Some issues in dialogue-based question-answering

Arne Jönsson & Magnus Merkel

NLPLAB, Department of Computer and Information Science
Linköping University,
S-581 83 Linköping, SWEDEN
{arnjo,magme}@ida.liu.se

Abstract

We present on-going work on how to combine a multimodal dialogue system with techniques from information extraction and question-answering systems. A first combined system including both dialogue features and information extraction (BirdQuest) is presented. We conclude by listing a number of issues for further research.

Combining dialogue and document processing

In the field of Question Answering, Information extraction (IE) techniques have been used successfully when it comes to handling simple factoid questions, but the Q&A approach has yet not reached the level of sophistication for handling connected dialogue as is present in dialogue systems tailored to background systems with structured data. Dialogue capabilities allow for more precise formulation of information requests and more natural interaction. The challenge is to combine the IE techniques and some of the features of Q&A approaches with dialogue systems (Burger *et al.* 2001). By successfully combining these techniques, the goal would be to allow users to access information derived from a large set of, initially unstructured, documents, using dialogue functionalities, such as a dialogue history and clarification requests.

We have developed a first version of such a combined system, BirdQuest, which is a dialogue system to a large amount of textual data. The source data is initially provided as unstructured text but refined with IE techniques to be used with a dialogue system framework.

Development of BirdQuest

BirdQuest was developed for a web site where people, watching nature programs on TV, can ask questions related to the TV program, in this case questions on Nordic birds. A corpus of 329 information requests was collected from users asking questions on a web page hosted by the Swedish National Television.

BirdQuest has been developed iteratively, as presented below, where running prototype systems have been incrementally refined with more capabilities in well defined steps (Degerstedt & Jönsson, 2001).

Simple Q&A

The information on birds is based on a bird encyclopaedia that was marked up as XML entities using simple patterns for tagging named entities, such as bird names, colours, measurements, etc. A lexicon for the domain was constructed with the aid of a POS tagger and lemmatizer. A simple ontology was developed with representation of the type of objects, their properties and the relations that hold among them in the domain. Thus, the ontology provides a common vocabulary that can be used to state facts and formulate questions about the domain.

The system was tested by using requests taken from the collected question corpus. This first iteration showed, not surprisingly, that the simple approach taken here would indeed cover some basic factoid questions, but that more was needed in order to handle complex questions where more domain knowledge and inferencing are required.

Dialogue History

Adding dialogue capabilities, such as clarifications and connected dialogue, facilitates formulations of complex and connected requests. The dialogue component utilised in BirdQuest is the MALIN dialogue manager (Flycht-Eriksson & Jönsson 2000), which handles various dialogue features such as dialogue history and clarification requests. MALIN is modularised and especially it separates dialogue management (DM) from domain knowledge management (DKM). The former handles the dialogue whereas the latter handles access to various background systems. Thus, to be able to access the XML documents, new functions for background system access had to be implemented for the DKM.

In MALIN dialogue history is represented in dialogue objects with a parameter termed Objects representing the focused entities and another parameter termed Properties representing related concept information. To cater for dialogue history most features from MALIN could be used with only minor modifications. Objects in BirdQuest are normally birds and Properties model the information about the birds, such as appearance, number of eggs and feed, as seen in Dialogue 1 and 2 below.

```
U1: How long is the Eurasian Bullfinch?
S2: BirdQuest found the following information
   about the height of the Eurasian Bullfinch:
   ”17 cm”
U3: How many eggs do they lay?
S4: BirdQuest found the following information
   about the number of eggs for the Eurasian
   Bullfinch:
   ”The brood consists of 5-6 eggs”
U5: What do they eat?
S6: BirdQuest found the following information
   about feed for the Eurasian Bullfinch:
   ”The Eurasian Bullfinch eats mostly
   vegetarian feed with different nutrition
   choices during different seasons”
```

Dialogue 1. Example of dialogue history in BirdQuest[1]

User utterance U1 introduces the value “Eurasian Bullfinch” to the Objects parameter. The value of the Properties parameter is mapped in the lexicon to “height” and in S2 an answer is presented. In a similar fashion U3 and U5 are interpreted with the Objects parameter provided from the dialogue history.

