LETTER

Example-Based Query Generation for Spontaneous Speech

Hiroya MURAO[†,††a)], **Nobuo KAWAGUCHI**[†,†††], **Shigeki MATSUBARA**[†,†††], ***Members***,
and **Yasuyoshi INAGAKI**[††††], ***Fellow***

SUMMARY This paper proposes a new method of example-based query generation for spontaneous speech. Along with modeling the information flows of human dialogues, the authors have designed a system that allows users to retrieve information while driving a car. The system refers to the dialogue corpus to find an example that is similar to input speech, and it generates a query from the example. The experimental results for the prototype system show that 1) for transcribed text input, it provides the correct query in about 64% of cases and the partially collect query in about 88% 2) it has the ability to create correct queries for the utterances not including keywords, compared with the conventional keyword extraction method.
key words: *spoken dialogue, query generation, example-based*

1. Introduction

Many authors have proposed models for spoken dialogue processing by using state-transition, frame and so on. See [1] for example. It is difficult for such models to cover the various phenomena in spontaneously spoken dialogue. Recently, to overcome this difficulty, corpus-based dialogue models have been used for semantic analysis of spoken language or the optimization of dialogue strategies. Such models have been shown to be effective for the understanding of spontaneous speech [2]–[5].

This paper proposes a framework for constructing an information retrieval dialogue system using a dialogue corpus. Although many dialogue systems create search queries using keywords included in input utterances, it often happens that there is no keyword in the input because users' requests are not clear. In such cases, the dialogue system can't create correct search queries. Using this framework, however, the system can create search queries for such utterances referring the dialogue corpus. In this framework, the utterances stored in the dialogue corpus are used as examples, and the actions of the system are determined by those examples. Since the aim of the user of the information retrieval system is to create a query corresponding to the user's request, we can say that the process of creating a query is simply a mapping operation from the input utterance to the query. That is, we can expect that by using the pair of input utterance and output query as the example, a query corresponding to a user's input can be generated.

We collected data of spontaneously spoken dialogue in a moving car environment to implement and evaluate a robust spoken dialogue system, which allows users to retrieve shop information while driving a car [6]. Using this data, we constructed the example database and the dialogue system.

In the following sections, we examine the informational flow in an information retrieval dialogue in order to model the dialogue, then propose the query and reply generation method based on the dialogue examples. We also describe the design of the prototype system based on this method, and evaluate the system.

Manuscript received June 11, 2004.
Manuscript revised September 3, 2004.
[†]The authors are with the Center for Integrated Acoustic Information Research, Nagoya University, Nagoya-shi, 464–8601 Japan.
[††]The author is with the Digital Systems Development Center, Sanyo Electric Co. Ltd., Hirakata-shi, 573–8534 Japan.
[†††]The authors are with the Information Technology Center, Nagoya University, Nagoya-shi, 464–8601 Japan.
[††††]The author is with the Faculty of Information Science and Technology, Aichi Prefectural University, Aichi-ken, 480–1198 Japan.
a) E-mail: murao@hr.hm.rd.sanyo.co.jp

2. Example-Based Dialogue

2.1 Dialogue Model

Before considering a human-to-machine dialogue, let us attempt to model a human-user-to-human-operator dialogue. Figure 1 shows the informational flow in the information retrieval dialogue between a user and a human operator.

1. **Request**
   Receiving the user's request, the operator generates a database query according to the current dialogue context.
2. **Search**

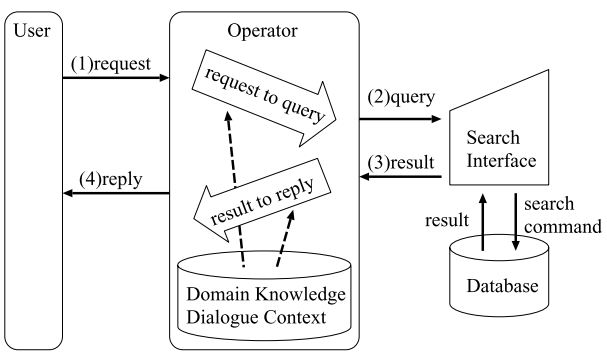


Fig. 1 Information flow of information retrieval dialogue.

