Personalized Question Answering

Silvia Quarteroni

DISI - University of Trento, Italy
silvia.quarteroni@disi.unitn.it

ABSTRACT. *A common problem in Question Answering – and Information Retrieval in general – is information overload, i.e. an excessive amount of data from which to search for relevant information. This results in the risk of high recall but low precision of the information returned to the user. In turn, this affects the relevance of answers with respect to the users' needs, as queries can be ambiguous and even answers extracted from documents with relevant content may be ill-received by users if they are too difficult (or simple) for them. We address the issue by integrating a User Modelling component to personalize the results of a Web-based open-domain Question Answering system based on the user's reading level and interests.*

RÉSUMÉ. *Un problème commun aux systèmes de Question-Réponse, et de recherche documentaire plus généralement, est la présence d'une quantité excessive de données parmi lesquelles chercher l'information pertinente. Ceci apporte un rappel élevé mais se traduit par un risque de faible précision de l'information retournée à l'utilisateur. De plus, ce problème concerne la pertinence des réponses vis-à-vis des besoins des utilisateurs, puisque les questions peuvent être ambiguës et même des réponses extraites de documents au contenu pertinent peuvent être mal reçues si elles sont trop compliquées (ou trop simples) pour les utilisateurs. Nous adressons ce problème en incorporant une composante de modélisation de l'utilisateur afin de personnaliser les résultats d'un système Web de Question-Réponse à domaine ouvert sur la base de son niveau de lecture et de ses interêts.*

KEYWORDS: *Question Answering, personalization, User Modelling.*

MOTS-CLÉS : *systèmes de Question-Réponse, personnalisation, modélisation des utilisateurs.*

TAL. Volume 51 – n° 1/2010, pages 97 à 123

1. Introduction

Question Answering (QA) can be defined as the task that, given a natural language question, aims at finding one or more concise answers in the form of sentences or phrases. The discipline of Question Answering began in the 1960s as a way to access small databases via natural language rather than query languages (Simmons, 1965). Since then, research has evolved to the current, ambitious goal of open-domain QA systems, extracting answers to any sort of questions (from names to descriptions) from large data repositories or even the Web (Kwok *et al.*, 2001). Indeed, current QA is often regarded as a cross-road discipline that tackles an information retrieval (IR) problem by applying sophisticated natural language processing (NLP) techniques to identify the complex syntactic and semantic relations in text. This allows the creation of pin-pointed responses to the user's information needs.

A commonly observed problem in Web-based Question Answering and information retrieval alike is information overload, i.e. the presence of an excessive amount of sources of potentially relevant information. This results in the risk of high recall but low precision of the information returned to the user, as underlined in (Belkin and Croft, 1992). In the open domain in particular, where the scope and type of questions is not constrained, 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 may be ill-received by users if they find their reading level overly simple or too complex. For instance, a primary school child and a History student may need different answers to the question: *When did the Middle Ages begin?* Ideally, a QA system should not only return relevant results, but also take advantage of a large body of different formulations of the same relevant content – as available on the Web – by selecting the formulations that best suit its current user.

The need for personalization has been postulated by the information retrieval community for a long time (Belkin and Croft, 1992), although seldom addressed in terms of natural language processing. Indeed, typical solutions for IR personalization derive from the analysis of browsing behavior, such as clicks, pointing, eye gaze, and other non-textual information providing insights on the user's navigation style and speed (Shen *et al.*, 2005). Even content-based techniques for IR personalization have mostly relied on non-linguistic representations such as the vector space model. Here, user profiles are represented as vectors whose dimensions are specific items and preferences are inferred by some form of regression learned via a statistical model (Zhang and Koren, 2007).

In the Question Answering community, very little effort has been carried out in the direction of personalization, if we set aside the advocation of personalization in the 2003 edition of TREC-QA, the major evaluation campaign [1] (Voorhees, 2003). Even here, the issue was expeditiously addressed by assuming a scenario where an "average

1. "Without any idea of who the questioner is and why he or she is asking the question, it is essentially impossible for a system to decide what level of detail in a response is appropriate

news reader" submitted the 2003 task's *definition* questions. It is clear that the problem of personalization in Question Answering is just postponed by such a solution; moreover, the issue does not solely concern certain types of questions. It remains the case that different users have different information needs and that Question Answering systems providing personalized results at least as an optional feature would play an important role among the current directions for improving this technology.

This work shows 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 regard this as an application of User Modelling (Kobsa, 2001) to personalized Question Answering, inspired by prior applications of the technique in item recommendation (Miller *et al.*, 2003), intelligent tutoring (Virvou and Moundridou, 2001), cultural heritage (Stock, 1993) and, in rare cases, information retrieval (Teevan *et al.*, 2005). As a direct consequence of the lack of personalization research in Question Answering, another missing aspect is a rigorous evaluation framework in the context of which a personalized system can be compared to a baseline, "standard" system on the grounds of quantitative or qualitative measurements. This issue is also addressed in this article by designing an evaluation methodology for personalized QA systems and reporting the results achieved by comparing the personalized and standard version of a specific system.

In this article, section 2 briefly discusses previous approaches to personalization in IR and natural language applications. Section 3 introduces the design of a personalized open-domain, Web-based QA system, characterized by a User Model, while section 4 outlines the attributes of such a model. Section 5 shows how the reading level of Web pages can be estimated and used to filter out unsuitable answers, and illustrates how the user's interests can be matched to candidate answer documents to achieve personalization. Section 6 describes the implementation of the personalized version of our Web-based QA system, YourQA. Section 7 introduces a methodology for user-centered evaluation of personalized QA and reports the positive results of such methodology. Finally, a number of conclusions are drawn in section 8.

