Personalized, Interactive Question Answering on the Web

Silvia Quarteroni
University of Trento
Via Sommarive 14
38100 Povo (TN), Italy
silviaq@disi.unitn.it

Abstract

Two of the current issues of Question Answering (QA) systems are the lack of personalization to the individual users' needs, and the lack of interactivity by which at the end of each Q/A session the context of interaction is lost.

We address these issues by designing and implementing a model of personalized, interactive QA based on a User Modelling component and on a conversational interface. Our evaluation with respect to a baseline QA system yields encouraging results in both personalization and interactivity.

1 Introduction

Information overload, i.e. the presence of an excessive amount of data from which to search for relevant information, is a common problem to Information Retrieval (IR) and its subdiscipline of Question Answering (QA), that aims at finding concise answers to questions in natural language. In Web-based QA in particular, this problem affects the relevance of results with respect to the users' needs, as queries can be ambiguous and even answers extracted from documents with relevant content but expressed in a difficult language may be ill-received by users.

While the need for user personalization has been addressed by the IR community for a long time (Belkin and Croft, 1992), very little effort has been carried out up to now in the QA community in this direction. Indeed, personalized Question Answering has been advocated in TREC-QA starting from 2003 (Voorhees, 2003); however, the issue was solved rather expeditiously by designing a scenario where an "average news reader" was imagined to submit the 2003 task's *definition* questions.

Moreover, a commonly observed behavior in users of IR systems is that they often issue queries not as standalone questions but in the context of a wider information need, for instance when researching a specific topic. Recently, a new research direction has been proposed, which involves the integration of QA systems with dialogue interfaces in order to encourage and accommodate the submission of multiple related questions and handle the user's requests for clarification in a less artificial setting (Maybury, 2002); however, Interactive QA (IQA) systems are still at an early stage or applied to closed domains (Small et al., 2003; Kato et al., 2006). Also, the "complex, interactive QA" TREC track (www.umiacs.umd.edu/~jimmylin/ciqa/) has been organized, but here the interactive aspect refers to the evaluators being enabled to interact with the systems rather than to dialogue *per se*.

In this paper, we first present an adaptation of User Modelling (Kobsa, 2001) to the design of personalized QA, and secondly we design and implement an interactive open-domain QA system, YourQA. Section 2 briefly introduces the baseline architecture of YourQA. In Section 3, we show how a model of the user's reading abilities and personal interests can be used to efficiently improve the quality of the information returned by a QA system. We provide an extensive evaluation methodology to assess such efficiency by improving on our previous work in this area (Quarteroni and Manandhar, 2007b).

Moreover, we discuss our design of interactive QA in Section 4 and conduct a more rigorous evaluation of the interactive version of YourQA by comparing it to the baseline version on a set of TREC-QA questions, obtaining encouraging results. Finally, a unified model of personalized, interactive QA is described in Section 5.

2 Baseline System Architecture

The baseline version of our system, YourQA, is able to extract answers to both factoid and non-factoid questions from the Web. As most QA systems (Kwok et al., 2001), it is organized according to three phases:

- **Question Processing**: The query is classified and the two top expected answer types are estimated; it is then submitted to the underlying search engine;
- **Document Retrieval**: The top n documents are retrieved from the search engine (Google, www.google.com) and split into sentences;

Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions, pages 33–40
Manchester, August 2008

- **Answer Extraction**:
    1. A sentence-level similarity metric combining lexical, syntactic and semantic criteria is applied to the query and to each retrieved document sentence to identify candidate answer sentences;
    2. Candidate answers are ordered by relevance to the query; the Google rank of the answer source document is used as a tie-breaking criterion.
    3. The list of top ranked answers is then returned to the user in an HTML page.

Note that our answers are in the form of sentences with relevant words or phrases highlighted (as visible in Figure 2) and surrounded by their original passage. This is for two reasons: we believe that providing a context to the exact answer is important and we have been mostly focusing on non-factoids, such as definitions, which it makes sense to provide in the form of a sentence. A thorough evaluation of YourQA is reported in e.g. (Moschitti et al., 2007); it shows an F1 of 48±.7 for non-factoids on Web data, further improved by a SVM-based re-ranker.

In the following sections, we describe how the baseline architecture is enhanced to accommodate personalization and interactivity.

3 User Modelling for Personalization

Our model of personalization is centered on a User Model which represents students searching for information on the Web according to three attributes:

1. age range $a \in \{7-10, 11-16, adult\}$,
2. reading level $r \in \{basic, medium, advanced\}$;
3. profile p, a set of textual documents, bookmarks and Web pages of interest.

