Dialogue Management for Language-Based Information Seeking

Kentaro Inui Akiko Yamashita Yuji Matsumoto

Nara Institute of Science and Technology
Takayama, Ikoma, Nara, 630-0192, Japan
tel. +81-743-72-5241 fax. +81-743-72-5249
{akiko-ya,inui,matsu}@is.aist-nara.ac.jp

Abstract

While the dialogue management for information seeking on table-based structured databases has been actively researched, little has been done on the management of language-based question-answering dialogues. This paper discusses how to incorporate some of the state-of-the-art techniques for information-seeking dialogue management into language-based question answering, proposing a basic framework where dialogue management is realized by coupling the inheritance of contextual information and the automatic relaxation of information requests. We present the current state of the implementation of our prototype system and clarify future issues.

keywords: dialogue management, question answering, information seeking

1 Introduction

The recent tremendous increase of electronically available documents has been stimulating research on question answering with the information source a document collection. It has been particularly driven by a series of evaluation-oriented workshops on question answering held at TREC[1] (QA Track) in the United States and NTCIR[2] (Question Answering Challenge, QAC) in Japan. We call this class of task *language-based question-answering* (LBQA).

LBQA offers significant potential for information access in that it does not require any structured database for the source of information but only assumes the availability of a document collection. However, research on LBQA has so far tended to focus on one-time question answering, where a question is answered independently of previous context. The next step of research should, therefore, be directed toward the incorporation of interactivity in question answering. Interactive LBQA would allow a user to ask context-dependent questions related to preceding questions and answers or decompose her/his information need into a sequence of smaller and fragmental questions.

The interactivity in information seeking has been pursued in a different context of research — natural language interfaces of table-based structured databases (e.g. [5, 3]). In this trend of research, the main issues has been:

- the context-dependent interpretation of user utterances that are often ambiguous and fragmental (elliptical) and sometimes are irrelevant or even wrong due to the user's misunderstanding or wrong belief,
- the generation of cooperative responses by inferring the purpose of user utterances, and
- the management of the mixed initiative of dialogue.

We call this class of task *table-based information seeking* (TBIS) to contrast it with LBQA. Unlike LBQA, TBIS assumes the source of information to be a structured table-based database.

Given this background, an interesting question is how to incorporate the state-of-the-art techniques for the management of TBIS dialogues into recently developed answer seeking techniques for LBIS. The coupling of these two research trends offers a new research field, i.e., LBIS dialogues. This paper reports on the present results of our attempts to make a computational model for the management of LBIS dialogues.

[1] http://trec.nist.gov/
[2] http://research.nii.ac.jp/ntcir/workshop/

2 Fundamental technologies

2.1 LBQA

A computational model of LBQA typically consists of the following three subprocesses:

1. *Question analysis*: Analyze a given question sentence to infer the *answer class* and extract a set of keywords which are to be used in Steps 2 and 3. For example, for a question "What city is the capital of Germany?", the answer class may be something like place-name and the keyword set may include "capital" and "Germany". For the set of answer classes, it has been common to use a named entity tag set as defined for evaluation-oriented workshops such as MUC[3] and IREX[4].

2. *Passage retrieval*: Use the extracted keywords to retrieve a limited number of passages that are likely to include the answer from the document collection.

3. *Answer seeking*: Use heuristics as below to seek the answer from the set of retrieved passages.

   a. Prefer a noun phrase that is consistent with the answer class.

   b. Prefer a noun phrase that bears the same syntactic/semantic relation with one of the keywords as the relation required by the question.

   c. Prefer a noun phrase that appears in proximity to the keywords.

Here, note that an information request (i.e., the content of a question) can approximately be represented as an answer class coupled with a set of keywords if we rely only on heuristics (a) and (c) ignoring syntactic/semantic relations. We call such a method is called a "bag-of-words"-based model.

While the research community for LBQA initially focused on the answer seeking for single context-independent questions, the QA track of TREC recently introduced a pseudo-interactive task called the CONTEXT subtask [2]. A similar subtask was also adopted at NTCIR-3 in 2002. However, in those problem settings, user-system interaction is over-simplified and thus is different from what we are aiming at because it is based on the following assumptions:

- The initial question of a given series of questions is a main question and the succeeding ones are all secondary and subordinate questions.

- The initial question is completely specified by the initial single user utterance.

- The user always takes the dialogue initiative and the system only answers the user's questions.

- Subsequent questions are asked whether the system has successfully answered the previous questions or not.

