Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering

Manuel Kirschner
KRDB Center, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
kirschner@inf.unibz.it

Abstract. In Interactive Question Answering (IQA), dialogue context is needed to resolve context-dependent discourse phenomena, which occur relatively frequently in these dialogues. A focus tree is one viable model for representing dialogue context. We present a new IQA system which is based on this kind of model, and give a detailed overview of how this system has been built. The resulting system is used both as a practical IQA system that will help users retrieve the information they need, and as a test-bed for studying dialogue context models with real users.

1 Introduction

Lately, there has been increasing interest in how to best enrich QA applications with dialogue capabilities[1]. Interactive Question Answering (IQA) allows users to get concise answers to their information needs via cooperative natural-language dialogue. While classical QA is concerned with questions posed in isolation, its interactive variant keeps track of the QA process and supports the user in finding the exact solution via natural-language dialogue. In order to do so, it needs to model the dialogue context in which utterances are issued. Context has to be considered for appropriately handling clarification subdialogues, to resolve anaphora, ellipses or fragmentary utterances, and finally, to merge all the information provided over a series of turns, so that an answer to the complex question can be determined. We believe that effective use of context modeling in IQA lags behind. One of the main goals behind our research is to study different models of dialogue context (the focus tree introduced in this paper being one of the possible models). For this research, the general plan is to adopt an empirical approach: implementing

[1] E.g., Workshop on Interactive Question Answering (IQA'06), at HLT-NAACL'06

Proceedings of the Twelfth ESSLLI Student Session
Ville Nurmi and Dmitry Sustretov (editors)

different models in a practical IQA system, and then validating them with real user data.

Dialogue context is needed to resolve context-dependent discourse phenomena that occur in dialogues. These phenomena typically include pronouns and anaphoric noun phrases, elided phrases (missing semantic arguments), and fragments. In the course of a series of user utterances within an IQA session, each of these phenomena establishes some kind of dependency between the single utterances. We conducted a Wizard-of-Oz study with librarians of the university library and actual library users (cf. Kirschner (2006) for the experimental setup). One of the goals of this study was to analyze discourse phenomena occurring in actual user log files. Of the initially collected dialogues, around one quarter exhibited some kind of discourse phenomenon. Albeit on the lower end, this ratio is still within the spectrum reported in the literature. Conversation log files of information-seeking tasks in particular have been reported to contain the lowest share of context-dependent turns (Dahlbäck and Jönsson (1989), in Bertomeu et al. (2006)). Interpreting our experimental results, we believe that studying models of dialogue context is worthwhile both from a theoretical point of view, as well as with respect to the practical IQA system that will be described later in this paper.

This paper is structured as follows. In section 2, we introduce the notion of dialogue context, and explain the concept of focus trees. We also provide an overview of some relevant literature. Section 3 explains the background and general design principles of the IQA system that we are developing. Finally, in section 4, we present the implementation in detail.

2 Modeling dialogue context: Previous work

In order to correctly interpret every user utterance from a series, an IQA system needs a model of dialogue context that incorporates context-dependent discourse phenomena. More generally, for every new user utterance, the dialogue context model should correctly predict whether the topic of the interaction has stayed the same or switched to something new (possibly related in a specific way to the previous topic). De Boni and Manandhar (2005) and Yang et al. (2006) describe two approaches to recognizing whether the topic has changed between two subsequent user utterances. In both cases, the decision is based on a set of linguistic features extracted from the utterances; the features are then combined in decision algorithms using heuristics or supervised machine learning, respectively.

While these approaches are effective in terms of detecting topic changes, they do not attempt to model the patterns of topic change. As for these patterns, Bertomeu et al. (2006) provide an empirical study of thematic relations holding between user questions and the preceding context, and of the location of antecedents between user utterances in IQA dialogues. Different architectures and dialogue theories have been proposed for modeling dialogue context, and to explain certain patterns of topic change. What follows is a review of how focus trees have been used in this respect.

2.1 Modeling changes of dialogue topic using a focus tree

A focus tree provides a way of modeling the dialogue context that can account for topic changes occurring in the course of a dialogue. The main idea is to organize all the topics of the IQA system's task domain hierarchically. The nodes of this tree represent the current conversational topic (i.e., a concept that has already been mentioned in the on-going dialogue and that is currently in the focus of the dialogue participants). Representing the current topic via a specific node in the tree is based on the following notion: topic shifts to a somehow "related" topic are more likely than jumping to unrelated ones. In a focus tree, relatedness can be modeled via structural relations between nodes of the tree.

