Incorporating User Models in Question Answering to Improve Readability

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

Abstract

Most question answering and information retrieval systems are insensitive to different users' needs and preferences, as well as their reading level. In (Quarteroni and Manandhar, 2006), we introduce a hybrid QA-IR system based on a a user model.
In this paper we focus on how the system filters and re-ranks the search engine results for a query according to their reading difficulty, providing user-tailored answers.
Keywords: *question answering, information retrieval, user modelling, readability.*

1 Introduction

Question answering (QA) systems are information retrieval systems accepting queries in natural language and returning the results in the form of sentences (or paragraphs, or phrases). They move beyond standard information retrieval (IR) where results are presented in the form of a ranked list of query-relevant documents. Such a finer answer presentation is possible thanks to the application of computational linguistics techniques in order to filter irrelevant documents, and of a consistent amount of question pre-processing and result post-processing.

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*. For instance, an elementary school child and a University history student would get the same answer to the question: "*When did the Middle Ages begin?*".

Secondly, most QA systems focus on *factoid* questions, i.e. questions concerning people, dates, numerical quantities etc., which can generally be answered by a short sentence or phrase (Kwok et al., 2001). The mainstream approach to QA evaluation, represented by TREC-QA campaigns[1], has long fostered the criterion that a "good" system is one that returns the "correct" answer in the shortest possible formulation. Although recent efforts in TREC 2003 and 2004 (Voorhees, 2003; Voorhees, 2004) denoted an interest towards list questions and definitional (or "other") questions, we believe that there has not been enough interest towards non-factoid *answers*. The real issue is "realizing" that the answer to a question is sometimes too complex to be formulated and evaluated as a factoid: some queries have multiple, complex or controversial answers (take e.g. "*What were the causes of World War II?*"). In such situations, returning a short paragraph or text snippet is more appropriate than exact answer spotting. For instance, the answer to "*What is a metaphor?*" may be better understood with the inclusion of examples. This viewpoint is supported by recent user behaviour studies which showed that even in the case of factoid-based QA systems, the most eligible result format consisted in a paragraph where the sentence containing the answer was highlighted (Lin et al., 2003).

The issue of non-factoids is related to the user modelling problem: while factoid answers do not necessarily require to be contextualized within the user's knowledge and viewpoint, the need is much stronger in the case of definitions, explanations and descriptions. This is mentioned in the TREC 2003 report (Voorhees, 2003) when discussing the evaluation of definitional questions: however, the issue is expeditiously solved by assuming a fixed user profile (the "average news reader").

We are currently developing an *adaptive* system which adjusts its output with respect to a user model. The system can be seen as an enhanced IR system which adapts both the content and presentation of the final results, improving their *quality*.

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

In this paper, we show that QA systems can benefit from the contribution of user models, and explain how these can be used to filter the information presented as an answer based on readability. Eventually, we describe preliminary results obtained via an evaluation framework inspired by user-centered search engine evaluation.

2 System Architecture

The high-level architecture as represented in Figure 1 shows the basic components of the system, the QA module and the user model.

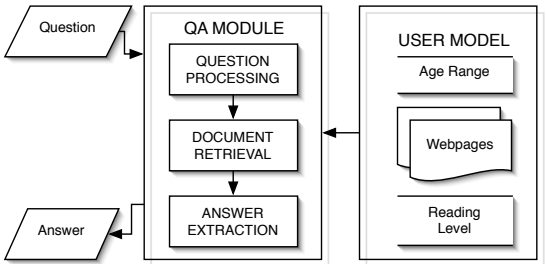

Figure 1: High level system architecture

The QA module, described in the following section, is organized according to the three-tier partition underlying most state-of-the-art systems: 1) question processing, 2) document retrieval, 3) answer generation. The module makes use of a web search engine for document retrieval and consults the user model to obtain the criteria to filter and re-rank the search engine results and to eventually present them appropriately to the user.

2.1 User model

Depending on the application of interest, the user model (UM) can be designed to suit the information needs of the QA module in different ways. Our current application, YourQA[2], is a learning-oriented system to help students find information on the Web for their assignments. Our UM consists of the user's:

- age range, $a \in \{7-11, 11-16, adult\}$
- reading level, $r \in \{poor, medium, good\}$
- webpages of interest/bookmarks, w

The age range parameter has been chosen to match the partition between primary school, contemporary school and higher education age in Britain; our reading level parameter takes three values which ideally (but not necessarily) correspond to the three age ranges and may be further refined in the future for more fine-grained modelling.