2. Related work

While the vast majority of IR systems does not provide user-sensitive results, from the late 1990s the value of Web personalization has been of increasing interest in item recommenders, where personalization allows product offerings, sales promotions, etc to be targeted to each individual user. This is generally achieved by creating a User Model from various aspects of previous interactions with the electronic merchant. For instance, in MOVIELENS (Miller *et al.*, 2003) predictions on possibly interesting movies are made based on explicit ratings provided by users and implicit ratings derived from navigational data and previously purchased products.

– presumably an elementary-school-aged child and a nuclear physicist should receive different answers for at least some questions." (Voorhees, 2003)

Indeed, User Modelling in Web applications is rarely achieved via NLP techniques, but rather by representing user profiles as regression models where the vector space is represented by recommendation items; then, approaches such as Bayesian hierarchical classification are adopted to learn the optimal parameters for each profile (Zhang and Koren, 2007). In contrast, a limited number of personalized IR systems take advantage of NLP approaches: (Alfonseca and Rodriguez, 2003) propose an adaptive system that adjusts the contents of existing Web sites to the needs of the users based on a model of the user's interests; the latter are obtained using topic identification and document classification techniques. (Paik *et al.*, 2001) also describes a technique based on topic categorization that extracts personalized information from email communications between financial analysts and their clients.

Not only few personalization systems make use of natural language techniques, but also few natural language systems deal with the issue of personalization: research in this direction mostly deals with natural language generation. For instance, in the PSET system (Carroll *et al.*, 1999), a tagger, a morphological analyzer/generator, and a parser reformulate newspaper text for users affected by aphasia; in (Inui *et al.*, 2003)'s lexical and syntactical paraphrasing system for deaf students, the judgment of experts is used to learn paraphrases selection rules. Again, in (Mairesse and Walker, 2008), a data-driven generation method projecting multiple personality traits is used in a restaurant recommendation service.

In other NLP areas, such as QA, the design of full-fledged personalized systems is rare. Indeed, a use case in the context of Business Analysis is proposed in (Thai *et al.*, 2006), where a formal specification of the semantics of a closed-domain is argued to be useful to provide user-tailored answers; however, the above approach does not return personalized answers based on user profiles, but rather provides users with tools to control answer extraction; moreover, it can be argued that a formal domain specification is unimaginable for an open-domain QA system. Again, (Liu and Agichtein, 2008) approach the problem of predicting individual users' satisfaction with respect to answers in an online answer collection, but do not propose an algorithm to personalize the answers returned by a QA system.

The main contribution of this work, with respect to the above approaches, can be regarded as the definition of a personalized QA model that suits open-domain (Web) applications rather than a limited domain. The approach to personalized QA discussed in the following sections has the distinguishing feature of *(semi-)automatically creating a User Model* by taking advantage of IR approaches to personalized relevance computation and NLP approaches such as automatic concept extraction. In particular, the following techniques are borrowed and adapted from different applications towards the construction of a User Model: the collection of personal data and explicit feedback from users, the automatic acquisition of implicit information such as textual readability, as well as the automatic extraction of topics and tags.

3. Architecture of an open-domain, Web based Question Answering system

The presented approach to the design of a personalized Question Answering system takes as a baseline a Web-based, open-domain QA system and illustrates the additional components it requires in order to provide personalized answers. Such an approach has the benefit of allowing a head-to-head evaluation with respect to a standard QA system. The baseline QA system architecture assumed here follows the high-level organization shared by many Web QA systems, involving the three subsequent phases of question processing, document retrieval and answer extraction (Kwok *et al.*, 2001).

As a case study, the YourQA system is adopted (Quarteroni and Manandhar, 2009). The latter's main feature is the ability to use the Web as a resource to find answers to both factoid questions, i.e. questions requiring fact-based responses such as names or dates, and non-factoid questions, such as definition and *how-to* questions. YourQA's Question Answering algorithm can be summarized as follows:

1) **Question Processing**:

a) the query is classified and the two top expected answer types are estimated;

b) the query is submitted to the underlying search engine [2];

2) **Document Retrieval**:

a) the top n documents are retrieved from the underlying search engine;

b) the selected documents are split into sentences;

3) **Answer Extraction**:

a) a sentence-level similarity metric is applied to the query and to each retrieved document sentence to identify candidate answer sentences;

b) candidate answers are ordered by relevance to the query; the search engine rank of the answer source document is used as a tie-breaking criterion;

c) the list of top ranked answers is then returned to the user in an HTML page.

The implementation details of the different QA phases are not the subject of this article; however, it is worth mentioning that the question/answer similarity metric adopted during answer extraction is a weighted combination of lexical, syntactic, and semantic criteria, as thoroughly explained in (Quarteroni and Manandhar, 2009). YourQA's performance on non-factoids is reported in (Moschitti *et al.*, 2007).

Answer format. YourQA's answer format requires additional details as this is a key feature in the perspective of personalization. Returned answers are in the form of sentences where relevant words (or phrases) are highlighted; in case the expected answer type is a factoid (i.e. a person, location, organization, temporal expression or numerical expression), answers undergo an additional class-specific processing; for instance, regular expressions are applied to identify temporal locutions, while a Named Entity tagger is used for locations. This allows to highlight not only relevant query keywords

2. Google, `www.google.com`.

in the answer passage but also the "exact" answer words/phrases located within the answer sentence, as visible in figure 1.