Users' age[1] and browsing history are typical UM components in news recommender systems (Magnini and Strapparava, 2001); personalized search systems such as (Teevan et al., 2005) also construct UMs based on the user's documents and Web pages of interest.

3.1 Reading Level Estimation

We approach reading level estimation as a supervised learning task, where representative documents for each of the three UM reading levels are collected to be labelled training instances and used to classify previously unseen documents.

Our training instances consist of about 180 HTML documents from a collection of Web portals[2] where pages are explicitly annotated by the publishers according to the three reading levels above. As a learning model, we use unigram language modelling introduced in (Collins-Thompson and Callan, 2004) to model the reading level of subjects in primary and secondary school.

Given a set of documents, a unigram language model represents such a set as the vector of all the words appearing in the component documents associated with their corresponding probabilities of occurrence within the set.

In the test phase of the learning process, for each unclassified document D, a unigram language model is built (as done for the training documents). The estimated reading level of D is the language model lm_i maximizing the likelihood that D has been generated by lm_i (In our case, three language models lm_i are defined, where $i \in \{basic, medium, advanced\}$.). Such likelihood is estimated using the function:

$$L(lm_i|D) = \sum_{w \in D} C(w, D) \cdot log[P(w|lm_i)], \quad (1)$$

where w is a word in the document, $C(w, d)$ represents the number of occurrences of w in D and $P(w|lm_i)$ is the probability that w occurs in lm_i (approximated by its frequency).

3.2 Profile Estimation

Information extraction from the user's documents as a means of representation of the user's interests, such as his/her desktop files, is a well-established technique for personalized IR (Teevan et al., 2005).

Profile estimation in YourQA is based on key-phrase extraction, a technique previously employed in several natural language tasks (Frank et al., 1999).

For this purpose, we use Kea (Witten et al., 1999), which splits documents into phrases and chooses some of the phrases as be key-phrases based on two criteria: the first index of their occurrence in the source documents and their $TF \times IDF$ score[3] with respect to the current document collection. Kea outputs for each document in the set a ranked list where the candidate key-phrases are in decreasing order; after experimenting with several values, we chose to use the top 6 as key-phrases for each document.

The profile resulting from the extracted key-phrases is the base for all the subsequent QA activity: any question the user submits to the QA system is answered by taking such profile into account, as illustrated below.

3.3 Personalized QA Algorithm

The interaction between the UM component and the core QA component modifies the standard QA process at the Answer Extraction phase, which is modified as follows:

[1] Although the reading level can be modelled separately from the age range, for simplicity we here assume that these are paired in a reading level component.

[2] Such Web portals include: `bbc.co.uk/schools`, `www.think-energy.com`, `kids.msfc.nasa.gov`.

[3] The $TF \times IDF$ of a term t in document D within a collection S is: $TF \times IDF(t, D, S) = P(t \in D) \times -logP(t \in [S/D])$.

1. The retrieved documents' reading levels are estimated;
2. Documents having a different reading level from the user are discarded; if the remaining documents are insufficient, part of the incompatible documents having a close reading level are kept;
3. From the documents remaining from step 2, key-phrases are extracted using Kea;
4. The remaining documents are split into sentences;
5. Document topics are matched with the topics in the UM that represent the user's interests;
6. Candidate answers are extracted from the documents and ordered by relevance to the query;
7. As an additional answer relevance criterion, the degree of match between the candidate answer document topics and the user's topics of interest is used and a new ranking is computed on the initial list of candidate answers.

Step 7 deserves some deeper explanation. For each document composing the UM profile and the retrieved document set, a ranked list of key-phrases is available from the previous steps. Both key-phrase sets are represented by YourQA as arrays, where each row corresponds to one document and each column corresponds to the rank within such document of the key-phrase in the corresponding cell.

As an illustrative example, a basic user profile, created from two documents about Italian cuisine and the movie "Ginger and Fred", respectively, might result in the following array:

$$\begin{bmatrix} pizza & lasagne & tiramisu & recipe & chef & egg \\ fred & ginger & film & music & movie & review \end{bmatrix}$$