The operator performs the search.
3. **Search results**
   A search result is generated.
4. **Response**
   The operator responds to the user according to the dialogue context and the search result.

As Fig. 1 shows, the operator makes the following two decisions:

1. Generating a search query on the basis of the user's utterance.
2. Responding to the user on the basis of the search result.

The skilled operator is considered to use domain knowledge, dialogue context, and past experience, and so forth, to make a "decision" regarding the appropriate response to a user's request.

However, it is difficult to make a comprehensive set of rules for such a "decision." Thus, we conclude that it is effective to make such a "decision" by referring to examples which a skilled human operator has performed.

2.2 Example-Based Query and Reply Generation

From Fig. 1 we understand that to design an example-based dialogue system, it is necessary to fix the process of query and reply generation and the form of examples. In our "example-based dialogue," we made them as follows:

- **Construction of the example database** The dialogues between the user and the operator are collected, with the operations performed at that time. The two actions for generating a query and a reply can be determined with the following two sets of information:

  Info A: For a decision regarding query generation

  1. User's utterance
  2. Context of dialogue

  Info B: For a decision regarding reply generation

  1. User's utterance
  2. Context of dialogue
  3. Search result

  Therefore, the example database should have five kinds of information: 1) user's utterances, 2) search queries, 3) operator's utterances, 4) results of the search, and 5) context information (past requests, past replies, past search results).
- **Query Generation Process** ("request to query" arrow in Fig. 1) For a user's request, the most similar example in the example database is selected for Info A. Then the query in the example is modified to render it suitable for the present situation. A search is then performed using this query.
- **Reply Generation Process** ("result to reply" arrow in Fig. 1) For the search result, the most similar example in the example database is selected for Info B. Then the reply statement in the example is modified to render it suitable for the present situation.

3. In-car Shop Information System

We have implemented a prototype system based on our idea proposed above. As the first step in the development, we targeted an operation for the context independent utterances.

3.1 System Configuration

The configuration of the system is shown in Fig. 2.

- **Dialogue Example Database (DEDB)**
  The dialogue example database has been constructed using the CIAIR-HCC (CIAIR spoken language dialogue corpus) [6]. For each utterance of a user's request, a search query corresponding to the utterance is recorded. A form of search queries is as follows:

  search ALL ITEM X KEY = $K_1, .., K_n, .., K_N$
  X : Sort key for the search result
  (NONE,POPULAR or NEAR)
  K_n : n-th keyword.

  For the sort key NONE, search results are sorted based on the number of matched keywords, as well as on the popularity value in SIDB (mentioned below) for POPULAR, and on the distance between the current position and each shop for NEAR. For each utterance of the operator's reply, the ID numbers of the search results are recorded. The text is analyzed morphologically. Important words (shop name, food name and so on) are classified semantically and assigned word class tags in advance. Figure 3 shows a sample of the DEDB.
- **Word Class Database (WCDB)**
  This database consists of the important words classified semantically. We classified them experientially on the basis of a dialogue corpus. The number of classes comes to 43, presently.
- **Shop Information Database (SIDB)**

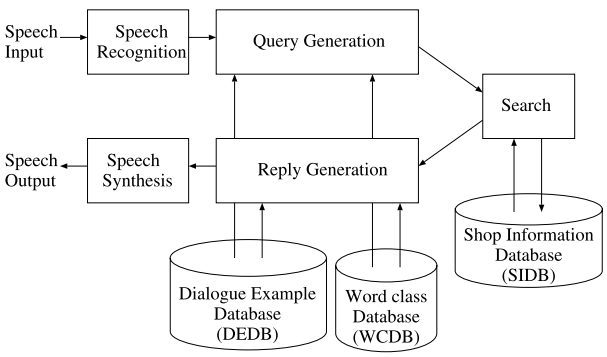


Fig. 2 System configuration.