Analogies can be found with the SeAn (Ardissono et al., 2001), and SiteIF (Magnini and Strapparava, 2001) news recommender systems, where information such as age and browsing history, resp. are part of the UM. More generally, our approach is similar to that of personalized search systems (Teevan et al., 2005; Pitkow et al., 2002), which construct UMs based on the user's documents and webpages.

In our system, UM information is explicitly collected from the user; while age and reading level are self-assessed, the user's interests are extracted from the document set w using a keyphrase extractor (see further for details). Eventually, a dialogue framework with a history component will contribute to the construction and update of the user model in a less intruding and thus more user-friendly way. In this paper we focus on how to adapt search result presentation using the reading level parameter: age and webpages will not be discussed.

2.2 Related work

Non-factoids and user modelling As mentioned above, the TREC-QA evaluation campaign, to which the vast majority of current QA systems abide, mainly approaches factoid-based answers. To our knowledge, our system is among the first to address the need for a different approach to non-factoid *answers*. The structure of our QA component reflects the typical structure of a web-based QA system in its three-tier composition. Analogies in this can be found for instance in MULDER (Kwok et al., 2001), which is organized according to a question processing/answer extraction/passage ranking pipeline. However, a significant aspect of novelty in our architecture is that the QA component is supported by the user model.

Additionally, we have changed the relative importance of the different tiers: while we drastically reduce linguistic processing during question processing and answer generation, we give more relief to the post-retrieval phase and to the role of the UM. Having removed the need for fine-grained answer spotting, the emphasis is shifted towards finding closely connected sentences that are highly

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

relevant to answer the query.

Readability Within computational linguistics, several applications have been designed to address the needs of users with low reading skills. The computational approach to textual adaptation is commonly based on *natural language generation*: the process "translate" a difficult text into a syntactically and lexically simpler version. In the case of PSET (Carroll et al., 1999) for instance, a tagger, a morphological analyzer and generator and a parser are used to reformulate newspaper text for users affected by aphasia. Another interesting research is Inui et al.'s lexical and syntactical paraphrasing system for deaf students (Inui et al., 2003). In this system, the judgment of experts (teachers) is used to learn selection rules for paraphrases acquired using various methods (statistical, manual, etc.). In the SKILLSUM project (Williams and Reiter, 2005), used to generate literacy test reports, a set of choices regarding output (cue phrases, ordering and punctuation) are taken by a micro-planner based on a set of rules.

Our approach is conceptually different from the above: exploiting the wealth of information available in the context of a Web-based QA system, we can afford to *choose* among the documents available on a given subject those which best suit our readability requirements. This is possible thanks to the versatility of language modelling, which allows us to tailor the readability estimation of documents to any kind of user profile in a dynamic manner, as explained in section 3.2.3.

3 QA Module

In this section we discuss the information flow among the subcomponents of the QA module (see Figure 2 for a representative diagram) and focus on reading level estimation and document filtering. For further details on the implementation of the QA module, see (Quarteroni and Manandhar, 2006).

3.1 Question Processing

The first step performed by YourQA is query expansion: additional queries are created replacing question terms with synonyms using WordNet[3].

[3] `http://wordnet.princeton.edu`

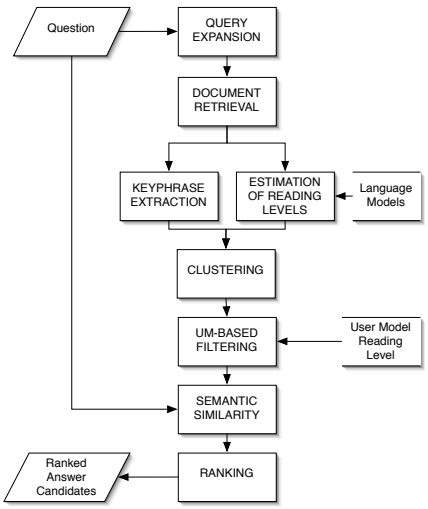

Figure 2: Diagram of the QA module

3.2 Retrieval and Result Processing

3.2.1 Document retrieval

We use Google[4] to retrieve the top 20 documents returned for each of the queries issued from the query expansion phase. The subsequent steps will progressively narrow the parts of these documents where relevant information is located.