The arrays of UM profile and retrieved document key-phrases are named P and $Retr$, respectively. We call $Retr_i$ the document represented in the i-th row in $Retr$, and P_n the one represented in the n-th row of P [4]. Given k_{ij}, i.e. the j-th key-phrase extracted from $Retr_i$, and P_n, i.e. the n-th document in P, we call $w(k_{ij}, P_n)$ the *relevance* of k_{ij} with respect to P_n. We define

$$w(k_{ij}, P_n) = \begin{cases} \frac{|Retr_i| - j}{|Retr_i|}, & k_{ij} \in P_n \\ 0, & otherwise \end{cases} \quad (2)$$

where $|Retr_i|$ is the number of key-phrases of $Retr_i$. The total relevance of document $Retr_i$ with respect to P, $w_P(Retr_i)$, is defined as the maximal sum of the relevance of its key-phrases, obtained for all the rows in P:

$$w_P(Retr_i) = max_{n \in P} \sum_{k_{ij} \in Retr_i} w(k_{ij}, P_n). \quad (3)$$

The personalized answer ranking takes w_P into account as a *secondary* ranking criterion with respect to the baseline system's similarity score; as before, Google rank of the source document is used as further a tie-breaking criterion.

Notice that our approach to User Modelling can be seen as a form of implicit (or quasi-implicit) relevance feedback, i.e. feedback not explicitly obtained from the user but inferred from latent information in the user's documents. Indeed, we take inspiration from (Teevan et al., 2005)'s approach to personalized search, computing the relevance of unseen documents (such as those retrieved for a query) as a function of the presence and frequency of the same terms in a second set of documents on whose relevance the user has provided feedback.

Our approaches to personalization are evaluated in Section 3.4.

3.4 Evaluating Personalization

The evaluation of our personalized QA algorithms assessed the contributions of the reading level attribute and of the profile attribute of the User Model.

3.4.1 Reading Level Evaluation

Reading level estimation was evaluated by first assessing the robustness of the unigram language models by running 10-fold cross-validation on the set of documents used to create such models, and averaging the ratio of correctly classified documents with respect to the total number of documents for each fold. Our results gave a very high accuracy, i.e. 94.23% ± 1.98 standard deviation.

However, this does not prove a direct effect on the user's perception of such levels. For this purpose, we defined *Reading level agreement (A_r)* as the percentage of documents rated by the users as suitable to the reading level to which they were assigned. We performed a second experiment with 20 subjects aged between 16 and 52 and with a self-assessed good or medium English reading level. They evaluated the answers returned by the system to 24 questions into 3 groups (basic, medium and advanced reading levels), by assessing whether they agreed that the given answer was assigned to the correct reading level.

Our results show that altogether, evaluators found answers appropriate for the reading levels to which they were assigned. The agreement decreased from 94% for A_{adv} to 85% for A_{med} to 72% for A_{bas}; this was predictable as it is more constraining to conform to a lower reading level than to a higher one.

3.4.2 Profile Evaluation

The impact of the UM profile was tested by using as a baseline the standard version of YourQA, where the UM component is inactive. Ten adult participants from various backgrounds took part in the experiment; they were invited to form an individual profile by brainstorming key-phrases for 2-3 topics of

[4] Note that, while column index reflects a ranking based on the relevance of a key-phrase to its source document, row index only depends on the name of such document.

their interest chosen from the Yahoo! directory (`dir.yahoo.com`): examples were “ballet”, “RPGs” and “dog health”.

For each user, we created the following 3 questions so that he/she would submit them to the QA system: Q_{per}, related to the user’s profile, for answering which the *personalized* version of YourQA would be used; Q_{bas}, related to the user’s profile, for which the *baseline* version of the system would be used; and Q_{unr}, unrelated to the user’s profile, hence not affected by personalization. The reason why we handcrafted questions rather than letting users spontaneously interact with YourQA’s two versions is that we wanted the results of the two versions to be different in order to measure a preference. After examining the top 5 results to each question, users had to answer the following questionnaire[5]:

- For each of the five results *separately*:

  TEST1: *This result is useful to me*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

  TEST2: *This result is related to my profile*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

- For the five results taken as a whole:

  TEST3: *Finding the info I wanted in the result page took*:
  1) Too long, 2) Quite long, 3) Not too long, 4) Quite little, 5) Very little

  TEST4: *For this query, the system results were sensitive to my profile*:
  5) Yes, 4) Mostly yes, 3) Maybe, 2) Mostly not, 1) Not at all

The experiment results are summarized in Table 1. The first row reports a remarkable difference between the perceived usefulness for question Q_{rel} with respect to question Q_{bas} (answers to TEST1).

Table 1: Profile evaluation results (avg ± st. dev.)