In addition, the original passage containing the answer sentence is also returned: such a format, less concise than in typical QA systems, responds to the aim of providing a context to the exact answer and is convenient for returning non-factoid answers, such as definitions; moreover, the longer passage format is a motivation for including textual readability among the components of personalization, as explained further. A ranked list of YourQA answer paragraphs is illustrated in figure 2.

1. Title: GradeSaver: ClassicNote: About Pride and Prejudice, **URL:** http://www.gradesaver.com/classicnotes/titles/ **Google Rank:** 6, **file:** about.html

About Pride and Prejudice.

Pride and Prejudice, published in 1813, is Jane's Austen's earliest work, and in some senses also one of her most mature works.

Austen began writing the novel in 1796 at the age of twenty-one, under the title First Impressions.

Figure 1. *YourQA's top answer to: "When was 'Pride and Prejudice' published?"*

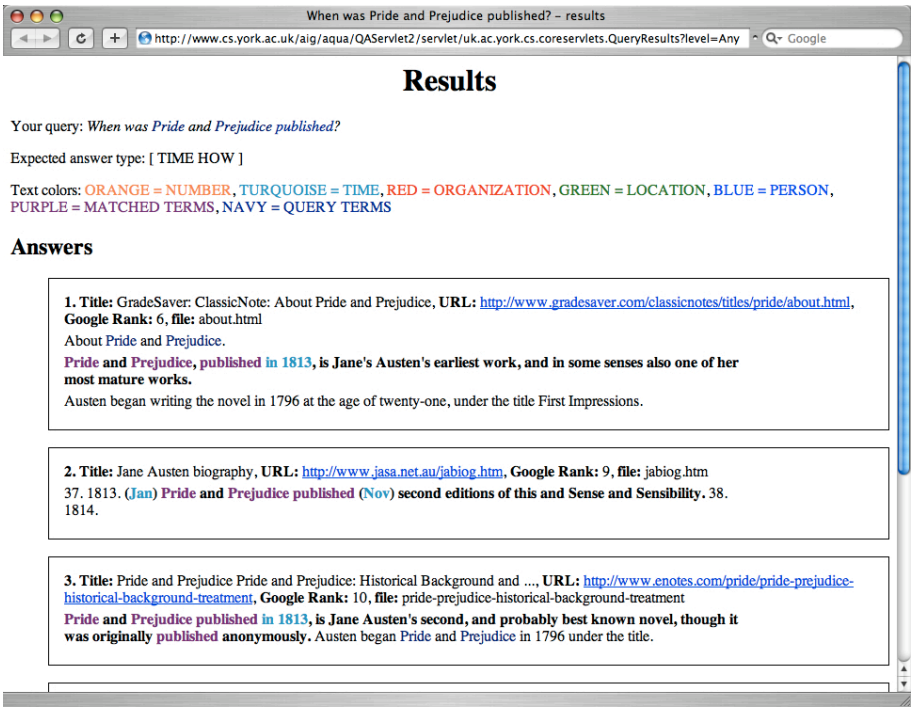


Figure 2. *YourQA's list of answers for: "When was 'Pride and Prejudice' published?"*

4. A User Model for open-domain Question Answering

In this work, the salient feature of a personalized QA system with respect to a traditional one (see section 3) is the presence of a User Modelling component, constructing, maintaining and updating a representation of the current user. User Model design is a complex task that inherently depends on the type of concrete application sought and for which a substantial body of research in the Question Answering case is missing. Here, the aim is to achieve personalization in open-domain Web QA, where closed-domain assumptions about users cannot be deployed due to the absence of an explicit domain knowledge representation.

To meet this aim, the proposed User Model (UM) is centered on two types of information: on the one hand, an estimation of the user's age and reading level; on the other, a representation of his/her topics of interest. The objective of such dimensions are an increased readability of answers and a profile-based answer filtering, respectively. Although it can be argued that personalization has many facets, we believe that the latter are important objectives toward the tailoring of answers in terms of both contents and language. Moreover, they are affordable in the open domain, where natural language applications generally struggle to apply knowledge-intensive data representations.

As an application scenario that would be generic enough as a proof-of-concept of personalized open-domain Question Answering and at the same time allow for a concrete, task-oriented application of User Modelling, an education scenario was designed for YourQA. Here, the User Model represents students searching for information on the Web for their assignments and consists of three components:

1) age range, $a \in \{7-10, 11-16, adult\}$; the first two ranges correspond to the primary and secondary school age in Britain, respectively;

2) reading level, $r \in \{basic, medium, advanced\}$;

3) profile, p, a set of textual documents, bookmarks and Web pages of interest.

Analogous UM components can be found in news recommender systems such as SeAn (Ardissono *et al.*, 2001) and SiteIF (Magnini and Strapparava, 2001), where age and browsing history, respectively, are part of the model. Moreover, constructing User Models based on the user's documents and Web pages of interest is frequent in personalized search systems (Teevan *et al.*, 2005).