3.2.2 Keyphrase extraction

Keyphrase extraction is useful in two ways: first, it produces features to group the retrieved documents thematically during the clustering phase, and thus enables to present results by groups. Secondly, when the document parameter (w) of the UM is active, matches are sought between the keyphrases extracted from the documents and those extracted from the user's set of interesting documents; thus it is possible to prioritize results which are more compatible with his/her interests.

Hence, once the documents are retrieved, we extract their keyphrases using Kea (Witten et al., 1999), an extractor based on Naïve Bayes classification. Kea first splits each document into phrases and then takes short subsequences of these initial phrases as candidate keyphrases. Two attributes are used to classify a phrase p as a keyphrase or a non-keyphrase: its TF $\times$ IDF score within the set of retrieved documents and the index of p's first appearance in the document. Kea outputs a ranked list of phrases, among which we select the top three as keyphrases for each of our documents.

[4] `http://www.google.com`

3.2.3 Estimation of reading levels

In order to adjust search result presentation to the user's reading ability, we estimate the reading difficulty of each retrieved document using the Smoothed Unigram Model, a variation of a Multinomial Bayes classifier (Collins-Thompson and Callan, 2004). Whereas other popular approaches such as Flesch-Kincaid (Kincaid et al., 1975) are based on sentence length, the language modelling approach accounts especially for lexical information. The latter has been found to be more effective as the former when approaching the reading level of subjects in primary and secondary school age (Collins-Thompson and Callan, 2004). Moreover, it is more applicable than length-based approach for Web documents, where sentences are typically short regardless of the complexity of the text.

The language modelling approach proceeds in two phases: in the training phase, given a range of reading levels, a set of representative documents is collected for each reading level. A unigram language model lm_s is then built for each set s; the model consists of a list of the word stems appearing in the training documents with their individual probabilities. Textual readability is not modelled at a conceptual level: thus complex concepts explained in simple words might be classified as suitable even for a poor reading level; However we have observed that in most Web documents lexical, syntactic and conceptual complexity are usually consistent within documents, hence it makes sense to apply a reasoning-free technique without impairing readability estimation. Our unigram language models account for the following reading levels:
1) *poor*, i.e. suitable for ages 7 – 11;
2) *medium*, suitable for ages 11–16;
3) *good*, suitable for adults.
This partition in three groups has been chosen to suit the training data available for our school application, which consists of about 180 HTML pages (mostly from the "BBC schools"[5], "Think Energy"[6], "Cassini Huygens resource for schools"[7] and "Magic Keys storybooks"[8] websites), explicitly annotated by the publishers according to the reading levels above.

In the test phase, given an unclassified document D, the estimated reading level of D is the language model lm_i maximizing the likelihood $L(lm_i|D)$ that D has been generated by lm_i. Such likelihood is estimated using the formula:

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

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

An advantage of language modelling is its portability, since it is quite quick to create word stem/frequency histograms on the fly. This implies that models can be produced to represent more fine-grained reading levels as well as the specific requirements of a single user: the only necessary information are sets of training documents representing each level to be modelled.

3.2.4 Clustering

As an indicator of inter-document relatedness, we use document clustering (Steinbach et al., 2000) to group them using both their estimated reading difficulty and their topic (i.e. their keyphrases). In particular we use a hierarchical algorithm, Cobweb (implemented using the WEKA suite of tools (Witten and Frank, 2000) as it produces a cluster tree which is visually simple to analyse: each leaf corresponds to one document, and sibling leaves denote documents that are strongly related both in topic and in reading difficulty. Figure 3 illustrates an example cluster tree for the the query: *"Who painted the Sistine chapel?"*. Leaf labels represent document keyphrases extracted by Kea for the corresponding documents and ovals represent non-terminal nodes in the cluster tree (these are labelled using the most common keyphrases in their underlying leaves).

3.3 Answer Extraction

The purpose of answer extraction is to present the most interesting excerpts of the retrieved documents according to both the user's query topics and reading level. This process, presented in sections 3.3.1 – 3.3.4, follows the diagram in Figure 2: we use the UM to filter the clustered documents, then compute the similarity between the question and the filtered document passages in order to return the best ones in a ranked list.