Measurement	Q_{rel}	Q_{bas}	Q_{unr}
TEST1	3.6±0.4	2.3±0.3	3.3±0.3
TEST2	4.0±0.5	2.2±0.3	1.7±0.1
TEST3	3.1±1.1	2.7±1.3	3.4±1.4
TEST4	3.9±0.7	2.5±1.1	1.8±1.2

The results were compared by carrying out a one-way analysis of variance (ANOVA) and performing the Fischer test using the usefulness as factor (with the three queries as levels) at a 95% level of confidence. The test revealed an overall significant difference between factors, confirming that users are positively biased towards questions related to their own profile when it comes to perceived utility.

To analyze the answers to TEST2 (Table 1, row 2), which measured the perceived relatedness of each answer to the current profile, we used ANOVA again and and obtained an overall significant difference. Hence, answers obtained without using the users’ profile were perceived as significantly less related to those obtained using their own profile, i.e. there is a significant difference between Q_{rel} and Q_{bas}. As expected, the difference between Q_{rel} and Q_{unr} is even more significant.

Thirdly, the ANOVA table computed using average perceived time (TEST3) as variable and the three questions as factors did not give any significance, nor did any of the paired t-tests computed over each result pair. We concluded that apparently, the time spent browsing results is not directly correlated to the personalization of results.

Finally, the average sensitivity of the five answers altogether (TEST4) computed over the ten participants for each query shows an overall significant difference in perceived sensitivity between the answers to question Q_{rel} (3.9±0.7) and those to question Q_{bas} (2.5±1.1) and Q_{unr} (1.8±1.2).

To conclude, our experience with profile evaluation shows that personalized QA techniques yield answers that are indeed perceived as more satisfying to users in terms of usefulness and relatedness to their own profile.

4 Interactivity

Making a QA system interactive implies maintaining and efficiently using the current dialogue context and the ability to converse with the user in a natural manner. Our implementation of IQA is guided by the following conversation scenario:

1. An optional reciprocal greeting, followed by a question *q* from the user;
2. *q* is analyzed to detect whether it is related to previous questions or not;
3. (a) If *q* is unrelated to the preceding questions, it is submitted to the QA component;
   (b) If *q* is related to the preceding questions (follow-up question), it is interpreted by the system in the context of previous queries; a revised version of *q*, *q’*, is either directly submitted to the QA component or a request for confirmation (grounding) is issued to the user; if he/she does not agree, the system asks the user to reformulate the question until it can be interpreted by the QA component;
4. As soon as the QA component results are available, an answer *a* is provided;

[5] The adoption of a Likert scale made it possible to compute the average and standard deviations of the user comments with respect to each answer among the top five returned by the system. It was therefore possible to replace the binary measurement of perceived usefulness, relatedness and sensitivity used in (Quarteroni and Manandhar, 2007b) in terms of total number of users with a more fine-grained one in terms of average computed over the users.

5. The system enquires whether the user is interested in submitting new queries;

6. Whenever the user wants to terminate the interaction, a final greeting is exchanged.

4.1 Choosing a Dialogue Manager

Among traditional methods for implementing information-seeking dialogue management, Finite-State (FS) approaches are the simplest. Here, the dialogue manager is represented as a Finite-State machine, where each state models a separate phase of the conversation, and each dialogue move encodes a transition to a subsequent state (Sutton, 1998). However, an issue with FS models is that they allow very limited freedom in the range of user utterances: since each dialogue move must be pre-encoded in the models, there is a scalability issue when addressing open domain dialogue.

On the other hand, we believe that other dialogue approaches such as the Information State (IS) (Larsson et al., 2000) are primarily suited to applications requiring a planning component such as closed-domain dialogue systems and to a lesser extent to open-domain QA.

As an alternative approach, we studied conversational agents ("chatbots") based on AIML (Artificial Intelligence Markup Language), such as ALICE[6]. Chatbots are based on the pattern matching technique, which consists in matching the last user utterance against a range of dialogue patterns known to the system. A coherent answer is created by following a range of "template" responses associated with such patterns.

As its primary application is small-talk, chatbot dialogue appears more natural than in FS and IS systems. Moreover, since chatbots support a limited notion of context, they can handle follow-up recognition and other dialogue phenomena not easily covered using standard FS models.

4.2 A Wizard-of-Oz Experiment

To assess the utility of a chatbot-based dialogue manager in an open-domain QA application, we conducted an exploratory Wizard of Oz experiment.

Wizard-of-Oz (WOz) experiments are usually deployed for natural language systems to obtain initial data when a full-fledged prototype is not yet available (Dahlbaeck et al., 1993) and consist in "hiding" a human operator behind a computer interface to simulate a conversation with the user, who believes to be interacting with a fully automated prototype.

We designed six tasks reflecting the intended typical usage of the system (e.g.: "Find out who painted Guernica and ask the system for more information about the artist") to be carried out by 7 users by interacting with an instant messaging platform, which they were told to be the system interface.