Several different ways of designing such trees have been mentioned in the literature (generally without giving a formulation of some rigorous algorithm). The applications for which focus trees have been employed are varied. McCoy and Cheng (1991) use the tree to constrain what should be said next in a natural-language generation system, by representing the cognitive load of different topic shifts in the tree. Jokinen et al. (1998) start with a manually built tree for marking up main topics in task-based dialogue data, which is extended by an n-gram-based model for topic shifts. The application: predicting the next topic in a spoken dialogue system. Finally, Stede and Schlangen (2004) propose to use a focus tree (in the form of a LOOM taxonomy) for dialogue management; given a user's dialogue act, the system retrieves a reply from the taxonomy based (at least partly) on structural aspects of the tree.

As in the three approaches just mentioned, the focus tree we are using in our IQA implementation (introduced in section 4) is also built entirely by hand. Thus, it relies critically on the exact way the tree was constructed. While this seems to be a more general problem with knowledge-intensive NLP systems, we hope to alleviate it in the future by defining some formal requirements for the construction of focus trees. Another way of avoiding the uncertainty of building focus trees by hand is to try to learn them from data. To point to one data-driven alternative: Niwa et al. (1997) learn certain relations between topics from free text. However, in our case, the lack of large amounts of training data prevents us from using such data-driven approaches.

2.2 Extended system interactivity by exploiting the dialogue context model

Besides the question of how to best model the dialogue context, another interesting issue for research has been the role of dialogue management in IQA. Here, we are concerned with the general dialogue strategy that the system should adhere to in a conversation (see Core et al. (2003) for a comparison of dialogue strategies in the context of tutorial dialogue). More specifically, one should identify the most helpful system responses at any point in the IQA dialogue. For example, Varges et al. (2006) describe a system that can modify the constraints of a user query by engaging in clarification subdialogues. A further goal of extended system interactivity could be to let the system actively guide the user through the information seeking process.

It is an open question whether an IQA system can provide certain types of extended interactivity by exploiting its model of dialogue context. The underlying notion is to use the dialogue context as a source for supporting (meta) knowledge that can be communicated to the user via system initiated turns. Thus, the system would not only answer user questions from within the task domain using the structured knowledge source, but also implicit knowledge extracted directly from the current state of the context model. The idea is that the user might benefit from viewing some version of contextual information that is normally not visible to him. An interesting starting point in this direction is provided by Chai and Jin (2004). As an IQA dialogue evolves and grows longer, they build up rich contextual information in the form of a directed acyclic graph that encodes the discourse roles and discourse relations introduced so far; they conjecture that these graphs could be used also as a basis for collaborative QA.

3 Proposed approach

For our study of dialogue context in IQA, we have been adopting a bottom-up approach: we start by implementing a baseline system that, while still being rather simplistic regarding the underlying theories, works robustly for a large proportion of the use cases. Talking about a practical IQA system, we start with a shallow Natural Language Understanding component (namely regular expression pattern matching) to map from user queries to system responses. Our initial model for dialogue context is a focus tree, which provides a simple way of keeping the dialogue state between two user turns. The type of interaction in our baseline IQA system is limited to a user-initiated stimulus-response loop, i.e., there is no system initiative yet, but the system simply returns one fixed response for every user utterance it receives. As soon as this system will be running, we propose to start collecting real data (from user interactions). Under the bottom-up paradigm, we expect to gain insights from the analysis of the data, especially regarding communication problems that will occur. This analysis will guide us as to which aspect of our current system should be fixed or improved first. We have started implementing these principles for designing a practical IQA system as a case study, which we now introduce.

3.1 Case study

Our university library is striving to improve their on-line information services. An IQA system provides permanent and instant access to library-specific information. As the experiences of other libraries have shown, such systems can surpass static information resources like FAQ lists in that they guide users towards a solution when initially they did not know the exact question.

Together with a team of librarians, we have started building a practical IQA system (BoB: the Bolzano library Bot). This project serves as a case study for implementing theories of dialogue context in IQA, and for validating them with real user data. As our research project (and our implementation of BoB) evolves, the library will have at their disposal an increasingly powerful IQA system. One of the long-term goals of the project is to support information seeking dialogues in three languages (English, Italian and German). See Kirschner (2007) for an overview of dedicated software tools that the librarians and domain experts use for the administration and translation of BoB's knowledge base from German into the other two target languages.