The age range/reading level attribute is used to modify the presentation aspect of the QA system; during the document retrieval phase, an estimation of the suitability of documents to the user's age and reading abilities is used to filter out undesired documents (see section 5.1). Although the reading level can be modelled separately from the age range, e.g. an adult may have a basic reading level, we assume for simplicity in this paper that the values of these two dimensions are paired and therefore work with the reading level only. The profile attribute is applied during answer extraction in order to select answers from documents having topics in common with the topics extracted from the set of documents in the user profile, as explained in section 5.2.

4.1. *Reading level component*

Prior to this work, a number of natural language applications have been designed to address the needs of users with differing reading skills. In most cases, the computational approach to textual adaptation is based on natural language generation: the process "translates" a difficult text into a syntactically and lexically simpler version. In the PSET system mentioned in Section 2, newspaper text was reformulated using a tagger, a morphological analyzer/generator, and a parser (Carroll *et al.*, 1999); in SKILLSUM (Williams and Reiter, 2005), a set of choices regarding output (cue phrases, ordering, and punctuation) are taken, based on a set of rules to generate literacy test reports. The approach presented in this work is conceptually different: exploiting the wealth of information available via the Web, the QA system can afford to choose among the documents available on a given subject those which best suit the given readability requirements; such a choice derives from an accurate estimation of their reading level.

Reading level estimation of a document is a well-known problem in the education domain; the most widely used approaches to reading level estimation are models based on sentence length, such as "Flesch-Kincaid" (Kincaid *et al.*, 1975), Fry's model (Fry, 1969), or SMOG (McLaughlin, 1969). The key idea behind these approaches is that the readability of a text is inversely proportional to its length. In contrast, the approach adopted in YourQA is based on language modelling, a technique that accounts specifically for lexical information. Following the language modelling technique, reading level estimation is modelled as a multi-classification task where each class represents one reading level. In the YourQA case, the classes mirror the three different age ranges defined in the UM:

1) *basic*, representing a document suitable for ages 7 – 11;

2) *medium*, representing a document suitable for ages 11 – 16;

3) *advanced*, representing a document suitable for adults.

The reading level classifier requires labelled data to serve as training instances consisting of representative documents for each of the three classes. To this end, we collected a dataset of about 180 HTML documents, deriving from a number of Web portals where pages on a variety of topics are explicitly annotated by the publishers according to the above reading levels. The three Web document sets representing the 7–11, 11–16 and adult age ranges contain between 33K and 35K words after converting the HTML format to plain text. They contain documents randomly collected from Web portals such as BBC education (`bbc.co.uk/schools`), suitable for all age ranges, or NASA for kids (`kids.msfc.nasa.gov`), suitable especially for the 7–11 and 11–16 ranges. Readability judgments available directly from the Web portals are the reference for learning reading level classification; the fact that training instances are labelled by an external and trusted source contributes to the objectivity and soundness of the approach.

The adopted learning algorithm is the Smoothed Unigram Model, a variation of a Multinomial Bayes classifier based on unigram language modelling. This was proven in (Collins-Thompson and Callan, 2004) to be at least as effective as the "Flesch-Kincaid" length-based approach in estimating the reading level of subjects in primary and secondary school age. Indeed, the unigram method was preferred to the latter technique following the intuition that the discriminative power of length-based metrics might be less effective on Web documents, where the difference in length between complex documents and simple ones is often not as wide as for the printed text.

Reading level estimation proceeds in a training and a test phase. In the training phase, given a set of documents S representing a given reading level, the corresponding unigram language model is created. A unigram language model represents a set of documents as the vector of all the words appearing in the component documents associated with their corresponding probabilities of occurrence within the set. For generality, and to account for data sparseness, in our approach, words are replaced by their stems obtained by applying the Porter stemmer (Porter, 1980).

Once a number of language models lm_i are available for different reading levels, in the testing phase, where the reading level of a previously unseen document D must be estimated, a unigram language model is built to represent D (as done during the training phase). The estimated reading level of D is the one corresponding to the language model lm_i, maximizing the likelihood $L(lm_i|D)$ that D has been generated by lm_i. The likelihood is estimated as:

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

where w is in turn each word appearing within D, $C(w, D)$ represents the number of occurrences of w in D, and $P(w|lm_i)$ is the probability of w occurring in lm_i (approximated by occurrence frequency).

4.2. *Profile component*

In the current model of personalization, the user's interests are estimated based on the profile component of the User Model, defined as a set of both textual documents and Web pages of interest. Information extraction from user documents such as desktop files as a means of representation of their interests is a well-established technique for personalized IR. For instance, (Teevan *et al.*, 2005) experiment by indexing various amounts of user data to create a personalized search model, while in the Outride system (Pitkow *et al.*, 2002) a browser plugin accesses links and bookmarks building a model of the user's browsing preferences. In YourQA, both the collected textual documents and Web documents are used to estimate the user's interests.

4.2.1. *Profile estimation*

Profile estimation in YourQA is based on key-phrase extraction, a technique previously employed in several natural language tasks, including topic search, document

clustering, and summarization (Frank *et al.*, 1999; D'Avanzo *et al.*, 2004). Key-phrase extraction can be defined as a classification task where the aim is to extract the most important words or phrases to represent the semantics of a given text. Unlike text categorization, where a fixed set of domain-specific key-phrases must be assigned by the classifier, key-phrase extraction does not require them to be known in advance, proving a suitable technique for open-domain applications.