[5] `http://bbc.co.uk/schools`
[6] `http://www.think-energy.com`
[7] `http://www.pparc.ac.uk/Ed/ch/Home.htm`
[8] `http://www.magickeys.com/books/`

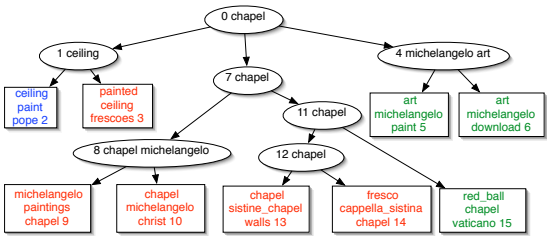

Figure 3: Cluster tree for *"Who painted the Sistine chapel?"*. Leaf 3 and the leaves grouped under nodes 8 and 12 represent documents with an estimated good reading level; leaf 15 and the leaves underlying node 4 have a medium reading level; leaf 2 represents a poor reading level document.

3.3.1 UM-based filtering

The documents in the cluster tree are filtered according to the UM reading level, r: only those compatible with the user's reading ability are retained for further analysis. However, if the number of retained documents does not exceed a given threshold, we accept in our candidate set part of the documents having the next lowest readability in case $r \in \{good, medium\}$ or a medium readability in case $r = poor$.

3.3.2 Semantic similarity

Within each of the documents retained, we seek for the sentences which are semantically most relevant to the query. Given a sentence p and the query q, we represent them as two sets of words $\mathcal{P} = \{pw_1, \ldots, pw_m\}$ and $\mathcal{Q} = \{qw_1, \ldots, qw_n\}$. The semantic distance from p to q is then:
$dist_q(p) = \sum_{1 \leq i \leq m} min_j[d(pw_i, qw_j)]$
where $d(pw_i, qw_j)$ represents the Jiang-Conrath word-level distance between pw_i and qw_j (Jiang and Conrath, 1997), based on WordNet 2.0. The intuition is that for each question word, we find the word in the candidate answer sentence which minimizes the word-level distance and then we compute the sum of such minima.

3.3.3 Passage and cluster ranking

For a given document, we can thus isolate a sentence s minimizing the distance to the query. The passage P, i.e. a window of up to 5 sentences centered on s, will be a candidate result. We assign to such passage a *score* equal to the similarity of s to the query; in turn, the score of P is used as the score of the document containing it. We also define a ranking function for clusters, which allows to order them according to the maximal score of their component documents. Passages from the highest ranking cluster will be presented first to the user, in decreasing order of score, followed by the passages from lower ranking clusters.

3.3.4 Answer presentation

To present our answers, we fix a threshold for the number of results to be returned following the ranking exposed above. Each result consists of a title and document passage where the sentence which best answers the query is highlighted; the URL of the original document is also available for loading if the user finds the passage interesting and wants to read more.

4 Results

We report the results of running our system on a range of queries, which include factoid/simple, complex and controversial questions[9].

4.1 Simple answer

As an example of a simple query, we present the results for: *"Who painted the Sistine Chapel?"*, the system returned the following passages:
—UM_{good}: ***"Sistine Chapel (sis-teen). A chapel adjoining Saint Peter's Basilica, noted for the frescoes of biblical subject painted by Michelangelo on its walls and ceilings."***
—UM_{med}: ***"In all Michelangelo painted more than 300 different figures on the Sistine Chapel ceiling."***
—UM_{poor}: ***"My name is Jacopo L'Indaco and I was an assistant to Michelangelo when he painted the Sistine Chapel."***
To obtain the above answers the system was run 3 times with different values for the reading level parameter in the UM, as defined in 3.2.3. As we can see, in all cases the correct information is present although not always explicitly, as in the first two cases. This is because our current semantic similarity metric only operates at word level. In this example, all sentences containing "painted", "Sistine" and "Chapel" obtain a distance of 0 to the query, regardless of their formulation. Also notice how the difference in language complexity is clearly discernible in the different answers.