The role of the Wizard was to simulate a limited range of utterances and conversational situations handled by a chatbot.

User feedback was collected mainly by using a post-hoc questionnaire inspired by the experiment in (Munteanu and Boldea, 2000), which consists of questions Q_1 to Q_6 in Table 2, col. 1, to be answered using a scale from 1="Not at all" to 5="Yes, absolutely".

From the WOz results, reported in Table 2, col. "WOz", users appear to be generally very satisfied with the system's performances: Q_6 obtained an average of 4.5±.5. None of the users had difficulties in reformulating their questions when this was requested: Q_4 obtained 3.8±.5. For the remaining questions, satisfaction levels were high: users generally thought that the system understood their information needs (Q_2 obtained 4) and were able to obtain such information (Q_1 obtained 4.3±.5).

The dialogue manager and interface of YourQA were implemented based on the dialogue scenario and the successful outcome of the WOz experiment.

4.3 Dialogue Management Algorithms

As chatbot dialogue follows a pattern-matching approach, it is not constrained by a notion of "state": when a user utterance is issued, the chatbot's strategy is to look for a pattern matching it and fire the corresponding template response. Our main focus of attention in terms of dialogue manager design was therefore directed to the dialogue tasks invoking external resources, such as handling follow-up questions, and tasks involving the QA component.

4.3.1 Handling follow-up questions

For the *detection* of follow-up questions, the algorithm in (De Boni and Manandhar, 2005) is used, which uses features such as the presence of pronouns and the absence of verbs in the current question and word repetitions with the n previous questions to determine whether q_i is a follow-up question with respect to the current context. If the question *q* is not identified as a follow-up question, it is submitted to the QA component. Otherwise, the *reference resolution* strategy below is applied on *q*, drawing on the stack S of previous user questions:

1. If *q* is *elliptic* (i.e. contains no verbs), its keywords are completed with the keywords extracted by the QA component from the previous question in S for which there exists an answer. The completed query is submitted to the QA component;

2. If *q* contains *pronoun/adjective anaphora*, a chunker is used to find the most recent compatible antecedent in S. This must be a NP compatible in number with the referent.

3. If *q* contains *NP anaphora*, the first NP in S containing all the words in the referent is used to replace the latter in *q*. When no antecedent can be

[6] www.alicebot.org/

found, a clarification request is issued by the system until a resolved query can be submitted to the QA component.

When the QA process is terminated, a message directing the user to the HTML answer frame (see Figure 1) is returned and a follow-up proposal or an enquiry about user satisfaction is optionally issued.

4.4 Implementation

To implement the dialogue manager and allow a seamless integration with our Java-based QA system, we extended the Java-based AIML interpreter Chatterbean[7]. We started by augmenting the default AIML tag set (including tags such as **<srai>** and **<that>**) with two tags: **<query>**, to seamlessly invoke the core QA module, and **<clarify>**, to support follow-up detection and resolution.

Moreover, the interpreter allows to instantiate and update a set of variables, represented as context properties. Among others, we defined:
a) **userID**, which is matched against a list of known user IDs to select a UM profile for answer extraction (see Section 5);
b) the current **query**, which is used to dynamically update the stack of recent user questions used by the clarification request detection module to perform reference resolution;
c) the **topic** of conversation, i.e. the keywords of the last question issued by the user which received an answer. The latter is used to clarify elliptic questions, by augmenting the current query keywords with those in the topic when ellipsis is detected.

Figure 1 illustrates YourQA's interactive version, which is accessible from the Web. As in a normal chat application, users write in a text field and the current session history as well as the interlocutor replies are visualized in a text area.

4.5 Interactive QA evaluation

For the evaluation of interactivity, we built on our previous results from a Wizard-of-Oz experiment and an initial evaluation conducted on a limited set of handcrafted questions (Quarteroni and Manandhar, 2007a). We chose 9 question series from the TREC-QA 2007 campaign[8]. Three questions were retained per series to make each evaluation balanced. For instance, the three following questions were used to form one task: 266.1: "When was Rafik Hariri born?", 266.2: "To what religion did he belong (including sect)?" and 266.4: "At what time in the day was he assassinated?".

Twelve users were invited to find answers to the questions to one of them by using the standard version of the system and to the second by using the interactive version. Each series was evaluated at least once using both versions of the system. At the end of the experiment, users had to give feedback about both versions of the system by filling in the satisfaction questionnaire reported in Table 2.