While this appears to be an innovative technique for User Modelling, there is evidence from previous work of the use of alternative content-based techniques for User Model creation. For instance, in (Magnini and Strapparava, 2001), documents are processed and relevant WordNet synsets (Miller, 1995) are extracted to predict potentially interesting documents based on such synsets. Our approach is lighter as it does not need to access an external lexical database such as WordNet, with a variable (at times inexistent) coverage of different topics. Moreover, open-domain QA may involve highly ambiguous data, complicating the computation of relevant "senses".

The adopted key-phrase extractor is Kea (Witten *et al.*, 1999), a domain independent extractor shown to be very robust across different document sizes and domains in a comparative study (Frank *et al.*, 1999): dealing with Web pages, that can have different lengths and structures, the above is a definite advantage.

Kea first splits each document D in a set of documents S into phrases, taking short subsequences of such initial phrases as candidates. Then, for each candidate phrase ϕ and each document D, two criteria determine whether ϕ is a key-phrase with respect to D in S: O, the index of ϕ's first occurrence in D, and T, the $TF \times IDF$ score [3] obtained by ϕ with respect to D in S. Following Naïve Bayes, T and O are assumed independent; hence, the probability that ϕ is a key-phrase for D is:

$$P(key_D^\phi|T,O) = \frac{P(T|key_D^\phi) \cdot P(O|key_D^\phi) \cdot P(key_D^\phi)}{P(T,O)} \quad [2]$$

where $P(T|key_D^\phi)$ is the probability that ϕ has $TF \times IDF$ score T assuming that ϕ is a key-phrase for D, and $P(O|key_D^\phi)$ is the probability that ϕ has index O assuming that ϕ is a key-phrase for D; finally, $P(key_D^\phi)$ is the a priori probability that ϕ is a key-phrase for D, and $P(T,O)$ is a normalization factor. The latter probabilities are estimated by the frequency of the corresponding event in the training data (Frank *et al.*, 1999). Based on [2], Kea outputs for each document D in the set a ranked list where candidate phrases are in decreasing order of relevance. The top k phrases are selected as document key-phrases for document D [4].

3. $TF \times IDF$ score is a measure of salience of term contained in a document within a given collection. The $TF \times IDF$ of a term t in document D belonging to collection S is measured as: $TF \times IDF(t, D, S) = P(t \in D) \times -log[P(t \in [S/D])]$.

4. After experimenting with several values, we fixed $k = 6$ in YourQA.

4.2.2. *Profile representation*

The internal representation of a profile document set reflecting an individual user is in the form of a two-dimensional key-phrase array, P. Each row of P corresponds to a profile document, and each column is associated with the rank of the corresponding key-phrase in the list of key-phrases output by Kea. As an illustrative example, a basic user profile, created from a document about Italian cuisine and one about the movie "Ginger and Fred", might result in the following array:

$$P = \begin{bmatrix} pizza & lasagne & baking & recipe & chef & eggs \\ fred & ginger & dancing & musical & movie & review \end{bmatrix}. \quad [3]$$

A further treatment on the outcome of key-phrase extraction is the stemming of key-phrases, carried out via the Porter Stemmer (Porter, 1980). The P array becomes:

$$P' = \begin{bmatrix} pizza & lasagne & \textbf{bak} & \textbf{recip} & chef & \textbf{egg} \\ fred & ginger & \textbf{danc} & \textbf{music} & \textbf{movi} & review \end{bmatrix}. \quad [4]$$

The user's profile is the basis for all the subsequent QA activity: the profile key-phrases will be compared against key-phrases extracted from the documents obtained during the document retrieval phase, optionally re-ranking the final answer list. Profile extraction and reading level filtering are the two core phases of the personalized QA algorithm illustrated in section 5.

5. Personalized Question Answering algorithm

The interaction between the User Modelling component and the core Question Answering component modifies the standard QA process exposed in section 3 at several stages, resulting in a new personalized algorithm. While question processing remains unchanged, the User Model affects both the document retrieval phase and the answer extraction phase. The resulting personalized QA algorithm is:

1) **Question Processing**:

   a) the query is classified and the two top expected answer types are estimated;

   b) the query is submitted to the underlying search engine;

2) **Document Retrieval**:

   a) the top n documents are retrieved from the underlying search engine;

   b) the reading levels of retrieved documents are estimated;

   c) documents whose reading level is incompatible with the user are discarded;

   d) key-phrases are extracted from the remaining documents;

   e) documents are split into sentences;

3) **Answer Extraction**:

a) document key-phrases are matched with the user profile key-phrases;

b) candidate answers are extracted from the documents and ordered by relevance to the query;

c) the degree of match between the topics from each candidate answer document and the user's profile is used as an additional relevance criterion and a new ranking is computed.

It may be worth highlighting that such a model of personalization affects the results to all types of questions, regardless of their expected answer classes. Thus, both factoid and non-factoid questions can receive personalized answers according to the proposed algorithm. The need to personalize answers to non-factoid questions may appear as the most intuitively justified, notably to account for ambiguous questions yielding answers about different domains. For instance, the IR engine will respond to the question: *What is "Ginger and Fred"?* with documents relating to a film, a building and a dancing couple. Again, acronyms such as "UM" can refer to several different entities (*University of Michigan*, *United Methodists*, *User Modelling*, etc) and the query *Where is the UM conference this year?* would thus yield several possible answers to choose from. However, personalization can also affect the factoid domain; for instance, it is intuitive that the answer to *When did the Middle Ages begin?* – clearly a temporal (i.e. factoid) question – can be different depending on the age and reading abilities of the reader. While a child might welcome the answer *The Middle Ages start with the fall of the Roman Empire in 476 AD*, an adult might prefer an answer highlighting how it makes little sense to think of a unique starting date to the medieval era.