[3] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/
[4] http://www.cs.nyu.edu/cs/projects/proteus/irex/

2.2 Form-based dialogue management for TBIS

There have been various approaches to the management of TBIS dialogues, including the form-based approach [6, 3], the dialogue-model-based approach [7, 4] and the plan-based approach [1, 5]. For the basis of our approach to LBIS dialogues, we adopt the form-based approach for the following reasons:

- It can relatively domain-independent compared with other approaches because it requires very little of domain-dependent knowledge.

- A form used in this approach can be approximately represented as a keyword set, which fits the existing "bag-of-words"-based method for LBQA as noted above.

In the form-based dialogue management, a TBIS dialogue is viewed as a sequence of collaborative actions to fill the slots of a query form. A query form is a frame-based data representation that is designed to represent an information request. For example, the system developed by Chu-Carroll [3] represents an information request "What's playing in Montclair?" as:

Q-type = What
Movie = null
Theater = null
Town = Montclair

```
file:  en_ex2002
<title><date>平成 15 年度</date><event>入試</event>日程概要</title>
<org>情報科学研究科</org><prsn_num>146 名</prsn_num>※春期第 1 回/H14 秋期第 2 回
<date>H14.7.22(月)〜H14.7.25(木)</date>春期第 2 回/H14 秋期第 3 回<date>H14.9.17(火)
〜H14.9.18(水)</date>春期第 3 回/H15 秋期第 1 回<date>H15.3.5(水)</date><org>バイオサイエ
ンス研究科</org><prsn_num>114 名</prsn_num>...

file:  oc2002
<title><event>オープンキャンパス 2002<event></title>
<date>2002 年 6 月 8 日（土）</date><time>10:00〜16:00</time><text><org>奈良先端大</org>・
<org>情報科学研究科</org>を完全オープン！ 最先端の研究内容のみならず、<event>入試</event>
や生活環境についても説明します。...  (snipped) ... 全国各地から<prsn_num>784 人</prsn_num>
の方に参加していただき ました。当日は、各講座・研究設備の見学、教官や院生との懇談、研究科概要
と<event>入試の説明会<event>、研究内容の実演、パネル展示、<facilit>電子図書館</facilit>や
<facilit>学生宿舎</facilit>の公開など様々な催しが行われました。...
```

Figure 1: Document sample

The task of specifying a query form consists of the selection of an appropriate form template followed by the actions for slot filling. In the above *theater information* domain, for example, one may prepare such a form template as:

Q-type = What
Movie = forbidden
Theater = mandatory
Town = optional

The form-based dialogue management consists of the following three subtasks, in each of which form templates are effectively used.

- *Validation of information requests*: A template specifies the optionality of each slot (mandatory, optional or forbidden). An information request is invalid if a mandatory slot is unspecified or a forbidden slot is filled. If an invalid request is found, the system moves into a subdialogue such as a clarification subdialogue to fix it.

- *Completion of underspecified information requests*: If an inherit slot is unspecified, the system fills it with the item that appeared in the same slot of the previous query form.

- *Relaxation of information requests*: If no answer is found, the information request turns out to be too specific. In such a case, the system prompts the user to relax the value filling optional slots.

3 Management of LBIS dialogues

As a preliminary step toward the full-ranged realization of LBIS dialogues, we first want to realize the above-mentioned three subtasks of dialogue management in the context of LBIS dialogues.

3.1 Problem setting

Document collection We consider a problem setting where the information source is a closed and relatively small-sized collection of documents and each document is partially annotated with semantic tags. Figure 1 gives a document sample.

Domain-dependent knowledge The system is assumed to have domain-dependent knowledge about the semantic tag set:

- *Mapping relations between wordings and semantic tags*: For example, words "講座" (laboratory) and "研究室" (laboratory) both correspond to semantic tag `lab`.

- *Hierarchical relations between semantic tags*: For example, `lab` (laboratory) is a sub class of `org` (organization).

Answer seeking We assume an LBQA module that uses the "bag-of-words"-based method to seek answers (see Section 2.1). The method has been empirically proven to work reasonably well in a case where target documents are sufficiently annotated with semantic tags. This condition may

Input: *Who is the professor of Computational Linguistics Laboratory?*

QT: Who AC: person-name KL: {(Computational Linguistics Laboratory; lab; w_1), (professor; prof; w_2)}

Figure 2: Representation of an information request

seem unrealistic given the state-of-the-art technology for semantic tagging. However, in our problem setting, where we assume a closed document collection, the condition is likely to be satisfiable because semantic tags can be manually corrected.

