Adaptivity in Question Answering with User Modelling and a Dialogue Interface

Silvia Quarteroni and **Suresh Manandhar**
Department of Computer Science
University of York
York YO10 5DD
UK
{silvia,suresh}@cs.york.ac.uk

Abstract

Most question answering (QA) and information retrieval (IR) systems are insensitive to different users' needs and preferences, and also to the existence of multiple, complex or controversial answers. We introduce adaptivity in QA and IR by creating a hybrid system based on a dialogue interface and a user model. ***Keywords:*** *question answering, information retrieval, user modelling, dialogue interfaces.*

1 Introduction

While standard information retrieval (IR) systems present the results of a query in the form of a ranked list of relevant documents, question answering (QA) systems attempt to return them in the form of sentences (or paragraphs, or phrases), responding more precisely to the user's request.

However, in most state-of-the-art QA systems the output remains independent of the questioner's characteristics, goals and needs. In other words, there is a lack of *user modelling*: a 10-year-old and a University History student would get the same answer to the question: "*When did the Middle Ages begin?*". Secondly, most of the effort of current QA is on *factoid* questions, i.e. questions concerning people, dates, etc., which can generally be answered by a short sentence or phrase (Kwok et al., 2001). The main QA evaluation campaign, TREC-QA [1], has long focused on this type of questions, for which the simplifying assumption is that there exists only one correct answer. Even recent TREC campaigns (Voorhees, 2003; Voorhees, 2004) do not move sufficiently beyond the factoid approach. They account for two types of non-factoid *questions* –list and definitional– but not for non-factoid *answers*. In fact, a) TREC defines list questions as questions requiring multiple factoid answers, b) it is clear that a definition question may be answered by spotting definitional passages (what is not clear is how to spot them). However, accounting for the fact that some simple questions may have complex or controversial answers (e.g. "*What were the causes of World War II?*") remains an unsolved problem. We argue that in such situations returning a short paragraph or text snippet is more appropriate than exact answer spotting. Finally, QA systems rarely interact with the user: the typical session involves the user submitting a query and the system returning a result; the session is then concluded.

To respond to these deficiencies of existing QA systems, we propose an *adaptive* system where a QA module interacts with a user model and a dialogue interface (see Figure 1). The dialogue interface provides the query terms to the QA module, and the user model (UM) provides criteria to adapt query results to the user's needs. Given such information, the goal of the QA module is to be able to discriminate between simple/factoid answers and more complex answers, presenting them in a TREC-style manner in the first case and more appropriately in the second.

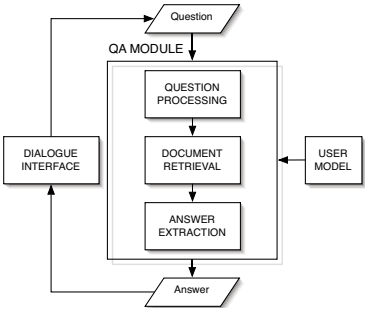

Figure 1: High level system architecture

Related work To our knowledge, our system is among the first to address the need for a different approach to non-factoid (complex/controversial)

[1] http://trec.nist.gov

answers. Although the three-tiered structure of our QA module reflects that of a typical web-based QA system, e.g. MULDER (Kwok et al., 2001), a significant aspect of novelty in our architecture is that the QA component is supported by the user model. Additionally, we drastically reduce the amount of linguistic processing applied during question processing and answer generation, while giving more relief to the post-retrieval phase and to the role of the UM.

2 User model

Depending on the application of interest, the UM can be designed to suit the information needs of the QA module in different ways. As our current application, YourQA[2], is a learning-oriented, web-based system, our UM consists of the user's:
1) age range, $a \in \{7-11, 11-16, adult\}$;
2) reading level, $r \in \{poor, medium, good\}$;
3) webpages of interest/bookmarks, w.
Analogies can be found with the SeAn (Ardissono et al., 2001) and SiteIF (Magnini and Strapparava, 2001) news recommender systems where age and browsing history, respectively, are part of the UM. In this paper we focus on how to filter and adapt search results using the reading level parameter.

3 Dialogue interface

The dialogue component will interact with both the UM and the QA module. From a UM point of view, the dialogue history will store previous conversations useful to construct and update a model of the user's interests, goals and level of understanding. From a QA point of view, the main goal of the dialogue component is to provide users with a friendly interface to build their requests. A typical scenario would start this way:

— *System:* *Hi, how can I help you?*

— *User:* *I would like to know what books Roald Dahl wrote.*

The query sentence *"what books Roald Dahl wrote"*, is thus extracted and handed to the QA module. In a second phase, the dialogue module is responsible for providing the answer to the user once the QA module has generated it. The dialogue manager consults the UM to decide on the most suitable formulation of the answer (e.g. short sentences) and produce the final answer accordingly, e.g.:

— *System:* *Roald Dahl wrote many books for kids and adults, including: "The Witches", "Charlie and the Chocolate Factory", and "James and the Giant Peach".*

[2] http://www.cs.york.ac.uk/aig/aqua

4 Question Answering Module

The flow between the three QA phases – question processing, document retrieval and answer generation – is described below (see Fig. 2).