MALIN also allows for clarification sub dialogues. Initially this is mostly used to help the user to specify requests, as shown in Dialogue 2.

```
U1: Information about Crows
S2: BirdQuest can give information about the
   following properties for Crows:
   call, habitat, appearance, feed, breeding
   Please specify what you’re interested in.
U3: I am interested in Call
S4: Call is vague, BirdQuest can give
   information about the following aspects of
   call:
   general, warning, song
   Please specify what your interested in.
U5: General, please.
S6: BirdQuest found the following information
   about general call for the Crow:
   ”The crows croak, krah, krah, is well known”
```

Dialogue 2. Example of clarification sub dialogue in BirdQuest

The utterance in U1 is too vague, but it is not suitable to present all properties in the domain to the user, instead the request is iteratively refined in the following interaction. In this example the system has the initiative and presents concepts from the domain ontology. However, often user concepts and system concepts differ which has to be taken care of, as will be discussed below, but first we need to improve the capacities for inference of information.

[1] Dialogues 1 and 2 are constructed from corpus questions.

Database conversion and SQL

After having a running BirdQuest dialogue system, we began fine-tuning the system by adding further capabilities. It turned out that much work was devoted to writing inference rules for finding information in the XML-tagged documents. We therefore considered transforming the documents to a relational database and utilising the built-in efficient inference abilities of SQL to improve the system.

We transformed parts of the XML documents into a relational database. The selection of what information to extract was guided by the collected question corpus and a wide variety of pattern extractor rules were used to identify the relevant information as slots and fillers. The objective was to only fill the database with relevant information and ignore text segments that did not meet the needs of the users as illustrated in the collected information requests. The slot and filler-type information in the database is illustrated in the right-hand column in Figure 1 below.

Original text	**Extracted information (DB)**
Black-throated diver *Gavia arctica* 58-73 cm, wingspan 110-130 cm. Somewhat larger than the red-throated diver with wider neck and straight, dagger-shaped beak…	NAME: Black-throated diver LATIN_NAME: Gavia arctica MAX_WING:130 MIN_WING: 110 MAX_HEIGHT: 73 MIN_HEIGHT: 58 BEAK_SHAPE: dagger-shaped, straight

Figure 1. Original text passage from the text book and the corresponding entry in the database (translated from Swedish).

This allowed BirdQuest to handle more complex requests concerning relations and comparisons, such as *Which is the largest bird?*, or *Are there any bird laying more eggs than this one?* The advantage here is that BirdQuest now can make sophisticated information searches in the database without the need for new inference rules. All that is needed is a straight-forward mapping from contextually interpreted questions into SQL queries. Such a module is already present in the DKM in MALIN.

The BirdQuest database relies heavily on the performance of the text extraction component. The more advanced this component is, the more features from the text documents can be accessed. In the current version, BirdQuest only has access to the information extracted to the database, i.e., there is no fallback strategy, such as performing free text search in the source text if the DB query does not return an answer. However, this is an option that is worth consideration even if it means loss of precision.

Currently, apart from the standard attribute-value pairs, like the ones shown in Figure 1, a number of the text segments are stored in the database as text and are presented to the user as such. Initial investigations revealed that users do refer to items in the text segments. However, as the system cannot access interpreted linguistic and ontological information inside these text items at present, a simple strategy of searching the associated text segments for each bird entry could be used as a last resort.

Integrating ontologies

Users often have different perspectives on the domain in question compared to the concepts expressed and used in reference material written by domain experts, such as a bird encyclopaedia. For instance, in the case of bird anatomy, an expert would have specialized terms to refer to different kinds of feathers whereas the novice would use more every-day descriptions. Thus, we need to identify different ontologies, in our case we identify a *user ontology* as developed from the corpus and a *system/domain ontology* based on the textual documents. The former is utilised by the DM and the latter is needed by the DKM to formulate SQL requests. However, we also need means for combining them.

In the current implementation we resolve conflicting concepts in various ways. One example is the use of the word "large", as seen in the first user utterance, *Which is the largest bird*, in Figure 2. There is not a single corresponding concept for the word "large" in the domain ontology, instead a user ontology concept, *size*, is introduced which in turn is mapped to the two domain ontology concepts: *wingspan* and *height*. This mapping is domain dependent and consequently not done in the lexicon but in the ontology, as being large does not imply, for instance, having large wingspan in all domains. Another example is "*small birds*" *(*Sw. *småfåglar)* which is not in the domain ontology, only in the user ontology. This concept cannot be mapped to other concepts as above; instead it is added as a new complementing concept.