3.2 Key ideas

Representation of information requests In LBIS dialogues, we cannot use slot-based query forms to represent information requests. We therefore represent an information request approximately as a keyword list (KL) coupled with a question type (QT) and answer class (AC) as exemplified in Figure 2. Each item of a keyword list is a triplet of keyword, semantic class and weight. The specification of the semantic class plays a role analogous to a slot in the form-based representation of an information request. A weight represents the degree of salience of the keyword in the current context, which is used in the constraint relaxation process we will describe later. The initial value of the weight of a keyword is determined in the question analysis process, and it is updated through the course of unfolding dialogue.

Completion of underspecified information requests In the form-based dialogue management, elliptical or fragmental utterances are interpreted with the help of inherit specifications of form templates. Namely, if an inherit slot is unspecified, it gets filled with the item that appeared in the same slot of the previous request. However, we cannot employ the same means because form templates are not available. Our approach to this problem is two folds. We first allow the system to inherit any of the previous keywords from the context. We then enable the system to revise the interpretation of a given request so that it can yield an appropriate answer.

Revision/relaxation of information requests Since form templates are not available, we cannot use the information of optionality specifications (i.e., mandatory, optional or forbidden) to check the validity of an information request. However, note that, if a given information request is invalid (e.g. if a forbidden slot is filled), the system will obtain either no answer due to overconstraining or too many answers due to underconstraining. Namely, the system can be aware of a flaw of a given information request, if any, by carrying out the answer seeking process for the request. Based on this consideration, we introduce the process of revising a given request. If a given request has something wrong, the system tries to revise it by deleting some of the existing keywords or inheriting new keywords from the preceding context until it finds a request that yields an appropriate number of answers.

3.3 Algorithm

Figure 3 gives an overview of the dialogue management process. The following is an example of the implementation of the above-mentioned key ideas.

Question analysis Analyze a given question (ⓐ) to extract an information request (ⓑ) through the following procedure (①):

1. Extract keywords from the question to create a weighted keyword list and identify the question type (QT) and answer class (AC).
2. If either QT or AC is missing, inherit it from the previous request.
3. Reduce the weights of the keywords registered in the current dialogue history and add them to the current keyword list.
4. Remove keywords with weights that are smaller than a threshold from the list.
5. Remove keywords whose semantic class is the same as AC from the list.
6. If there are more than one keyword of the same semantic class, remove those with less weights.

Answer seeking Carry out LBQA for the current information request and take one of the following actions depending on the result:

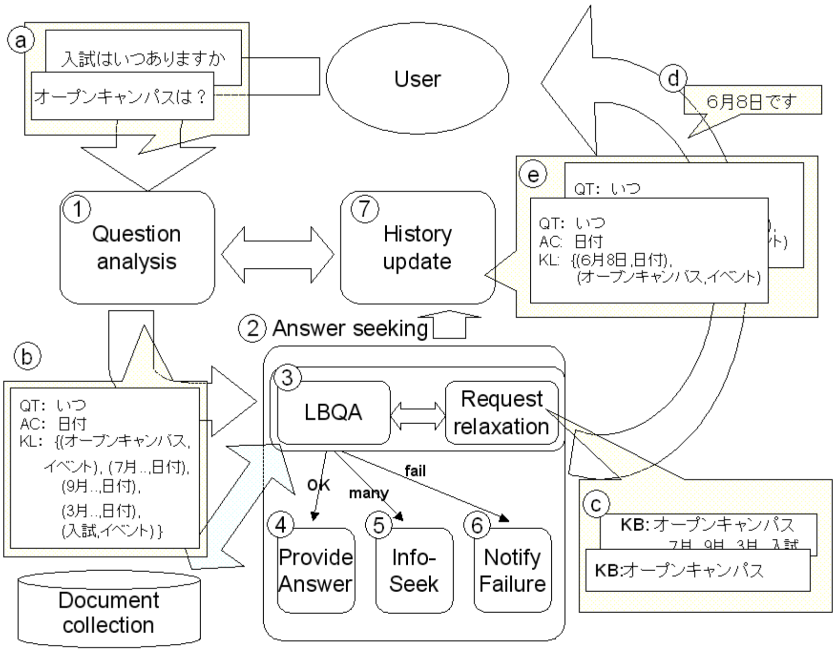

Figure 3: Flow of dialogue management

a. *Provide-Answer*: If an answer is found, provide it (④).

b. *Seek-Const*: If more than one answer are found, prompt the user to add more constraints (⑤).

c. *Relax-Request*: If no answer is found, relax the current request and carry out LBQA again (③).

d. *Notify-Failure*: If no answer is found after the request relaxation, notify the failure to the user (⑥).