4 Implementation of a practical IQA system

We will now describe some results in terms of the current implementation of BoB. We start by showing how we built the focus tree from hand-coded data that we imported from another system. We then elaborate on the current implementation of BoB, describing in detail how it uses the library domain focus tree to yield a baseline IQA system.

4.1 A focus tree for the library domain

Through a cooperation with the library of the University of Hamburg, we acquired the library domain knowledge base[2] of Stella, a "chatterbot" (simple text-based dialogue system) implementation based on proprietary code. We planned to use these data for two purposes: to jump-start the creation of a focus tree for our own university library domain, and to extract and re-use as much as possible of the information that has been encoded by a team of librarians in Hamburg over several years.

[2]Note that the term "knowledge base" is used informally here; it refers to a hierarchy of library topics whose structure is not formalized.

With respect to jump-starting the creation of our own focus tree, we considered the Hamburg data to be interesting because of the following properties. Firstly, the application domain is very similar to ours: both Hamburg's chatterbot and our proposed IQA system provide a wide array of support to the users of a university library. Also, their knowledge base has been created, extended and fine-tuned by a team of around five librarians over several years. As a consequence, we hope that the quality and quantity of library topics encoded in the knowledge base let us build our first baseline IQA system with a good coverage of library-related questions and answers. Although we doubt that the Hamburg knowledge base will serve us directly for reaching new insights about dialogue context modeling, we do expect that from a data-driven perspective to building dialogue systems, the more data we have, the better.

The Hamburg library knowledge base encodes 230 topics in a focus tree[3]. As stated above, our secondary goal for incorporating the Hamburg data was to extract and take advantage of as much hand-coded information as possible that had been entered by Hamburg's librarians. Looking inside the 230 topics, the knowledge base consists of an overall of over 2000 pairs of 1. a regular expression pattern to match some user input, and 2. a canned-text system response to be returned to the user.

We do not know the exact principles with which Hamburg's focus tree was constructed, but after looking at their knowledge base in some detail, we conjecture that their librarians mixed different principles of organizing topics into a hierarchy[4]. A detailed analysis of the data showed that, besides containing the previously mentioned regular expression patterns and system responses organized into the focus tree, they contain a host of additional information, some of which we found useful to import into our own system's knowledge base. In the next section, we describe these additional features, as well as our current implementation of BoB, explaining in detail our focus tree search algorithm.

4.2 The BoB system

We have implemented BoB as a Java-based web application that will eventually be deployed on the library web site. Using the focus tree as a model for dialogue context, the system can in principle process user utterances that contain certain discourse phenomena (i.e., the above mentioned fragments, ellipses and anaphora). What follows is a description of the underlying no-

[3]At the highest level of the tree, there are 22 main topics, including: library buildings, organization, services, catalog query, books, journals, topics, articles, lending, inter-library loan, web site.

[4]E.g., based on semantically related concepts in the library domain, or related tasks that library users often perform, or local proximity of different concepts in a library building.

tion behind how BoB uses a focus tree to represent the dialogue context, and to generate a system response to a user query. Like most chatterbots in the tradition of ELIZA (Weizenbaum (1966)), our system is based on a stimulus-response loop for mapping a user utterance to some corresponding answer. All responses are stored as canned-text strings. Responding to some user query is thus a problem of identifying the best response, which is then simply output to the user. The mapping from user input to system response is done on the basis of regular expression patterns; for every system response, we have stored a regular expression pattern that matches certain types of user input.

In BoB, each regular expression pattern for matching user input is stored in combination with a pre-canned system response. Unlike in most chatterbots, these pairs are organized hierarchically as nodes of a focus tree, where each node represents a specific dialogue context. In the course of a dialogue, the current topic switches between the nodes of the tree, depending on what regular expression patterns the current user utterance matches, and at which node the search for a matching regular expression pattern starts. In this simple model of dialogue history, the current focus node represents the dialogue state, i.e., it encodes all the information that is preserved between two succeeding user utterances.

As mentioned above, the knowledge base we acquired from Hamburg contains a host of information that goes beyond the topic hierarchy and the regular expression patterns and canned-text answers encoded in each focus node. Some parts of this additional information seemed too idiosyncratic to re-use for the BoB system. For example, some nodes in the Hamburg focus tree contain hand-tuned weights for changing the precedence in which they are processed by the search algorithm. Since we do not know how these weights were chosen, nor how exactly the original search algorithm uses them, we did not consider them in our system. On the other hand, we do re-use two extra features encoded in Hamburg's focus tree, namely context-dependent follow-up questions and system-initiated subdialogues.