4.2 Complex answer

We illustrate the results of our system with the query *"Definition of metaphor"*, which relates to a difficult concept. Our top results, highlighted in

[9]Notice that this partition is not to be interpreted as a methodological division, as we currently approach complex and controversial answers the same way.

boldface within their containing passages, are:

—UM_{good}: *"**Definition of Metaphor. Metaphor is a process by which we conceive "one thing in terms of another, and its primary function is understanding" (Lakoff and Johnson, Metaphors We Live By 36).** That is, we often do (and must) talk about things that do not physically exist, that we cannot reach out and touch or take apart with our hands to see what is inside and how it works: love, time, anger, life. [...] We make them tangible by, in a process of thought, applying models of physical things to them: love is falling; argument is war. "*

—UM_{med}: *"**Definition: A metaphor is a figure of speech that compares unlike things by saying that one thing is the other.** Often metaphors are simple comparisons, but they can be extended so that different aspects of the things compared are treated separately. "*

—UM_{poor}: *we did not find any document exactly matching the reading level, we thus returned the results for UM_{med} (see above).*

As before, there is a clear difference in the language complexity of the two answers.

4.3 Controversial answer

We illustrate this case using the query *"Who wrote the Iliad?"*, which is a subject of debate: the existence of a single author, as well as the historicity of the poet Homer are under discussion. These are among our 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, a generations-old technique that was the collective inheritance of many singer-poets, aoidoi. [...]* ***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 – earliest traditions involve conjecture (e.g. conflicting claims to be his place of origin) and legend (e.g. Homer as son of river-god).* ***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 this case we can see how the problem of attribution of the *Iliad* is made clearly visible: in the three results, document passages provide a context which helps to explain such controversy at different levels of difficulty.

5 Evaluation

5.1 Methodology

Our system is not a QA system in the strict sense, as it does not single out one correct answer phrase. The key objective is an improved satisfaction of the user towards its adaptive results, which are hopefully more suitable to his reading level. A user-centred evaluation methodology that assesses how the system meets individual information needs is therefore more appropriate for YourQA than TREC-QA metrics.

We draw our evaluation guidelines from (Su, 2003), which proposes a comprehensive search engine evaluation model. We define the following metrics (see Table 1):

1. *Relevance*:
   - strict precision (P_1): the ratio between the number of results rated as relevant and all the returned results,
   - loose precision (P_2): 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[10] is used to assess satisfaction with:
   - loose precision of results (S_1),
   - query success (S_2).
3. *Reading level accuracy (A_r)*. This metric was not present in (Su, 2003) and has been introduced to assess the reading level estimation. Given the set $\mathcal{R}$ of results returned by the system for a reading level r, it is the ratio between the number of documents $\in \mathcal{R}$ rated by the users as suitable for r and $|\mathcal{R}|$. We compute A_r for each reading level.
4. *Overall utility (U)*: the search session as a whole is assessed via a 7-point Likert scale.

We have discarded some of the metrics proposed by (Su, 2003) when they appeared as linked to technical aspects of search engines (e.g. connectivity), and when response time was concerned as at the present stage this has not been considered

[10] This measure – ranging from 1= "extremely unsatisfactory" to 7="extremely satisfactory" – is particularly suitable to assess the degree to which the system meets the user's search needs. It was reported in (Su, 1991) as the best single measure for information retrieval among 20 tested.

an issue. Also, we exclude metrics relating to the user interface which are not relevant for this study.

Metric	*field*	*description*
Relevance	P_1	strict precision
	P_2	loose precision
Satisfaction	S_1	with loose precision
	S_2	with query success
Accuracy	A_g	good reading level
	A_m	medium reading level
	A_p	poor reading level
Utility	U	overall session

Table 1: Summary of evaluation metrics

5.2 Evaluation results

We performed our evaluation by running 24 queries (partly reported in Table 3) on both Google and YourQA[11]. The results – i.e. snippets from the Google result page and passages returned by YourQA – were given to 20 evaluators. These were aged between 16 and 52, all having a self-assessed good or medium English reading level. They came from various backgrounds (University students/graduates, professionals, high school) and mother-tongues. Evaluators filled in a questionnaire assessing the relevance of each passage, the success and result readability of the single queries, and the overall utility of the system; values were thus computed for the metrics in Table 1.

	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 2: Evaluation results

5.2.1 Relevance

The precision results (see Table 2) for the whole search session were computed by averaging the values obtained for the 20 queries. Although quite close, they show a 10-15% difference in favour of the YourQA system for both strict precision (P_1) and loose precision (P_2). This suggests that the coarse semantic processing applied and the visualisation of the context contribute to the creation of more relevant passages.