Request relaxation Relax the constraints imposed by a given information request by deleting keywords from the current keyword list. Keywords with less weights have preference for deletion. The simplest way of realizing this would consist of the following steps:

1. Let P be the power set of the current keyword set.

2. Remove a keyword set k that maximizes the sum of weights from P and carry out LBQA for k.

3. Repeat Step 2 until an appropriate answer is found

Obviously, this procedure is too naive and computationally inefficient. However, it should be straightforward to refine it, for example, by replacing "the sum of weights" with a more sophisticated criterion which reflects the result of the previous run of passage retrieval.

History update Update the dialogue history each time when the system gives an utterance through the following operations.

a. Add the keywords that have been extracted from the user's current utterance to the history.

b. When the system provides an answer, add the answer phrase and keywords in the current keyword list to the history with those assigned the highest weights.

3.4 Examples

Here we give two examples. Keywords are assumed to be sorted according to their weights in the decreasing order. The keyword sets that are actually used for answer seeking are indicated in the keyword buffer slots (KB_i).

History inheritance

To interpret fragmental question U2, the system inherits the values of QT and AC from the context and finds the answer through two trials.

U1: 入試はいつありますか
When will the entrance exam be held?
QT/AC: いつ/date
KL: (入試, event)
KB_1: 入試
S1: 入試は H14.7.22(月)〜H14.7.25(木),H14.9.17(火)〜H14.9.18(水),H15.3.5(水) です
The entrance exams will be held on ...
KL: (H14.7.22(月)〜H14.7.25(木),date) (H14.9.17(火)〜H14.9.18(水),date), (H15.3.5(水),date), (入試,event)
U2: オープンキャンパス (OC) は？
What about the Open Campus?
QT/AC: null/null → いつ/date
KL: (OC, event), (H14.7.22(月)〜H14.7.25(木), date), (H14.9.17(火)〜H14.9.18(水), date), (H15.3.5(水),date), (入試,event)
KB_1: (OC), (H14.7.22 (月)〜H14.7.25(木)), (H14.9.17(火)〜H14.9.18(水)), (H15.3.5(水)), (入試)
KB_2: (OC)
S2: オープンキャンパスは 6 月 9 日です
The Open Campus will be held on ...
KL: (6 月 9 日,date), (OC,event)

Request relaxation

In reply to U3, the system finds the answer through repeated trials KB_1 to KB_5.

U1: 松本先生の部屋はどこですか
Where is Prof. Matsumoto's room?
QT/AC: どこ/room
KL:(松本,person), (先生, title)
S1: 松本先生の部屋は A701 B301 A511 です.
Prof. Matsumoto's rooms are A701, B301, A511.
KL:(A701,room), (B301,room), (A511,room), (松本,person), (先生, title)
U2: 言語の研究をしている方です
He is doing research on language.
QT/AC: null/null → どこ/room
KL: (言語, research_topic), (研究, null), (A701,room), (B301,room), (A511,room), (松本,person), (先生,title)
KB_1: (言語), (研究), (松本), (先生)
S2: 松本裕治先生の部屋は A701 です
Prof. Yuji Matsumoto's rooms is A701.
KL: (A701,room), (松本裕治,person), (言語,research_topic), (先生,title), (研究, null)
U3: 事務室はどこですか
Where is the secretariat office?
QT/AC: どこ/room
KB_1: (事務室), (A701), (松本裕治), (言語), (先生), (研究)
KB_2: (事務室), (A701), (松本裕治), (言語), (先生)
...(snipped)...
KB_5: 事務室
S3: 事務室は右手後方の部屋です
The secretariat office is a room to the right behind.

---

4 Implementation

We developed a prototype system. We designed it so that it will serve as a subcomponent of ASKA, a receptionist guidance robot developed at NAIST, which was designed to give information about staffs, labs and facilities at and around NAIST in reply to users' questions. At present, our prototype system accepts only keyboard inputs while ASKA is designed to participate in spoken dialogue.

In question analysis, the system employs *ChaSen* [10] for morphological analysis and *CaboCha* [9] for *Bunsetsu*-phrase chunking. It uses a manually encoded pattern set to extract the question type, answer class and keywords from a question. In answer seeking, it employs Namazu[5] for passage retrieval and uses proximity-based heuristics taking the answer class into account.