Table 2: Interactive QA evaluation results obtained for the WOz, Standard and Interactive versions of YourQA. Average ± st. dev. are reported.

	Question	WOz	Stand	Interact
Q_1	Did you get all the information you wanted using YourQA?	4.3±.5	4.1±1	4.3±.7
Q_2	Do you think YourQA understood what you asked?	4.0	3.4±1.3	3.8±1.1
Q_3	How easy was it to obtain the information you wanted?	4.0±.8	3.9±1.1	3.7±1
Q_4	Was it difficult to reformulate your questions when requested?	3.8±.5	-	3.9±.6
Q_5	Do you think you would use YourQA again?	4.1±.6	3.3±1.6	3.1±1.4
Q_6	Overall, are you satisfied with YourQA?	4.5±.5	3.7±1.2	3.8±1.2
Q_7	Was the pace of interaction with YourQA appropriate?	-	3.2±1.2	3.3±1.2
Q_8	How often was YourQA sluggish in replying to you?	-	2.7±1.1	2.5±1.2
Q_9	Which interface did you prefer?	-	41.7%	58.3%

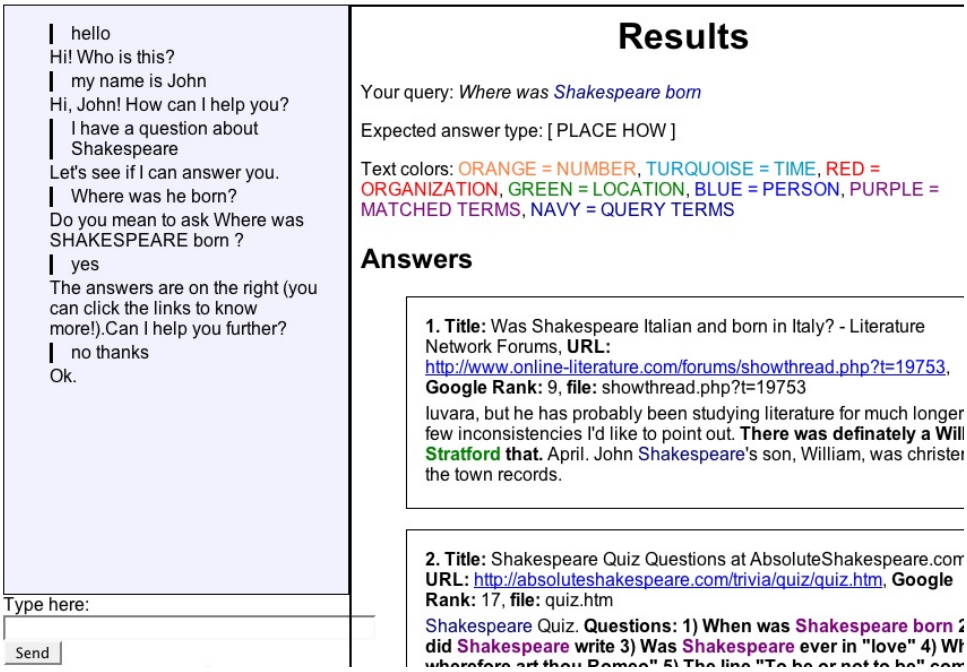
Although the paired t-test conducted to compare questionnaire replies to the standard and interactive versions did not register statistical significance, we believe that the evidence we collected suggests a few interesting interpretations.

First, a good overall satisfaction appears with both versions of the system (Q_6), with a slight difference in favor of the interactive version. The two versions of the system seem to offer different advantages: while the ease of use of the standard version was rated higher (Q_3), probably because the system's reformulation requests added a challenge to users used to search engine interaction, users felt they obtained more information using the interactive version (Q_1).

Concerning interaction comfort, users seemed to feel that the interactive version understood better their requests than the standard one (Q_2); they also found it easy to reformulate questions when the former asked to (Q_6). However, while the pace of interaction was judged slightly more appropriate in the interactive case (Q_7), interaction was considered faster when using the standard version (Q_4). This partly explains the fact that users seemed more ready to use again the standard version of the system (Q_5).

7 out of 12 users (58.3%) answered the "preference" question Q_9 by saying that they preferred the interactive version. The reasons given by users in their

[7] chatterbean.bitoflife.cjb.net.
[8] trec.nist.gov



hello
Hi! Who is this?
my name is John
Hi, John! How can I help you?
I have a question about Shakespeare
Let's see if I can answer you.
Where was he born?
Do you mean to ask Where was SHAKESPEARE born ?
yes
The answers are on the right (you can click the links to know more!).Can I help you further?
no thanks
Ok.