[11]To make the two systems more comparable, we turned off query expansion and only submitted the original question sentence

5.2.2 User satisfaction

After each query, we asked evaluators the following questions: *"How would you rate the ratio of relevant/partly relevant results returned?"* (assessing S_1) and *"How would you rate the success of this search?"* (assessing S_2). Table 2 denotes a higher level of satisfaction tributed to the YourQA system in both cases.

5.2.3 Reading level accuracy

Adaptivity to the users' reading level is the distinguishing feature of the YourQA system: we were thus particularly interested in its performance in this respect. Table 3 shows that altogether, evaluators found our results appropriate for the reading levels to which they were assigned. The accuracy tended to decrease (from 94% to 72%) with the level: this was predictable as it is more constraining to conform to a lower reading level than to a higher one. However this also suggests that our estimation of document difficulty was perhaps too "optimisitic": we are currently working with better quality training data which allows to obtain more accurate language models.

Query	A_g	A_m	A_p
Who painted the Sistine Chapel?	0,85	0,72	0,79
Who was the first American in space?	0,94	0,80	0,72
Who was Achilles' best friend?	1,00	0,98	0,79
When did the Romans invade Britain?	0,87	0,74	0,82
Definition of metaphor	0,95	0,81	0,38
What is chickenpox?	1,00	0,97	0,68
Define german measles	1,00	0,87	0,80
Types of rhyme	1,00	1,00	0,79
Who was a famous cubist?	0,90	0,75	0,85
When did the Middle Ages begin?	0,91	0,82	0,68
Was there a Trojan war?	0,97	1,00	0,83
Shakespeare's most famous play?	0,90	0,97	0,83
average	**0,94**	**0,85**	**0,72**

Table 3: Queries and reading level accuracy

5.2.4 Overall utility

At the end of the whole search session, users answered the question: *"Overall, how was this search session?"* relating to their search experience with Google and the YourQA system. The values obtained for U in Table 2 show a clear preference (a difference of $\simeq 1$ on the 7-point scale) of the users for YourQA, which is very positive con-

sidering that it represents their general judgement on the system.

5.3 Future work

We plan to run a larger evaluation by including more metrics, such as user *vs* system ranking of results and the contribution of cluster by cluster presentation. We intend to conduct an evaluation also involving users with a poor reading level, so that each evaluator will only examine answers targeted to his/her reading level. We will analyse our results with respect to the individual reading levels and the different types of questions proposed.

6 Conclusion

A user-tailored open domain QA system is outlined where a user model contributes to elaborating answers corresponding to the user's needs and presenting them efficiently. In this paper we have focused on how the user's reading level (a parameter in the UM) can be used to filter and re-order the candidate answer passages. Our preliminary results show a positive feedback from human assessors on the utility of the system in an information seeking domain. Our short term goals involve performing a more extensive evaluation, exploiting more UM parameters in answer selection and implementing a dialogue interface to improve the system's interactivity.

References

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

J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of EACL'99*, pages 269–270.

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

K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura. 2003. Text simplification for reading assistance: a project note. In *ACL Workshop on Paraphrasing: Paraphrase Acquisition and Applications*, pages 9–16.

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)*.

J. Kincaid, R. Fishburne, R. Rodgers, and B. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel. Technical Report Branch Report 8-75, Chief of Naval Training.

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

J. Lin, D. Quan, V. Sinha, and K Bakshi. 2003. What makes a good answer? the role of context in question answering. In *Proceedings of INTERACT 2003*.

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.

James Pitkow, Hinrich Schuetze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. 2002. Personalized search. *Commun. ACM*, 45(9):50–55.

S. Quarteroni and S. Manandhar. 2006. User modelling for adaptive question answering and information retrieval. In *Proceedings of FLAIRS'06*.

M. Steinbach, G. Karypid, and V. Kumar. 2000. A comparison of document clustering techniques.

L. T. Su. 1991. *An investigation to find appropriate measures for evaluating interactive information retrieval*. Ph.D. thesis, New Brunswick, NJ, USA.

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.

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of SIGIR '05*, pages 449–456, New York, NY, USA. ACM Press.

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*.

S. Williams and E. Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proceedings of ENLG-2005*, pages 140–147.

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.