We created 30 sample scenarios, each of which was an independent dialogue consisting of 5.2 utterances on average, to conduct a preliminary evaluation of the system. For the target documents, we collected HTML documents from the Web site of NAIST, manually transformed them into 83 documents in total, and then partially annotate them with semantic tags. The user utterances in the scenarios were manually created by referring to (a) dialogue logs obtained by running the current version of ASKA against public users and (b) a collection of frequently asked questions collected by students who were not involved in our research. Since the syntactic and semantic analysis of user utterances

[5] http://www.namazu.org

is out of the present research focus, we heavily restricted the syntactic patterns of utterances so that they could analyzed by simple pattern matching.

We input the user utterances of the scenarios to the system and observed how often it could exhibit the same behavior as specified in the scenarios. The results are the following. First, the LBQA module worked well; it could find correct answers for roughly 90% of the questions as far as the information request was appropriately created. Second, the interpretation of elliptical/fragmental utterances was also reasonably successful. There were 17 dialogues that involved elliptical/fragmental utterances, and all the cases were correctly interpreted. Third, there were 6 dialogues where the information request was underspecified and the system was expected to prompt the user to provide further constraints. They were beyond the capability of the current system.

Note that the scenarios we used cannot be regarded as open data. Clearly, we need further evaluation in a more practical setting.

5 Discussion

This paper addressed the issue of incorporating existing dialogue management methods for TBIS into IBIS dialogue. There are still many open issues.

First, our present framework strongly relies on the assumption that the LBQA module can seek the answer accurately if a given information request is appropriate. This assumption is not so unrealistic as it may seem if the document collection is moderately limited in scale and sufficiently annotated with semantic tags. However, it should be verified against more practical settings.

Second, the representation we adopted for information requests also needs to be further examined. In the present algorithm, if there are more than one keyword of the same semantic class in the keyword list, we remove those with less weights. This operation is intended to roughly simulate the operation of updating slots of query forms. However, the robustness of it is still unclear.

Third, it is also important to address the issue of generating system utterances, which has been addressed in neither TREC nor NTCIR yet. Recent individual studies such as [8] are suggestive for this issue.

6 Conclusion

This paper discussed how to incorporate some of the state-of-the-art dialogue management techniques into language-based question answering. We proposed a preliminary framework where dialogue management was realized through coupling the inheritance of contextual information and the automatic relaxation of information requests. We reported on the current state of the implementation of our prototype system and discussed future directions.

References

[1] S. Carberry. Plan recognition in natural language dialogue. In *MIT Press*, 1990.

[2] S. Harabagiu, D. Moldovan, M. Pasca, M. Surdeanu, R. Mihalcea, R. Girju, V. Rus, F. Lacatusu, P. Morarescu, and R. Bunescu. Answering complex, list and context questions with LCC's Question-Answering Server. In *Proceedings of the Text Retrieval Conference (TREC-10)*, 2001.

[3] J. Chu-Carroll. Form-based reasoning for mixed-initiative dialogue management in information-query system. In *Eurospeech 99*, pp. 1519–1522, 1999.

[4] M. Nagata, T. Morimoto. An experimental statistical dialogue model to predict the speech act type of the next utterance. In *International Symposium on Spoken Dialogue, Tokyo, Japan*, 1993.

[5] J. D. Moore. Participating in explanatory dialogues. In *MIT Press*, 1995.

[6] R. Pieraccini, E. Levin, and W. Eckert. AMICA: the AT&T mixed initiative conversational architecture. In *Eurospeech 97*, pp. 1875–1878, 1997.

[7] T. Kawahara, M. Araki, and S. Doshita. Reducing syntactic perplexity of user utterances with automaton dialogue model. In *Proceedings of International Symposium on Spoken Dialogue – New Directions in Human and Man-Machine Communication –*, pp. 65–68, 1993.

[8] Y. Kiyota, S. Kurohashi and F. Kido. Dialog Navigator" A Question Answering System based on Large Text Knowledge Base. In *Proceedings of 19th COLING*, pp.460-466, 2002.

[9] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chuncking Journal of Information Processing Sciety of Japan, Vol. 43, No. 6, pp. 1834–1842, 2002.

[10] Y. Matsumoto et al. Japanese morphological analyzer *ChaSen*: Users Manual version 2.0. In *NAIST Technical Report NAIST-IS-TR99012*, 1999.