The BoB IQA System: a Domain Expert's Perspective

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

Introduction

We present BoB, a multilingual Interactive Question Answering system we have been developing to be deployed on the web-site of our university library. While being rather simplistic regarding the underlying theories of language and dialogue, it is an adequate baseline system that could be developed in around one year's time, and is easy to tailor and maintain by library domain experts. In this paper, we describe the current development version of BoB from a domain expert's perspective, giving an overview of the ways in which they can enhance the system, and what tools they use to make these modifications. With the tools presented here, our domain experts can autonomously extend BoB's knowledge base with new question topics, dialogue features and additional languages. Currently, BoB is running (although not publicly accessible) and can be tested in the German version; also, the tools described here have been fully implemented and are regularly used.

1 BoB's Search Algorithm

Like typical "chatterbots", BoB uses a stimulus-response loop for mapping a user utterance to some corresponding "canned-text" response. Answering a user question thus becomes a problem of retrieving the best response. The mapping from user input to system response is done on the basis of regular expression patterns; for every system response, there is a pattern that is supposed to match a specific class of user input. In BoB, these patterns and the corresponding system answers are stored in pairs. Unlike in most chatterbots that have no representation of state, these pairs are stored in a focus tree that represents the dialogue context. In the course of a dialogue, the current topic switches between the focus nodes of the tree, depending on what regular expression patterns the current user utterance matches, and on the previously active node. In this simple model, the current focus node thus represents the dialogue state.

2 Jump-starting the Focus Tree

Through a cooperation with the library of the University of Hamburg, we acquired the knowledge base of Stella, a relatively sophisticated German chatterbot application for the library domain[1]. Our main goal for using these data was to extract a part of the encoded library application-specific information. We jump-started the creation of BoB's focus tree by extracting the regular expression patterns and corresponding system responses (both for German), as well as the topic hierarchy in which these pairs were organized. In this way, we got a topic hierarchy consisting of 230 topics[2], containing an overall of over 2000 pairs of regular expression patterns and system responses.

There are two problems with re-using the data from Stella in our project. First, many of the topics are unique to the University of Hamburg library and have to be removed from BoB's focus tree, while other topics are obviously missing; a similar problem concerns the regular expression patterns that often contain Hamburg-specific parts that have to be changed. The second problem is related to the seemingly ad hoc way that the Stella topic hierarchy is organized. In practice, this makes it potentially difficult for the domain experts to decide under which topic nodes to insert new pattern-response pairs. While in principle a topic hierarchy could be used as an interesting model for tracking dialogue focus, only few of the existing regular expression patterns in Stella actually happen to require a specific location within the hierarchy structure (these are patterns for underspecified user questions that can only be inter-

[1] http://www.sub.uni-hamburg.de/informationen/projekte/infoass.html

[2] Examples from the 22 main topics: library buildings, organization, services, catalog query, books, journals, topics, articles, lending, inter-library loan, web site.

preted by knowing the dialogue context: *context-dependent follow-up questions*). So far, our domain experts are using the Stella topic hierarchy to cluster and organize the pattern-response pairs; in the absence of concise criteria for how the hierarchy should be built, they are free to add, move and delete topics as they see fit. In practice, the situation in which BoB's search algorithm critically depends on the tree structure is when faced with a context-dependent follow-up question: in this case, the algorithm begins by searching focus nodes with a "follow-up" tag (see below) among the children of the last active node.

3 Controlling BoB's Dialogue Features

As mentioned above, one of BoB's dialogue-related features is its ability to handle context-dependent follow-up questions. This requires the domain expert to foresee possible ways in which users might follow up on a topic encoded in BoB's hierarchy, and add new focus nodes (with the "follow-up" tag) as children of the respective topic node. We are currently exploring principled ways in which to support the domain expert in this task (so that catering for follow-ups becomes less dependent on human intuition).

Another dialogue feature of BoB which is under the domain expert's control are sub-dialogues. They are used to implement system-initiated clarification requests, or more generally, to guide the user through the library domain via system initiative. In the first case, an ambiguous user question would trigger a clarification sub-dialogue, while in the second case, some system response might provide the user with a choice of related topics about which the user could then ask in more detail. Technically, both cases are implemented with sub-dialogues, by assigning special "link" elements to certain nodes in the tree. If in a dialogue BoB reaches a node with a "link" element, the normal tree search algorithm is suspended and resumes processing only after moving to the sub-dialogue node referenced in the link element.

4 Tools for the Domain Experts

The central tool used by the domain experts to edit BoB's knowledge base is a free, off-the-shelf XML editor[3] in conjunction with three style sheets for providing different views of the focus tree.

[3]XMLMind Standard Edition, `http://www.xmlmind.com/`

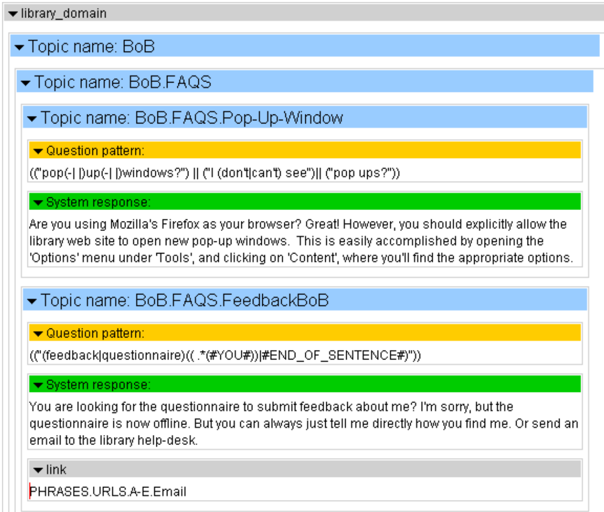

Figure 1: "Focus node" view of the BoB focus tree

This is the information that the respective views convey: (i) the BoB topic hierarchy, with fields for temporarily deactivating certain topic nodes; (ii) the focus nodes in detail, including regular expression pattern, system response, and possibly a "link" element (cf. Fig. 1); (iii) for each focus node, the (German) regular expression patterns and system responses along with fields for the Italian and English translations (used by the domain experts in charge of BoB's translation).

The "follow-up" tag is realized as an XML attribute that the domain expert can assign to any focus node, using the XML editor's standard methodology for changing attributes. Besides the customized XML editor, the domain experts also use a tool for checking BoB's extended regular expression syntax for correctness and matching against possible user questions. Additionally, they constantly test their latest changes to the focus tree on their own development version of BoB.

5 Future Work

We are exploring several ways of adding more intelligence to BoB; we are currently concentrating on follow-up questions, and how to accommodate them in the focus tree in a more systematic way. For the domain experts, this will mean less "guessing" possible user questions, since a theory should be able to automatically predict likely follow-ups. Also, once BoB goes public, we will collect user log files, which could further guide the domain experts in adjusting the focus tree. The data will also prove useful for validating our approach to IQA in general.