The two phases of the QA algorithm affected by personalization, i.e. document retrieval and answer extraction, are discussed in sections 5.1 and 5.2.

5.1. *Document retrieval*

In the standard QA algorithm presented in section 3, document retrieval consists of retrieving the top search engine documents and splitting them into sentences. When the User Modelling component is active, two additional retrieval steps take place: reading level estimation and filtering, and key-phrase extraction.

5.1.1. *Reading level estimation and filtering*

Estimation of the reading level of each document returned by the IR engine in response to the query is conducted via the language modelling technique illustrated in section 4.2. The documents having an incompatible reading level with the user are discarded so that only those having the same estimated reading level as the latter are retained for further analysis. As there can be queries for which the number of retrieved documents matching the requested reading level is less than the number of documents returned by the system (currently five), this condition is relaxed so that some of the documents having different reading levels may be accepted as candidates. In all cases,

due to the absence of other criteria at this stage of the QA algorithm, the choice of which documents to retain for a given reading level is determined by the search engine rank of the former (a higher rank determines preference). The subsequent phase of answer extraction therefore begins with the documents remaining from the reading level filtering phase.

5.1.2. *Key-phrase extraction*

Once a working subset of the retrieved documents has been collected, key-phrases are extracted from the latter using the same approach as for the User Model profile (see section 4.2.1): Kea processes the set of retrieved documents to extract the top k key-phrases for each document. These are represented by the system as a two-dimensional array named $Retr$, similar to the P array created for the User Model profile (see the example in [3]). As an illustrative example, the $Retr$ array for the query: *What is "Ginger and Fred"?*[5] is reported in [5]. Each row represents a retrieved document, and column index represents a key-phrase rank; for instance, **movi** located in cell $(1, 1)$ is the first ranked key-phrase extracted from a document about Astaire and Roger's movies (notice that, also in this case, key-phrases are stemmed).

$$Retr = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \textbf{movi} & item & fred\_astaire & film & ginger\_rogers \\ build & resid & gehri & tower & project \\ fred & ginger & film & music & movi \\ film & fred & ginger & fellini & ginger\_and\_fred \\ gehri & build & citi & histor & destruct \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad [5]$$

5.2. *Answer extraction*

In the standard QA algorithm (Section 3), the primary passage ranking criterion is the similarity of the passage's central sentence to the question, and the secondary criterion is the IR rank of the document from which the passage has been extracted. The personalized QA algorithm applies an additional ranking criterion giving priority to answers from documents having common key-phrases with the user's profile documents.

5.2.1. *Relevance computation*

For each document composing the User Model profile set and the retrieved document set, a ranked list of key-phrases is available from the previous steps and represented as a row in the P and $Retr$ array, respectively (Section 5.1.2). Let us call $Retr_i$

5. As visible from the key-phrases, *Ginger and Fred* may refer to a famous dancing couple, the "dancing buildings" by architect F. Gehry, and a film directed by F. Fellini.

the document represented in the i-th row in $Retr$, and P_n the document represented in the n-th row of P [6]. Then we can compute how relevant $Retr_i$ is to P as follows. First, given the j-th key-phrase extracted from $Retr_i$, k_{ij}, and the n-th document in P, P_n, the *relevance* of k_{ij} with respect to P_n, named $w(k_{ij}, P_n)$, is defined as:

$$w(k_{ij}, P_n) = \begin{cases} \frac{|Retr_i|-j}{|Retr_i|}, & k_{ij} \in P_n \\ 0, & otherwise. \end{cases} \quad [6]$$

The total relevance of $Retr_i$ with respect to P, denoted by $w_P(Retr_i)$, is now defined as the maximal sum of the relevance of its key-phrases, obtained for all the rows in P:

$$w_P(Retr_i) = max_{n \in P} \sum_{k_{ij} \in Retr_i} w(k_{ij}, P_n). \quad [7]$$

Keeping the relevance computation separated across the single documents (rows) in the profile is a strategy to prevent errors and ambiguities. Without this precaution, a profile about programming and Greek islands might result in a high weight for a document about the *island* of Java. In contrast, following [7], a document about Java as a programming language would be a competitor in the ranking with a document about islands, as key-phrase matches from different documents add up to competing similarity scores.

The relevance formula [7] takes 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. More specifically, for each of the $|N|$ documents retrieved for a query, and for each term $t_i \in N$, the number of documents $\in N$ containing t_i, i.e. n_i, is computed. The relevance of term t_i with respect to the current user is then (Teevan *et al.*, 2005):

$$w(t_i) = log \frac{(r_i + 1/2)(N - n_i + 1/2)}{(n_i + 1/2)(R - r_i + 1/2)}, \quad [8]$$

where R is the number of documents for which relevance feedback has been provided (i.e. documents which have been indexed), and r_i is the number of documents which contain t_i among the R examined. We interpret R as the set of documents composing the User Model profile, while N evidently corresponds to the set of documents obtained by YourQA during document retrieval [7]. Moreover, instead of handling all the terms contained in the user's documents (which can be costly and introduce noise), we use the information deriving from key-phrase extraction and only analyze the terms contained in the key-phrase arrays P and $Retr$.

6. 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 (hence it does not determine a rank based on relevance to the question).