4.1 Question processing

We perform query expansion, which consists in creating additional queries using question word synonyms in the purpose of increasing the recall of the search engine. Synonyms are obtained via the WordNet 2.0 [3] lexical database.

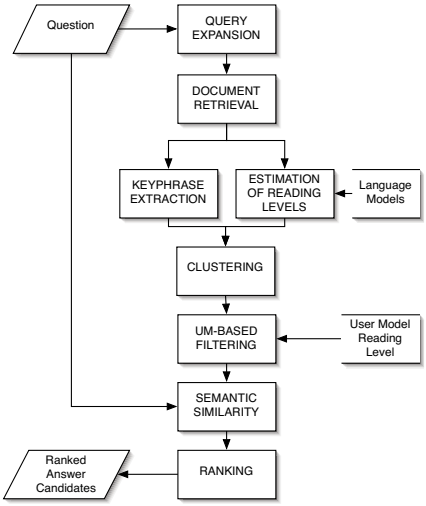

Figure 2: Diagram of the QA module

4.2 Retrieval

Document retrieval We retrieve the top 20 documents returned by Google[4] for each query produced via query expansion. These are processed in the following steps, which progressively narrow the part of the text containing relevant information.

Keyphrase extraction Once the documents are retrieved, we perform keyphrase extraction to determine their three most relevant topics using Kea (Witten et al., 1999), an extractor based on Naïve Bayes classification.

Estimation of reading levels To adapt the readability of the results to the user, we estimate the reading difficulty of the retrieved documents using the Smoothed Unigram Model (Collins-Thompson and Callan, 2004), which proceeds in

[3] http://wordnet.princeton.edu
[4] http://www.google.com

two phases. 1) In the training phase, sets of representative documents are collected for a given number of reading levels. Then, a unigram language model is created for each set, i.e. a list of *(word stem, probability)* entries for the words appearing in its documents. Our models account for the following reading levels: $poor$ (suitable for ages 7–11), $medium$ (ages 11–16) and $good$ (adults). 2) In the test phase, given an unclassified document D, its estimated reading level is the model lm_i maximizing the likelihood that $D \in lm_i$[5].

Clustering We use the extracted topics and estimated reading levels as features to apply hierarchical clustering on the documents. We use the WEKA (Witten and Frank, 2000) implementation of the Cobweb algorithm. This produces a tree where each leaf corresponds to one document, and sibling leaves denote documents with similar topics and reading difficulty.

4.3 Answer extraction

In this phase, the clustered documents are filtered based on the user model and answer sentences are located and formatted for presentation.

UM-based filtering The documents in the cluster tree are filtered according to their reading difficulty: only those compatible with the UM's reading level are retained for further analysis[6].

Semantic similarity Within each of the retained documents, we seek the sentences which are semantically most relevant to the query by applying the metric in (Alfonseca et al., 2001): we represent each document sentence p and the query q as word sets $\mathcal{P} = \{pw_1, \ldots, pw_m\}$ and $\mathcal{Q} = \{qw_1, \ldots, qw_n\}$. The distance from p to q is then $dist_q(p) = \sum_{1 \leq i \leq m} min_j[d(pw_i, qw_j)]$, where $d(pw_i, qw_j)$ is the word-level distance between pw_i and qw_j based on (Jiang and Conrath, 1997).

Ranking Given the query q, we thus locate in each document D the sentence p^* such that $p^* = argmin_{p \in D}[dist_q(p)]$; then, $dist_q(p^*)$ becomes the document score. Moreover, each cluster is assigned a score consisting in the maximal score of the documents composing it. This allows to rank not only documents, but also clusters, and present results grouped by cluster in decreasing order of document score.

Answer presentation We present our answers in an HTML page, where results are listed following the ranking described above. Each result consists of the title and clickable URL of the originating document, and the passage where the sentence which best answers the query is located and highlighted. Question keywords and potentially useful information such as named entities are in colour.

5 Sample result

We have been running our system on a range of queries, including factoid/simple, complex and controversial ones. As an example of the latter, we report the query *"Who wrote the Iliad?"*, which is a subject of debate. These are some top results:
— UM_{good}: *"Most Classicists would agree that, whether there was ever such a composer as "Homer" or not, the Homeric poems are the product of an oral tradition [...]* ***Could the Iliad and Odyssey have been oral-formulaic poems, composed on the spot by the poet using a collection of memorized traditional verses and phases?"***
— UM_{med}: *"No reliable ancient evidence for Homer – [...]* ***General ancient assumption that same poet wrote Iliad and Odyssey (and possibly other poems) questioned by many modern scholars: differences explained biographically in ancient world (e g wrote Od. in old age); but similarities could be due to imitation."***
— UM_{poor}: ***"Homer wrote The Iliad and The Odyssey (at least, supposedly a blind bard named "Homer" did)."***
In the three results, the problem of attribution of the *Iliad* is made clearly visible: document passages provide a context which helps to explain the controversy at different levels of difficulty.