Re-using context-dependent follow-up questions

One of the distinguishing features of *Interactive* QA is that it allows users to pose questions that are related in certain ways to the previous dialogue. We call every question in an IQA dialogue “follow-up” if there exists at least some previous user question or system response, since all follow-ups are potentially related to the dialogue context. An analysis of the kinds of (thematic) relations that may hold in these situations is outside the scope of this paper, but we believe it to be important to further the understanding of IQA dialogues in general, and the requirements for practical IQA systems (manuscript in preparation). At this point, we are interested only in the subset of *context-dependent* follow-up questions, i.e., that require some additional information from the dialogue context in order to be fully specified and unambigious. In fact, the Hamburg focus tree includes dedicated focus nodes that specifically cover context-dependent follow-up questions. These focus nodes were assigned a special "context-dependent" attribute by Hamburg's domain experts, which is interpreted by the focus tree search algorithm in that it searches these nodes first, and only in the specific dialogue context for which the domain experts have foreseen the follow-up question. What follows is an example taken from the Hamburg focus tree (and re-used for BoB). After the user has asked about the availability of guided tours to the library, he asks an elliptical follow-up question[5]: "Where is the meeting point?". This follow-up matches with the regular expression pattern from a focus node marked specifically as context-dependent, whose question pattern contains only "where", but does not require "guided tour" to appear in the question.

Re-using subdialogues

Subdialogues are used to encode relatively short, predefined sequences of system questions, to which the user's answer must come from a small, predefined set of possible answers (e.g., "yes/no" for simple questions). From the domain expert's point of view, they allow users to be guided through the domain by pointing them to relevant options in specific dialogue situations. This kind of guidance should be especially useful for inexperienced users who do not know how to formulate their problem or question explicitly. Subdialogues must be handled explicitly by the search algorithm, since their regular expression patterns must only be searched when the corresponding subdialogue is active. This prevents focus nodes with unspecific patterns like "yes" to be selected in the course of the global focus tree search (steps 3 and 4 of the algorithm described at the end of this section).

The decision in favor of importing the pre-defined subdialogues comes at a cost. The way that subdialogues are included in the focus tree using a special type of focus nodes breaks the otherwise clean and purely declarative nature of the tree structure. Besides encoding topics and sub-topics, the tree now contains nodes with procedural semantics, which require the focus tree search algorithm to follow a hard-coded link to some other (possibly remote) node, where the next user turn can then be processed as a continuation of the subdialogue. One solution for separating the declarative topic hierarchy from these procedural additions would be to have a more powerful dialogue manager that could generate subdialogue sequences on the fly, given information about the current topic. It would be a possible step towards a better understanding and control of system initiative in an IQA system.

[5] In the surface form, the follow-up question is lacking the attribute "...for guided tours?".

Searching the focus tree

We now describe the current implementation of BoB's search algorithm, and how it works in conjunction with the focus tree introduced earlier in this section. Every time the user enters a new question, a suitable node in the focus tree has to be identified, so that the system response stored in that specific topic node can be returned as an answer. The search for a focus node depends both on the user input and on the previously active focus node. By starting each search for the next system response at the currently active focus node, we take advantage of idea that topics which describe likely continuations of the conversation should have been placed closer (in terms of node distance) to each other in the tree. Figure 1.1 shows a flow chart of the focus tree search algorithm as it is currently implemented in BoB. For compactness of the diagram, the following notation is used:

C	the Current focus node
SD	a SubDialogue focus node
CD	a Context-Dependent focus node
N	a "normal" focus node (i.e., neither SD nor CD)
Match	specific focus node retrieved by previous search operation
.link	link attribute of SD, pointing to specific focus node where subdialogue processing will resume
.sysResponse	system response encoded in focus node
SD-Mode	flag indicating if system is currently in a subdialogue
"matching"	focus node's regular expression pattern matches current user input

Conceptually, the search algorithm can be divided into two parts. In the first part (consisting of the pattern matching steps marked with (1) and (2) in the diagram), subdialogues and context-dependent follow-up questions are dealt with on a local level, i.e., *without* the current focus shifting to a node more remote than the sibling nodes. In the second part of the algorithm (pattern matching steps marked with (3) and (4)), the search space for a system response is gradually extended to the entire focus tree.