Results

Your query: *Where was Shakespeare born*

Expected answer type: [PLACE HOW]

Text colors: ORANGE = NUMBER, TURQUOISE = TIME, RED = ORGANIZATION, GREEN = LOCATION, BLUE = PERSON, PURPLE = MATCHED TERMS, NAVY = QUERY TERMS

Answers

1. Title: Was Shakespeare Italian and born in Italy? - Literature Network Forums, **URL:** http://www.online-literature.com/forums/showthread.php?t=19753, **Google Rank:** 9, **file:** showthread.php?t=19753

luvara, but he has probably been studying literature for much longer few inconsistencies I'd like to point out. **There was definately a Will Stratford that.** April. John Shakespeare's son, William, was christe the town records.

2. Title: Shakespeare Quiz Questions at AbsoluteShakespeare.com **URL:** http://absoluteshakespeare.com/trivia/quiz/quiz.htm, **Google Rank:** 17, **file:** quiz.htm

Shakespeare Quiz. **Questions: 1) When was Shakespeare born 2 did Shakespeare write 3) Was Shakespeare ever in "love" 4) Wh wherefore art thou Romeo" 5) The line "To be or not to be" com**

Figure 1: YourQA's interactive interface

comments were mixed: while some of them were enthusiastic about the chatbot's small-talk features, others clearly said that they felt more comfortable with a search engine-like interface. Most of the critical aspects emerging from our overall satisfactory evaluation depend on the specific system we have tested rather than on the nature of interactive QA, to which none of such results appear to be detrimental.

We believe that the search-engine-style use and interpretation of QA systems are due to the fact that QA is still a very little known technology. It is a challenge for both developers and the larger public to cooperate in designing and discovering applications that take advantage of the potentials of interactivity.

5 A Unified Model

Our research so far has demonstrated the utility of personalization and interactivity in a QA system. It is thus inevitable to regard the formulation of a unified model of personalized, interactive QA as a valuable by-product of these two technologies. In this perspective, we propose the following dialogue scenario:

1. The user interacts with the dialogue interface formulating an utterance q;

2. If q is recognized as a question, it is analyzed by the dialogue manager (DM) to detect and resolve multiple and follow-up questions;

3. As soon as a resolved version q' of q is available, the DM passes q' to the QA module; the latter processes q' and retrieves a set $Retr(q')$ of relevant documents;

4. The QA module exchanges information with the UM component which is responsible of maintaining and updating the User Model of the current user, u; Based on u, the QA module extracts a list $L(q', u)$ of personalized results from $Retr(q')$;

5. The DM produces a reply r, which is returned along with $L(q', u)$ to the user via the dialogue interface;

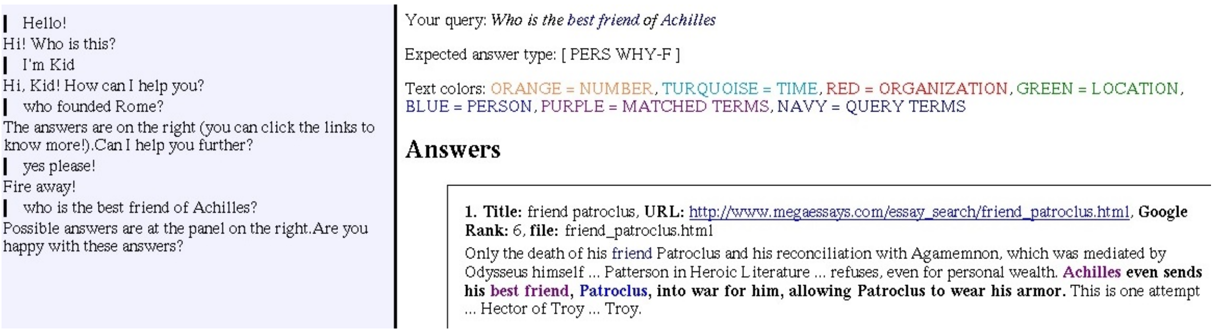
6. Once terminated, the current QA session is logged into the dialogue history $H(u)$, that will be used to update u;

Concerning step 4, an efficient strategy for eliciting the User Model from the user is yet to be specified at this stage: the current one relies on the definition of a context variable `userID` in the dialogue manager, which at the moment corresponds to the user's name. A number of AIML categories are created are created for YourQA to explicitly ask for the user's name, whihc is then assigned to the `userID` variable.

Figure 2 illustrates an example of a personalized, QA session in YourQA where the user's name is associated with a basic reading level UM. This affects the document retrieval phase, where only documents with simple words are retained for answer extraction.