7. For conciseness, we omit the full proof of applicability of [8], reported in (Quarteroni, 2007).

5.2.2. *Final answer ranking*

Having computed a relevance score for each retrieved document, the personalized algorithm uses the following answer ranking criteria, ordered by priority:

1) similarity of the answer passage to the question;

2) relevance of the passage's source document with respect to the UM profile;

3) original IR rank of the source document, as a tie-breaking criterion.

The above ranking is introduced in order to prevent user adaptivity from reducing result objectivity. Indeed, the compatibility of a given document with respect to a given UM is always a secondary ranking criterion to the similarity to the query, hence profile match is only considered in case of a tie between candidate answers.

Let us consider the question: *What is "Ginger and Fred"?* In the standard version of YourQA, the most relevant retrieved document is the top Google hit, yielding the top answer: "***Ginger and Fred (Italian: Ginger e Fred) is a 1985 comedy/drama film directed by Federico Fellini and starring Marcello Mastroianni and Giulietta Masina.***" When a profile containing a sufficient number of key-phrases relating to architecture is used, the most relevant retrieved document becomes the sixth Google hit, yielding the top answer: "***It is a 'dancing building' and was named 'Ginger & Fred' in an allusion to the American film icons***." This happens because, since both answer candidates obtain the same similarity score with the question (both contain the words *Ginger* and *Fred*, and in both cases such words are separated by one word, *and* resp. *&*), the tie-breaking criterion based on profile relevance can be applied.

6. A Web-based user interface

In order to evaluate the personalized QA algorithm and enable a comparison of the standard version of YourQA to its interactive variant, a Web-based prototype has been implemented. When accessing the YourQA Web prototype, the user has three options:

A) create a new profile from documents of interest and/or browser bookmarks; in this case, key-phrase extraction is used to obtain a list of key-phrases from the text documents or bookmarked Web pages; providing documents of interest is a way to solve the cold start problem of creating a profile from a previously unseen user; indeed, eliciting Web pages of interest or mining the user's bookmarks folder appears to be a fairly effortless process to collect initial information, in contrast to defining a complete profile, for instance by requiring the user to fill in a form as in (Ardissono *et al.*, 2001).

B) load a previously saved profile; in this case, the list of key-phrases contained in the loaded user profile are obtained;

C) decide not to use a profile; in this case, no key-phrases are extracted.

In cases A) and B), the user is shown key-phrases corresponding to his/her profile to enable the exclusion of unsuitable or incorrect ones (see figure 3). The profile resulting from the remaining key-phrases is the base for the subsequent QA activity:

any question the user will submit to YourQA will be answered by taking such a profile into account. The user can click on the "Save as..." button (see figure 3) to save a newly created profile or the current updates (i.e. selected/deselected key-phrases) and reload the profile in the future. Enabling the user to modify and save a profile, in addition to the implicit updates consisting in the user's evolving bookmarks and documents of interest, makes the UM component dynamic.

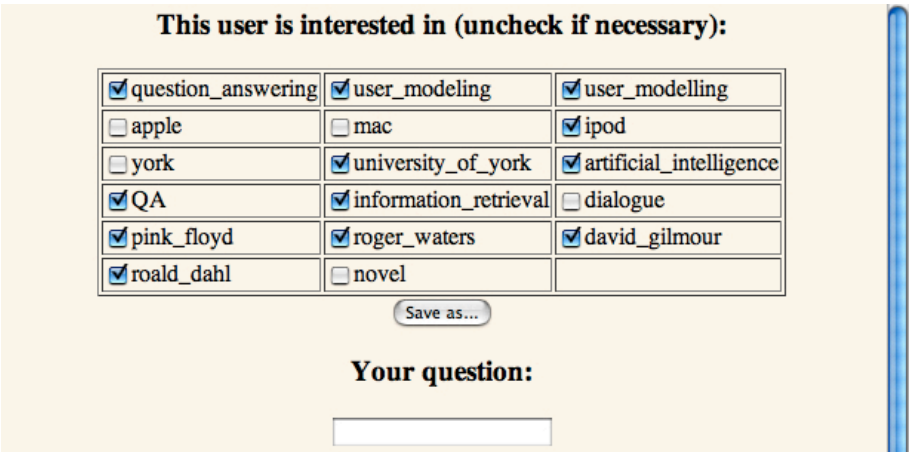


Figure 3. *Using the YourQA prototype (screenshot detail): modifying and saving the profile, issuing a query.*

Once a profile has been chosen, the actual Question Answering session can start, with the user entering a question in the dedicated text field. By default, the personalized prototype developed for YourQA performs no filtering based on reading levels. However, the user has the option to activate the filtering based on one of the reading levels specified in the User Model (i.e. basic, medium or advanced). An illustration of the query submission phase is reported in figure 3. The core Question Answering session continues as exposed in section 5; the result format is the same as in the baseline QA case, described in section 3 and depicted in figure 1.

The prototype described in this section has been used to perform a thorough evaluation of both reading level-based and profile-based evaluation according to the evaluation methodology designed in section 7.

7. Evaluation

The evaluation of personalized Question Answering was designed to separately assess the contributions of the reading level and of the profile attribute of the User Model. The motivation for this choice was, on the one side, to obtain a qualitative measure of each of the User Model components, so that these may be used separately

for the purpose of different applications; on the other side, this evaluation strategy minimizes the introduction of biases and interaction effects. sections 7.1 and 7.2 focus on the evaluation methodology for the reading level and the profile attribute of the User Model, respectively.