5 Conclusion and future directions

Currently, the contents and the topology of the focus tree are determined entirely by domain experts based on their intuition. We are currently exploring systematic ways of constructing or extending a focus tree, so as to get an understanding of which (follow-up) user questions will be covered by the system (manuscript in preparation). What we clearly lack at this point is an evaluation of the BoB system with respect to real user data. We are considering different possibilities for this. Regarding BoB's coverage

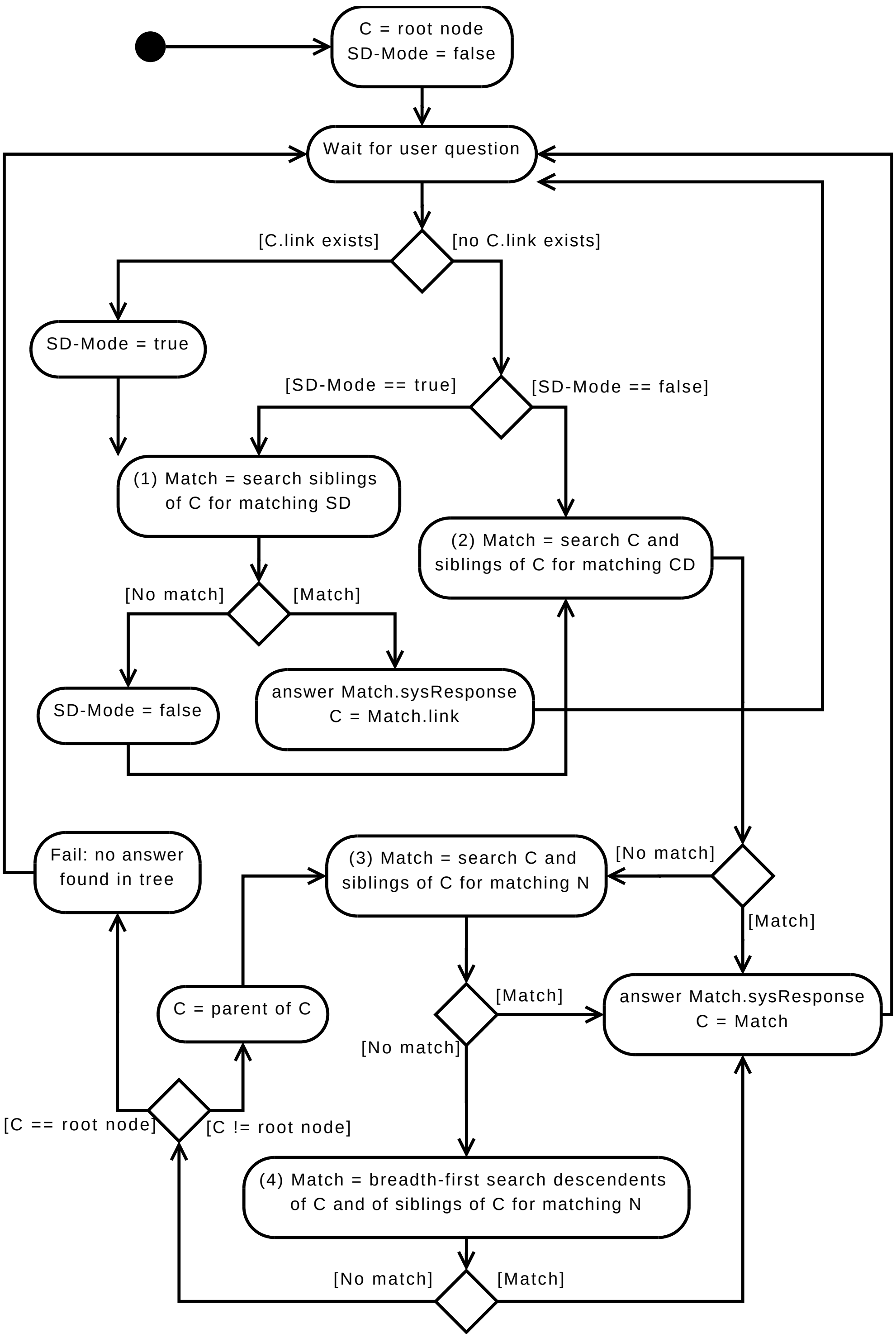

Figure 1.1: The BoB focus tree search algorithm

of the most important topics of user questions, we have only a preliminary result based on the not yet "localized" Hamburg focus tree (cf. Kirschner (2006)). A more thorough user study can be conducted once the focus tree has been completely adjusted by our domain experts (see below). This user study will also have to verify the ability of the system to handle context-dependent follow-up questions via the special "CD" focus nodes described earlier. We are currently studying how well our focus tree-based approach is able to model (context-dependent) follow-up user questions, using our previously collected corpus of Wizard-of-Oz dialogues, and how this depends on the topology of the focus tree (manuscript in preparation).

Although the current BoB system is simplistic in many ways (e.g., lacking linguistic knowledge of morphology or (lexical) semantics, or a model for the pragmatics of dialogue in IQA), the main advantage of our approach is evident: we were able to build a working IQA system from scratch in a relatively short time (around 1 year). At the time of writing, a team of domain experts is working on the localization of the focus tree in terms of the covered topics and the two additional target languages, using tools described in Kirschner (2007). Once this task will be finished, we will be able to compare different ways of modeling dialogue context in IQA (based on differently constructed focus trees), using the running BoB system as a test bed. Given dialogue log files, we plan to study patterns of topic change within IQA dialogues.

Acknowledgements

We would like to thank Bonnie Webber for the discussions about IQA; her talk on that topic, delivered at the Free University of Bozen-Bolzano in November 2006, inspired especially Section 2.

Bibliography

Bertomeu, N., H. Uszkoreit, A. Frank, H.-U. Krieger, and B. Jörg (2006). Contextual phenomena and thematic relations in database QA dialogues: results from a Wizard-of-Oz experiment. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, New York, NY, pp. 1–8.

