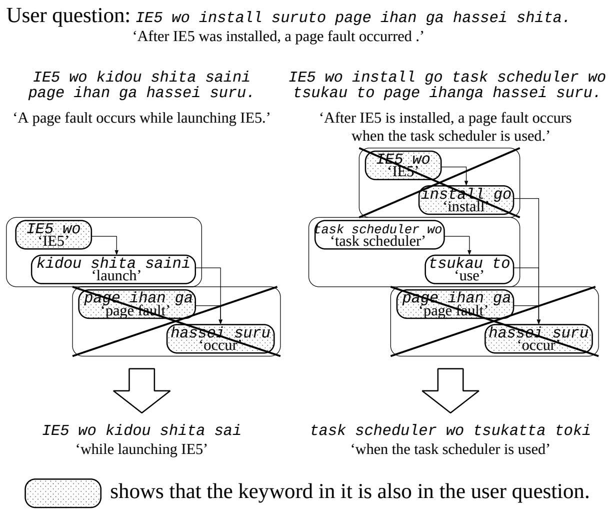

User question: IE5 wo install suruto page ihan ga hassei shita.
'After IE5 was installed, a page fault occurred .'
IE5 wo kidou shita saini page ihan ga hassei suru.
'A page fault occurs while launching IE5.'
IE5 wo install go task scheduler wo tsukau to page ihanga hassei suru.
'After IE5 is installed, a page fault occurs when the task scheduler is used.'
IE5 wo 'IE5'
install go 'install'
IE5 wo 'IE5'
kidou shita saini 'launch'
task scheduler wo 'task scheduler'
tsukau to 'use'
page ihan ga 'page fault'
hassei suru 'occur'
page ihan ga 'page fault'
hassei suru 'occur'
IE5 wo kidou shita sai
'while launching IE5'
task scheduler wo tsukatta toki
'when the task scheduler is used'
shows that the keyword in it is also in the user question.

Figure 8: Description extraction of answer candidates.

The system navigates a user as shown in Figure 7. If a user asks very vague question such as *Error ga hassei shita* 'An error has occurred', the system asks him/her back using **dialog cards**, until the user question is clarified. After that, it retrieves relevant texts, and extracts **descriptions** of them.

4.1 Description extraction of answer candidates

In most cases, the neighborhood of the part which matches the user question describes specific symptoms and conditions of the problem users often encounter. We developed a module which extracts such descriptions from the retrieved candidates.

The module extracts descriptions of answer candidates as shown in Figure 8. In this figure, a user question matches two sentences of candidates. Then each sentence is segmented into a few parts based on several criteria. As a result, each segmented part is a clause. Next, the module deletes parts in which all keywords are also in the user question. In the figure, one part in the left sentence and two parts in the right sentence are deleted. Finally, *IE5 wo kidou shita sai* 'while launching IE5' and *task scheduler wo tsukatta toki* 'when the task scheduler is used' are extracted as descriptions.