Conclusions, current work and research issues

In this paper, we have very briefly presented how techniques for document analysis can be combined with a generic dialogue system. Through successive iterations we arrive at a system very much like current dialogue systems to databases.

This work revealed a number of further research issues.

- **More advanced ontology**. A more fine-grained ontology with a more complex representation would make it possible to make more inferences and detect information with more details. This is crucial both for the information extraction phase and the dialogue manager. Combining general and core ontologies efficiently with domain ontologies is also important for rapid application development.

U: Which is the largest bird?
S: BirdQuest found the following information…
… re: wingspan (cm), greatest found value: Whooper Swan
… re: height (cm), greatest value: Osprey

U: How many eggs do they lay?
S: BirdQuest found the following information…
… re: Whooper Swan: 4-7 eggs
…re: Osprey 2-3 eggs.

U: Blackbird?
S: BirdQuest found the following information…
… re: Blackbird…

Figure 2. BirdQuest session. To the left the interface in Swedish is shown. To the right a translation of the dialogue into English is presented.

- **Improved extraction methods.** Moving between domains and tasks require customization, in most cases more than you want. Better tools to make the process of building extraction patterns more efficient must be developed. Several researchers have pointed out this within the IE framework (cf. Yangarber & Grishman 1997), but in advanced Q&A and dialogue systems, the problem is to identify the user tasks in relation to the document base. Distilling important user tasks from a question corpus is complicated. And yet, the users and the texts are the ones that set the limits and possibilities of the system.
- **Improved use of question corpora**. More knowledge on the relationship between question types and answer types can guide the information extraction as well as aiding the dialogue management, cf. Zukerman & Horvitz (2001). The problem of collecting user questions that will guide the design of dialogue systems must however be dealt with. Factoid questions are relatively easy to collect from real usage and reapply in Q&A systems. Information systems with dialogue capabilities require empirical data containing connected dialogue.
- **Instantiation of complex expressions through interaction.** Documents may contain complex descriptions, such as formulas that can be used to calculate specific requests. The pattern extractor must then be able not only to identify the formula as an object, but also to break it down to factors and variables that can be instantiated in the interaction in order to provide coherent answers. Consider a dialogue system to documents on legal work conditions. If a user wants to find out the exact compensation figure for working overtime, the user would not want to be presented with the a snippet of text that spells out how to calculate this amount. Instead the dialogue system should initiate a dialogue that collects the concrete information needed to fill the variables of the formula from the user, apply these figures in the formula and present the result for the questioner.

- **Which user tasks and domains benefit from dialogue?** It is not certain that every domain need connected dialogue to the same extent as others. In complex applications such as the tax domain or legal documents concerning salary settlements, many different factors contribute to the solution to a specific user problem and thereby require clarifications, additional information, resolution of ambiguities, etc. Other application areas such as encyclopaedic lookup may need less dialogue features.
- **Towards connected dialogue in open domain.** BirdQuest was developed for a closed domain and one future research issue concerns the move to multi-domain and then to open domain applications. It is still very much an open question to what extent the techniques for IE, shared knowledge sources and dialogue management presented in this paper can be applied for such applications.

Acknowledgement

Vinnova, Swedish Agency for Innovation Systems finances this research. We are indebted to Frida Andén and Sara Norberg who implemented BirdQuest. Annika Flycht-Ericsson developed the ontology.

References

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., Jacquemin, C., Lin, C., Maiorano, S., Miller, G., Moldovan, D., Ogden, B., Prager, J., Rilo_, E., Singhal, A., Shrihari, R., Strzalkowski, T., Voorhees, E., & Weishedel, R. (2001). Issues, tasks and program structures to roadmap research in question & answering (Q&A). http://wwwnlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper v2.doc.

Degerstedt, L. & Jönsson, A. (2001). A method for systematic implementation of dialogue management. In Workshop notes from the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Seattle, WA.

Flycht-Eriksson, A & Jönsson, A., (2000) Dialogue and Domain Knowledge Management in Dialogue Systems Proceedings of the *1st SIGdial Workshop on Discourse and Dialogue*, Hong Kong

Yangarber, R. & Grishman, R. (1997). Customization of information extraction systems. In *Proceedings of International Workshop on Lexically Driven Information Extraction;* Frascati, Italy.

Zukerman, I. & Horvitz, E. (2001). Using machine learning techniques to interpret wh-questions. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France.