U1: Wasyoku (Japanese Foods) ga tabetai na
(I'd like to eat Japanese Foods.)
Q1: search ALL ITEM NONE KEY=wasyoku
S1: Hai, wasyoku no omise wa 3-ken arimasu
(Well, I found 3 Japanese restaurants.)
A1: RESULT=3,ID1=020,ID2=098,ID3=865
#
U2: Ramen(noodles) wo tabe ni iki taina
(I'd like to eat noodles.)
Q2: search ALL ITEM NONE KEY=ramen
S2: Ramen no omise wa chikaku niwa arimasen
(There are no noodle restaurants near here.)
A2: RESULT=NONE

Fig. 3 Dialogue example database (part).

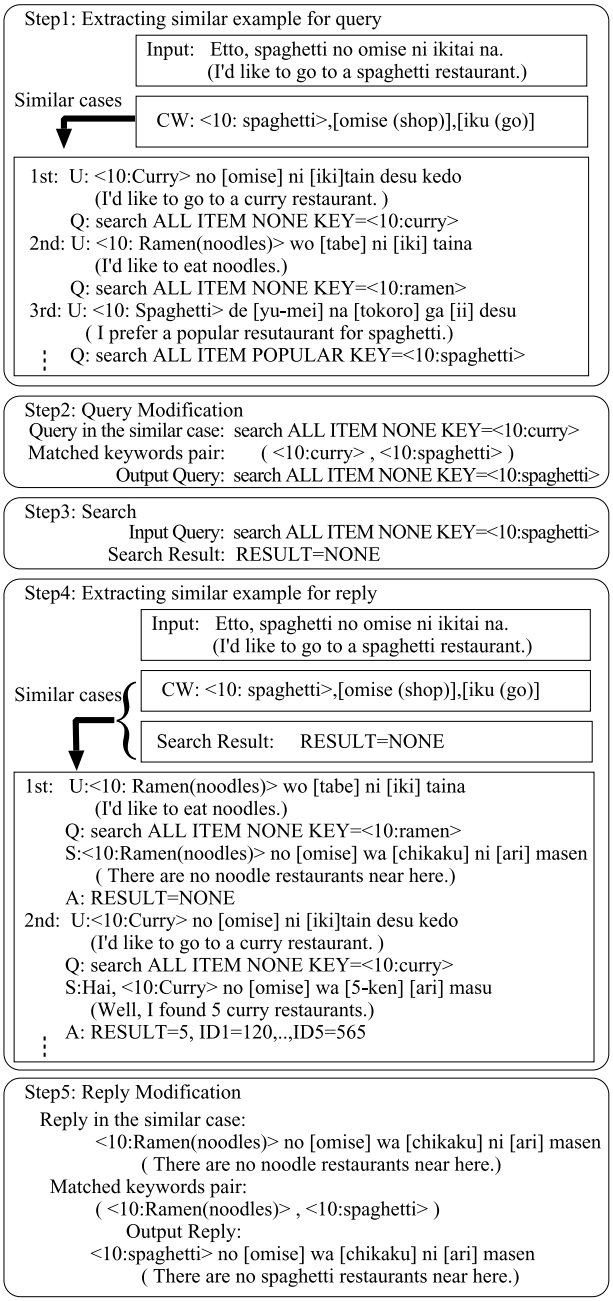
The restaurants in Nagoya are registered. The database is composed of about 800 shops.