For Support KB, the description extractor extracts the description from the representative sentence of each candidate. For Glossary and Help texts, their titles themselves are good descriptions, so the module is not used.

```
[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 9: Dialog cards.

4.2 Dialog cards

If the user question is vague, it is too difficult to detect matching parts with the user question, then the description extraction often fails. Moreover, such a question matches many texts, so users have to pay their labor on finding a relevant one.

We therefore made several **dialog cards** to throw questions back to users. Figure 9 shows two examples of dialog cards. <UQ> (User Question) is followed by a typical vague user question. If a user question matches it, the

S: *goshitsumon wo douzo.*
  ‘Please ask a question.’
U: *Error ga hassei shita.*
  ‘An error has occurred.’
S: *Error wa itsu hassei shimasuka?*
  ‘When does the error occurs?’
  1. *Windows kidou ji* ‘while booting Windows’
  2. *insatsu ji* ‘while printing out’
  3. *application kidou ji* ‘while launching applications’
  4. · · ·
U: *Windows kidou ji*
  ‘While booting Windows’
S: *Anata ga otsukai no Windows wo erande kudasai.*
  ‘Choose your Windows.’
  1. *Windows 95*
  2. *Windows 98*
  3. *Windows ME*
  4. · · ·
U: *Windows 95*
  ‘Windows 95’
S: *Ika no genshou no izure ni ataru ka erande kudasai.*
  ‘Choose your symptoms.’
  1. *“. . .” to iu error ga hassei suru*
     ‘An error “. . .” occurs’
  2. *font no yomikomi ni kansuru error ga hassei shita*
     ‘An error about reading fonts has occurred’
  3. · · ·

Figure 10: A dialog using dialog cards.

dialog manager asks the back question after <SYS>, showing choices between <SELECT> and </SELECT>. Every choice is followed by `goto` or `retrieve`. `goto` means that the system should follow the another dialog cards if this choice is selected. `retrieve` means that the system should retrieve texts using the following phrase.

Figure 10 shows a dialog between a user and the system according to the cards in Figure 9. First, a user asks “An error has occurred”, then the system makes the question match <UQ> parts of dialog cards through the processes described in Subsection 3.2 $\sim$ 3.5. It selects a card which has score no less than 0.8, then the [Error] card is selected. According to the card, the system asks back when the error occurs, showing choices. Then, he/she selects “while booting Windows”, so control of the system moves to the [Error/Booting Windows] card. Then the system asks back again which Windows he/she uses. Then, he/she selects “Windows 95”, therefore the system sends a query “An error occurs while booting Windows 95” to the text retriever. Finally, descriptions of the retrieved candidates are extracted. In Figure 10, verbose parts are eliminated from 1st and 2nd choices.

As the dialog shows, the system navigates users both by using dialog cards and by extracting descriptions.

5 Evaluation

Dialog navigator started its service on April 2002 at `http://www.microsoft.com/japan/navigator/`. All conversation logs have been stored as a dialog database.

We randomly selected some dialogs in the dialog database, then we segmented each dialogs into task units manually. We call this unit a *session*. As a result, we got 603 sessions. On average, one session consists of 1.5 turns.

A subject evaluates these sessions based on the following criteria.

Success: The system could return a satisfactory answer. It means that the system returned at least one relevant text.

Failure A: The system had no relevant texts, and it answered that no relevant texts were found.

Failure B: The system had no relevant texts, but it returned some irrelevant texts.

Failure C: The system had relevant texts, but it could not return any of them.

Miscellaneous: Out of the system domain.

Table 4 shows the results. The success ratio is 52%, excluding Miscellaneous. Meanwhile, from the point of view of system evaluation, Failure A is also regarded as the success. If Failure A is regarded as the success, the success ratio is 66%.

The dialog cards were used in 12 of 313 sessions (excluding Miscellaneous), and all of these sessions are Success. Assuming all the 12 sessions failed without dialog cards, the improvement of the success ratio achieved by dialog cards is +3.8% (12/313).

To evaluate the description extraction, we randomly selected 100 user questions which were replied with more than 5 candidates. Then the descriptions of Support KB in top 5 candidates were judged by a subject into three categories: Proper, Insufficient, and Verbose. 152 candidates from Glossary and Help texts were excluded, because they were simply showed with the titles. As a result, 348 ($= 100 \times 5 - 152$) descriptions were evaluated.

Table 5 shows the results. 61% of the extracted descriptions were proper. Assuming all the descriptions (titles) of Glossary and Help texts were proper, 73% of the descriptions would have been proper.

Table 4: Evaluation of dialog sessions.

evaluation	# of sessions
Success	162 (52% / 27%)
Failure A	44 (14% / 7%)
Failure B	54 (17% / 9%)
Failure C	53 (17% / 9%)
subtotal	313 (100% / 52%)
Miscellaneous	290 (—— / 48%)
total	603 (—— / 100%)

Table 5: Evaluation of the description extraction.

evaluation	# of candidates
Proper	213 (61%)
Insufficient	27 (8%)
Verbose	108 (31%)
total	348(100%)

6 Related Works

The most serious problem for QA systems is the amount of knowledge. It is obvious that without sufficient knowledge a system cannot reply satisfactory answers.

Another serious problem is that user questions are not always complete. To solve the problem, the system have to be able to answer the user back. In other words, a dialog between a user and a system is required.

Classical QA systems such as UC (Wilensky et al., 1984) were capable of asking back. But they utilized formal languages to represent knowledge, which requires the heavy cost of construction and maintenance and makes the scaling up quite difficult.

In contrast, along with the improvement of NLP, research activities which utilize natural language text as a knowledge base become popular, such as FAQ Finder (Hammond et al., 1995) and the systems of TREC-9 QA Track (NIST and DARPA, 2001) (Harabagiu et al., 2001). These systems, however, basically produce one time response, and do not have a dialog with users.

Dialog Helpsystem of CIMS, Kyoto University (Kurohashi and Higasa, 2000) realized basic dialogs based on flexible matching of user questions with natural language knowledge base. The main difference between our system and their system is the scale of knowledge. The knowledge base of their system was constructed for it manually, then it needs some cost. Our system utilizes existing knowledge base, resolving expression gaps between user questions and knowledge base using the synonymous expression dictionary. The dictionary is constructed universally, therefore we will need little cost to utilize large knowledge base.

7 Conclusion

This paper described a dialog based QA system which utilizes large text knowledge base. We evaluated the system, then we concluded that 66% of the sessions were successful and that 73% of the descriptions of candidates will be helpful for users. We also estimated that the improvement of the success ratio achieved by dialog cards was +3.8%.

References

K. Hammond, R. Burke, C. Martin, and S. Lytinen. 1995. FAQ Finder: A case-based approach to knowledge navigation. In *Proceedings of the 11th Conference on Artificial Intelligence for Appricitions.*

Sanda Harabagiu, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Răzvan Bunescu, Roxana Gîrju, Vasile Rus, and Paul Morărescu. 2001. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the Association for Computational Linguistics.*

Sadao Kurohashi and Wataru Higasa. 2000. Dialogue helpsystem based on flexible matching of user query with natural language knowledge base. In *Proceedings of 1st ACL SIGdial Workshop on Discourse and Dialogue*, pages 141–149, HongKong.

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).

NIST and DARPA. 2001. *The Ninth Text REtrieval Conference (TREC-9).* NIST Special Publication.

Robert Wilensky, Yigal Arens, and David Chin. 1984. Talking to UNIX in English: An Overview of UC. *Communications of the ACM*, 27(6):574–593.