6 Conclusions and Future Work

In this paper, we present an efficient and light-weight method to personalize the results of a Web-based QA system based on a User Model representing individual users' reading level, age range and interests. Our results show the efficiency of reading level estimation, and a



Hello!

Hi! Who is this?

I'm Kid

Hi, Kid! How can I help you?

who founded Rome?

The answers are on the right (you can click the links to know more!).Can I help you further?

yes please!

Fire away!

who is the best friend of Achilles?

Possible answers are at the panel on the right.Are you happy with these answers?

Your query: *Who is the best friend of Achilles*

Expected answer type: [PERS WHY-F]

Text colors: ORANGE = NUMBER, TURQUOISE = TIME, RED = ORGANIZATION, GREEN = LOCATION, BLUE = PERSON, PURPLE = MATCHED TERMS, NAVY = QUERY TERMS

Answers

1. Title: friend patroclus, **URL:** http://www.megaessays.com/essay_search/friend_patroclus.html, **Google Rank:** 6, **file:** friend_patroclus.html

Only the death of his friend Patroclus and his reconciliation with Agamemnon, which was mediated by Odysseus himself ... Patterson in Heroic Literature ... refuses, even for personal wealth. **Achilles even sends his best friend, Patroclus, into war for him, allowing Patroclus to wear his armor.** This is one attempt ... Hector of Troy ... Troy.

Figure 2: Screenshot from a personalized, interactive QA session. Here, the user's name ("Kid") is associated with a UM requiring a basic reading level, hence the candidate answer documents are filtered accordingly.

significant improvement in satisfaction when filtering answers based on the users' profile with respect to the baseline version of our system. Moreover, we introduce a dialogue management model for interactive QA based on a chat interface and evaluate it with optimistic conclusions.

In the future, we plan to study efficient strategies for bootstrapping User Models based on current and past conversations with the present user. Another problem to be solved is updating user interests and reading levels based on the dialogue history, in order to make the system fully adaptive.

Acknowledgements

The research reported here was mainly conducted at the Computer Science Department of the University of York, UK, under the supervision of Suresh Manandhar.

References

Belkin, N. J. and W.B. Croft. 1992. Information filtering and information retrieval: Two sides of the same coin? *Comm. ACM*, 35(12):29–38.

Collins-Thompson, K. and J. P. Callan. 2004. A language modeling approach to predicting reading difficulty. In *HLT/NAACL'04*.

Dahlbaeck, N., A. Jonsson, and L. Ahrenberg. 1993. Wizard of Oz studies: why and how. In *IUI '93*.

De Boni, M. and S. Manandhar. 2005. Implementing clarification dialogue in open-domain question answering. *JNLE*, 11.

Frank, E., G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *IJCAI '99*.

Kato, T., J. Fukumoto, F.Masui, and N. Kando. 2006. Woz simulation of interactive question answering. In *IQA'06*.

Kobsa, A. 2001. Generic user modeling systems. *UMUAI*, 11:49–63.

Kwok, C. T., O. Etzioni, and D. S. Weld. 2001. Scaling question answering to the web. In *WWW'01*.

Larsson, S., P. Ljunglöf, R. Cooper, E. Engdahl, and S. Ericsson. 2000. GoDiS—an accommodating dialogue system. In *ANLP/NAACL'00 WS on Conversational Systems*.

Magnini, B. and C. Strapparava. 2001. Improving user modelling with content-based techniques. In *UM'01*.

Maybury, M. T. 2002. Towards a question answering roadmap. Technical report, MITRE Corporation.

Moschitti, A., S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL'07*.

Munteanu, C. and M. Boldea. 2000. Mdwoz: A wizard of oz environment for dialog systems development. In *LREC'00*.

Quarteroni, S. and S. Manandhar. 2007a. A chatbot-based interactive question answering system. In *DECALOG'07*, Rovereto, Italy.

Quarteroni, S. and S. Manandhar. 2007b. User modelling for personalized question answering. In *AI*IA'07*, Rome, Italy.

Small, S., T. Liu, N. Shimizu, and T. Strzalkowski. 2003. HITIQA: an interactive question answering system- a preliminary report. In *ACL'03 WS on Multilingual summarization and QA*.

Sutton, S. 1998. Universal speech tools: the CSLU toolkit. In *ICSLP'98*.

Teevan, J., S. T. Dumais, and E. Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *SIGIR '05*.

Voorhees, E. M. 2003. Overview of the TREC 2003 Question Answering Track. In *TREC'03*.

Witten, I. H., G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *ACM DL*.