- **Speech Recognition**
  The Japanese dictation toolkit [7] is used for Japanese speech recognition. The N-gram language model is created from the transcription of the dialogue speech.
- **Query Generation**
  The module extracts the example which is the most similar to the input utterance from the DEDB. Then the query in the example is modified to render it suitable for the present situation.
- **Search**
  The search module accesses the SIDB and generates the search result.
- **Reply Generation**
  The module extracts the example which is the most similar to the search result and the input utterance from the DEDB. Then the reply statement in the example is modified to render it suitable for the present situation.
- **Speech Synthesizer**
  The module synthesizes the sound of the reply statement.

3.2 The Procedure of Query and Reply Generation

We describe the behavior of the system during the process shown as an example in Fig. 4.

Step 1: Extracting similar example for query
For a speech recognition result, the system extracts the most similar example from the DEDB. The robustness of the similarity calculation between the input utterance and the utterance in the DEDB should be considered against the speech recognition error. Therefore, words which characterize the meaning of the utterance (CW or characterizing words) are used for the similarity calculation. For a speech recognition result combined with a morphological analysis result, independent words and the important words to which the word class tags are assigned according to the information in the WCDB are regarded as CW, and their similarity is calculated as follows. For each of user's utterances in the DEDB, the number of matched independent words and the number of important words which belong to the same word class are accumulated with the correspondent weight and the result is treated as the similarity. The utterance which marks the highest similarity is regarded as the most similar one.



<N: > important word (belongs to the Nth word class), [ ] independent word

Fig. 4 Example of query and reply generation.

Step 2: Query Modification
The query for the extracted example is modified with reference to the input utterance. The modification is performed by replacing the keywords in the query with words in the input utterance if they belong to the same word class.

Step 3: Search
The SIDB is searched by using the modified query and a search result is obtained.

Step 4: Extracting similar example for reply
The system extracts the most similar example from the DEDB, by taking account of not only the similarity between the input utterance and the utterance in examples but also that between the number of items in the search result and that in the examples. For example, if there are no items in the search result, it matches only the examples which have no items in the search result.

Step 5: Reply Modification
The reply statement for the extracted example is modified with reference to the input utterance. The modification is performed by replacing the words in the reference reply statement by using word class information. Then a speech synthesizer produces a reply speech.

4. Evaluation

We have evaluated the query generation part of the method by using context independent utterances. First, to reveal the fundamental performance of the query generation part, an experiment on the transcribed user's utterance is performed. After that, we clarify the relationship between the error rate of speech recognition and the query generation performance.

4.1 An Experiment on Transcribed Text Input

Table 1 shows the experimental conditions. The evaluation is performed based on the following procedure, with changing the number of utterances in the DEDB.

1. Input the transcription of the test data into the query generation part, and generate a query.
2. Classify the obtained query into one of four classes subjectively. (See Table 2.)

Figure 5 shows the experimental result. In the case with 537 examples, the correct queries (Class 1 in Table 2) were generated for about 64 % of the test data and about 88 % for Class 1 + 2. Moreover, we can also see that the rate of the correct answers is improved in accordance with the number of examples.

Compared with a conventional method using keywords included in the input utterances (**KEY**), the proposed method (**PRO**) can create search queries for the utterances that don't have keywords. As below, there existed such cases in the experiment.

<**CASE 1**>
Input (*in Japanese*)**:**
Nodo kawaita na. (The throat wants moisture.)
Generated Query (KEY):
search ALL ITEM NEAR KEY = NULL
Failed to keyword extraction.
Matched Example (PRO) (*in Japanese*)**:**
U: Nodo ga kawaita kara dokoka arimasu ka.
(The throat wants moisture, is there shop?)
Q: *search ALL ITEM NEAR KEY* = cafe
Generated Query (PRO):
search ALL ITEM NEAR KEY = cafe
Successfully created a query.