6 Evaluation

Since YourQA does not single out one correct answer phrase, TREC evaluation metrics are not suitable for it. A user-centred methodology to assess how individual information needs are met is more appropriate. We base our evaluation on (Su, 2003), which proposes a comprehensive search engine evaluation model, defining the following metrics:
1. Relevance: we define strict precision (P_1) as the ratio between the number of results rated as relevant and all the returned results, and loose pre-

[5]The likelihood is estimated using the formula: $L_{i,D} = \sum_{w \in D} C(w, D) \cdot log(P(w|lm_i))$, where w is a word in the document, $C(w, d)$ is the number of occurrences of w in D and $P(w|lm_i)$ is the probability with which w occurs in lm_i

[6]However, if their number does not exceed a given threshold, we accept in our candidate set part of the documents having the next lowest readability – or a medium readability if the user's reading level is low

cision (P_2) as the ratio between the number of results rated as relevant or partially relevant and all the returned results.
2. User satisfaction: a 7-point Likert scale[7] is used to assess the user's satisfaction with loose precision of results (S_1) and query success (S_2).
3. Reading level accuracy: given the set $\mathcal{R}$ of results returned for a reading level r, A_r is the ratio between the number of results $\in \mathcal{R}$ rated by the users as suitable for r and $|\mathcal{R}|$.
4. Overall utility (U): the search session as a whole is assessed via a 7-point Likert scale.
We performed our evaluation by running 24 queries (some of which in Tab. 2) on Google and YourQA and submitting the results –i.e. Google result page snippets and YourQA passages– of both to 20 evaluators, along with a questionnaire. The relevance results (P_1 and P_2) in Tab. 1 show a

	P_1	P_2	S_1	S_2	U
Google	0,39	0,63	4,70	4,61	4,59
YourQA	0,51	0,79	5,39	5,39	5,57

Table 1: Evaluation results

10-15% difference in favour of YourQA for both strict and loose precision. The coarse semantic processing applied and context visualisation thus contribute to creating more relevant passages. Both user satisfaction results (S_1 and S_2) in Tab. 1 also denote a higher level of satisfaction tributed to YourQA. Tab. 2 shows that evaluators found our

Query	A_g	A_m	A_p
When did the Middle Ages begin?	0,91	0,82	0,68
Who painted the Sistine Chapel?	0,85	0,72	0,79
When did the Romans invade Britain?	0,87	0,74	0,82
Who was a famous cubist?	0,90	0,75	0,85
Who was the first American in space?	0,94	0,80	0,72
Definition of metaphor	0,95	0,81	0,38
average	**0,94**	**0,85**	**0,72**

Table 2: Sample queries and accuracy values

results appropriate for the reading levels to which they were assigned. The accuracy tended to decrease (from 94% to 72%) with the level: it is indeed more constraining to conform to a lower reading level than to a higher one. Finally, the general satisfaction values for U in Tab. 1 show an improved preference for YourQA.

[7]This measure – ranging from 1= "extremely unsatisfactory" to 7="extremely satisfactory" – is particularly suitable to assess how well a system meets user's search needs.

7 Conclusion

A user-tailored QA system is proposed where a user model contributes to adapting answers to the user's needs and presenting them appropriately. A preliminary evaluation of our core QA module shows a positive feedback from human assessors. Our short term goals involve performing a more extensive evaluation and implementing a dialogue interface to improve the system's interactivity.

References

E. Alfonseca, M. DeBoni, J.-L. Jara-Valencia, and S. Manandhar. 2001. A prototype question answering system using syntactic and semantic information for answer retrieval. In *Text REtrieval Conference*.

L. Ardissono, L. Console, and I. Torre. 2001. An adaptive system for the personalized access to news. *AI Commun.*, 14(3):129–147.

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

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference Research on Computational Linguistics (ROCLING X)*.

C. C. T. Kwok, O. Etzioni, and D. S. Weld. 2001. Scaling question answering to the web. In *World Wide Web*, pages 150–161.

Bernardo Magnini and Carlo Strapparava. 2001. Improving user modelling with content-based techniques. In *UM: Proceedings of the 8th Int. Conference*, volume 2109 of *LNCS*. Springer.

L. T. Su. 2003. A comprehensive and systematic model of user evaluation of web search engines: Ii. an evaluation by undergraduates. *J. Am. Soc. Inf. Sci. Technol.*, 54(13):1193–1223.

E. M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Text REtrieval Conference*.

E. M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Text REtrieval Conference*.

H. Witten and E. Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation*. Morgan Kaufmann.

I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255.