Chai, J. Y. and R. Jin (2004). Discourse structure for context question answering. In *Proc. of the HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, Boston, MA.

Core, M. G., J. D. Moore, and C. Zinn (2003). The role of initiative in tutorial dialogue. In *Proc. of EACL'03*, Morristown, NJ, USA, pp. 67–74. Association for Computational Linguistics.

Dahlbäck, N. and A. Jönsson (1989). Empirical studies of discourse representations for natural language interfaces. In *Proc. of the 4th Conference of the European Chapter of the ACL (EACL'89)*, Manchester, UK, pp. 291–298.

De Boni, M. and S. Manandhar (2005). Implementing clarification dialogues in open domain question answering. *Journal of Natural Language Engineering 11*(4), 343–361.

Jokinen, K., H. Tanaka, and A. Yokoo (1998). Context management with topics for spoken dialogue systems. In *Proc. of the Joint International Conference of Computational Linguistics and the Association for Computational Linguistics (COLING-ACL'98)*, Montreal, Canada, pp. 631–637.

Kirschner, M. (2006). Building a multi-lingual interactive question-answering system for the library domain. In *Proc. of Brandial (SemDial'06)*, Potsdam, Germany.

Kirschner, M. (2007). The BoB IQA system: a domain expert's perspective. In *Proc. of Decalog (SemDial'07)*, Rovereto, Italy.

McCoy, K. F. and J. Cheng (1991). Focus of attention: Constraining what can be said next. In C. L. Paris, W. R. Swartout, and W. C. Mann (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 103–124. Norwell, MA: Kluwer Academic Publishers.

Proceedings of the Twelfth ESSLLI Student Session
Ville Nurmi and Dmitry Sustretov (editors)

Niwa, Y., S. Nishioka, M. Iwayama, A. Takano, and Y. Nitta (1997). Topic graph generation for query navigation: Use of frequency classes for topic extraction. In *Proc. of NLPRS'97, Natural Language Processing Pacific Rim Symposium '97*, Phuket, Thailand, pp. 95–100.

Stede, M. and D. Schlangen (2004). Information-seeking chat: Dialogue management by topic structure. In *Proc. of SemDial'04 (Catalog)*, Barcelona, Spain.

Varges, S., F. Weng, and H. Pon-Barry (2006). Interactive question answering and constraint relaxation in spoken dialogue systems. In *Proc. of the 7th SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the Assiciation for Computing Machinery 9*, 36–45.

Yang, F., J. Feng, and G. Di Fabbrizio (2006). A data driven approach to relevancy recognition for contextual question answering. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, New York City, NY, pp. 33–40.