<**CASE 2**>
Input (*in Japanese*)**:**
Natsubate nano de karai mono ga tabetai na.
(I've lost my appetite because of the summer heat, so something hot, please.)
Generated Query (KEY):
search ALL ITEM NEAR KEY = NULL
Failed to keyword extraction.
Matched Example (PRO) (*in Japanese*)**:**
U: Natsubate de karai mon tabetai na.
(I've lost my appetite 'cause of the summer heat, something hot please.)
Q: *search ALL ITEM NEAR KEY* = Chinese
Generated Query (PRO):

Table 1 Experimental parameters.

Example data	537 utterances by 44 speakers (context independent)
Test data	89 utterances by 20 speakers (context independent)
Shop information database	785 items
Word class database	43 classes (2426 words)

Table 2 Classification for query evaluation.

Class 1	**Correct**
	There are enough elements (keywords, sort key).
Class 2	**Partially correct**
	It lacks some elements, but no wrong elements.
Class 3	**Wrong**
	There are wrong elements.
Class 4	**Query generation failure**
	Failed to create query. (No matched example.)

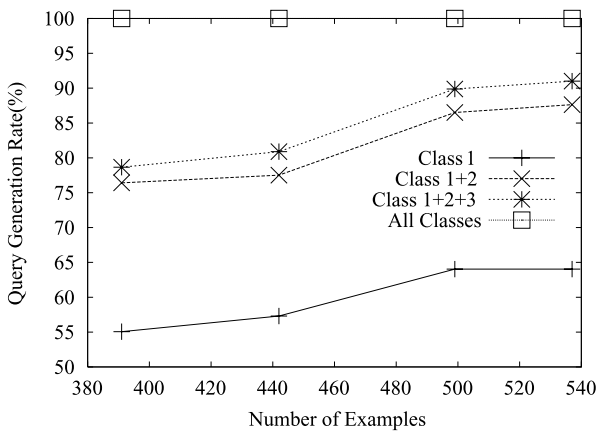


Fig. 5 Relationship between DEDB size and query generation rate (Transcribed text input).

search ALL ITEM NEAR KEY = Chinese
Successfully created a query.

In both cases, appropriate dialogue examples were extracted for input utterances that don't have keywords, and the queries for the extracted examples were used for output queries. As a result, correct search queries were created.

4.2 An Experiment on Speech Input

The system is required to have high performance in a driving car environment, so the robustness against error in speech recognition becomes important. In our method, when there are recognition errors in important words, it may be possible to create a correct query using the similar example that is extracted by the rest of independent words. An example is as follows:

Transcribed utterance:
[Assari] shita <3:wasyoku> ga [tabe] tai.
(I want to have light Japanese foods.)

Result of speech recognition (with *error*):
[Assari] shita [***watasi***] ga [tabe] tai.
(I want to have light ***me***.)

Matched example:
[Assari] shita no ga [tabe] tai no desu ga.

Obtained search query:
search ALL ITEM NEAR KEY = Japanese

<N:>: important word (belongs to the Nth word class),
[ ] : independent word

To examine the relationship between the error rate of speech recognition and the query generation performance, an experiment using speech input was performed. For simplicity, we used the reduced test data which contains only the utterances classified into Classes 1 and 2 in Table 2 in the testing of transcribed text input. The test data consists of 78 utterances. For these test data, the system can produce the correct query if the speech recognition performance is sufficient.

The main conditions of the speech recognition module are shown in Table 3. We used the "Japanese Dictation Toolkit 1999" [7] as a speech recognizer. For our test data, the word correct rate (WCR) is 62.17%, and the characterizing word correct rate (CWCR), which is the word correct rate for CW to be extracted for similarity calculation, is 61.31%.

The procedure for the evaluation is as follows: For each of the 78 test utterances, CWCR is calculated, and they are divided into five groups according to CWCR. The division rule is as follows (the numbers of utterances for each group are in parentheses):

Group 1:	0.0%	≤	**CWCR**	<	1.0%	(4)
Group 2:	1.0%	≤	**CWCR**	<	33.0%	(7)
Group 3:	33.0%	≤	**CWCR**	<	67.0%	(39)
Group 4:	67.0%	≤	**CWCR**	<	100.0%	(11)
Group 5:			**CWCR**	=	100.0%	(17)

Table 3 Main parameters of speech recognition module.

Acoustic model	PTM triphone HMM, 3000 states, 64 mixtures [7]
Language model (3-gram model)	CIAIR-HCC [6], 30,815 utterances by 106 speakers

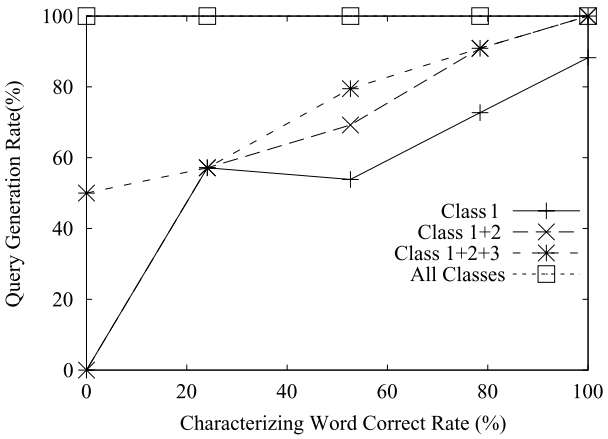


Fig. 6 Relationship between characterizing word correct rate (CWCR) and query generation rate.

Then, the query generation rate with 573 examples in the DEDB is calculated for each group. The total query generation rate for all 78 test utterances is 61.54% (Class 1) and 74.36% (Class 1 + 2).

The result is shown in Fig. 6. Each data is plotted on the x-axis with the value of the mean recognition rate of each five groups. From this data, we can see that, compared with the degradation of the CWCR, a higher query generation rate is maintained. This exemplifies the high robustness of our method with respect to errors of speech recognition.

5. Concluding Remarks

In this paper, we have proposed a method of generating a query by using practical human-to-human dialogues for information retrieval. The experimental results for the prototype system are as follows:

- For transcribed text input, it provides the correct query in about 64% of cases and the partially collect query in about 88%.
- For the input of speech recognition result, it achieves a relatively high query generation rate compared with the CW recognition rate.
- For the utterances not including keywords, it has the ability to create correct queries compared with the conventional keyword extraction method.

These results indicate that the method is effective.

Acknowledgement

This work is partially supported by a Grant-in-Aid for COE Research of the Ministry of Education, Culture, Sports, Science and Technology, Japan (No.11CE2005).

References

[1] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai, "A form-based dialogue manager for spoken language applications," Proc. ICSLP-96, pp.701–704, Oct. 1996.
[2] W. Minker, S. Bennacef, and J. L. Gauvain, "A stochastic case frame approach for natural language understanding," Proc. ICSLP-96, pp.1013–1016, Oct. 1996.
[3] M. Epstein, K. Papineni, S. Roukos, T. Ward, and S. Della Pietra, "Statistical natural language understanding using hidden clumpings," Proc. ICASSP-96, pp.176–179, May 1996.
[4] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialogue strategies," IEEE Trans. Speech Audio Process., vol.8, no.1, pp.11–23, Jan. 2000.
[5] S. Young, "Talking to machines (statistically speaking)," Proc. ICSLP-2002, pp.9–16, Sept. 2002.
[6] N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura, "Multi-dimensional data acquisition for integrated acoustic information research," Proc. LREC-2002, pp.2043–2046, May 2002.
[7] T. Kawahara, A. Lee, T. Kobayashi, K. Takeda, N. Minematsu, S. Sagayama, K. Itou, A. Ito, M. Yamamoto, A. Yamada, T. Utsuro, and K. Shikano, "Free software toolkit for Japanese large vocabulary continuous speech recognition," Proc. ICSLP-2000, vol.4, pp.476–479, Oct. 2000.