7.1. *Reading level evaluation*

The evaluation of reading level estimation was conducted according to two criteria: the first was an objective assessment of the robustness of the unigram language models created to represent the reading level; the second was user-centered and assessed the agreement of users with the system's estimation.

The robustness of unigram language models was computed by randomly splitting all the documents used to create the language models into ten different bins of the same size. Then, a ten-fold cross-validation classification experiment was run, where the ratio of correctly classified documents with respect to the total number of documents was computed for each bin, regardless of the reading level and used as accuracy for the bin. The average bin accuracy was 94.2% with a small standard deviation (2.0), proving the estimation method to be efficient and robust.

As robustness does not prove a direct effect on the user's perception of reading levels, an additional metric was introduced to assess the users' agreement with the system's reading level estimation: *Reading level precision (P_l)*. Given the set $\mathcal{L}$ of documents classified by the QA system as suitable for a reading level l, P_l is the ratio between $suitable(\mathcal{L})$, i.e. the number of documents $\in \mathcal{L}$ also rated by the users as suitable for l, and the size of $\mathcal{L}$: $P_l = \frac{suitable(\mathcal{L})}{|\mathcal{L}|}$.

An experiment was designed with 20 participants aged between 16 and 52, from various backgrounds and mother-tongues and with a self-assessed good or medium English reading level. The answers to such questions included factoids (such as *Who painted the Sistine Chapel?*), lists (*Types of rhyme*), and definitions (*What is chicken-pox?*). Each participant examined the results returned by YourQA to 8 of the 24 questions; for each question, results were returned in three different answer groups, corresponding to the basic, medium and advanced reading levels. Participants then evaluated the three sets of answers and specified for each answer passage whether or not they agreed that the given passage was assigned to the correct reading level.

The resulting agreement scores showed that, altogether, evaluators found reading level estimation accurate: while answers labeled as advanced obtained a reading level agreement of 94%, medium answers achieved 85% and basic ones obtained 72%. Such figures can be intuitively explained by the fact that it is more constraining to conform to a lower reading level than to a higher one.

With the current study on readability, we hope to initiate what we advocate to be a deeper future research on the potentials of reading level filtering. In particular, a thorough study of the impact of this technique, not only on objective and subjective

accuracy but also on its usefulness in the QA process, remains to be carried out. Additionally, studies where reading level and age are clearly separated, e.g. modelling adult users with low reading skills, would certainly prove the need for a separate representation of such two aspects of the User Models.

7.2. *Profile evaluation*

In designing an evaluation method for the profile component of the User Model, the aim was to assess whether user-adaptive answer filtering would be positive in terms of answer usefulness and, in any case, whether it would be perceived at all. The impact of the UM profile was tested by taking as a baseline the standard version of YourQA where the User Modelling component is inactive. The experiment involved ten adult participants from different backgrounds and occupations and took place in three phases: first, user profiles were collaboratively created with the participants; then a personalized QA session took place, and finally users replied to a satisfaction questionnaire.

7.2.1. *First phase: profile design*

In the first phase, participants were invited to explore the Yahoo! Directory [8], an online guide to the Web where sites are categorized by topic, and provide 2-3 categories of their interest: these ranged from dog care and music to role-playing games and computers. Participants were also invited to brainstorm as many key-phrases as they wanted relating to each of their chosen categories: examples of these were "frank capa", "muzzle", "little mermaid"; individual profile arrays were created from the key-phrases using the format in formula [3]. Then, for each category of interest, related queries were elaborated in such a way that the system's answers would be different when the profile filtering component was active, and entered in the experimental query set.

The design of "artificial" queries, by which users' interaction with the system was controlled, was necessary to ensure that the final answer ranking would be affected by the use of the profile, thus allowing to measure any difference in feedback with respect to the "profile off" case. Several queries invoked definitions of terms or expressions that had different meanings depending on the context. For instance, the question: *What is boot camp?* could refer to both computers and the military domain. Another example was the question: *What is Apollo?* for which results about NASA missions would be ranked highest for a user interested in space exploration than for a user with no such interests.

8. `http://dir.yahoo.com`.

7.2.2. *Second phase: Question Answering*

Once profiles and queries had been elaborated, participants were assigned an instruction sheet with three of such queries to be submitted to YourQA. Each query was chosen from the previously compiled artificial query set, following specific criteria:

Q_A: related to one of the current user's interest categories and answered by loading the user's profile in the personalized version of the system. Q_A tests the abilities of the personalized system, hence it was chosen for each user so that the final list of answers would be affected by the User Modelling component;

Q_B: related to another of the current user's interest categories and answered using the baseline version of the system (i.e. *without* User Modelling). Q_B's role is to compare the baseline system to the personalized version – producing the answers to Q_A – under the same conditions, as both Q_A and Q_B are related to the user's profile (although chosen from different interest categories in order to avoid biases);

Q_C: chosen from the interest categories of a different user (such that no overlap exists with the current user's profile) and answered using the baseline version of the system. Note that, since by construction there is no key-phrase overlap between the documents retrieved for Q_C and the user's profile, by the definition [6], the answers are the same regardless of the system version used. Indeed, the role of Q_C is to detect whether answers to Q_B and Q_C are received differently although the system performs no personalization in both cases. Hence, Q_C is a control query to detect any bias in users towards questions related to their profile.

Moreover, since the same queries were used as Q_A and Q_C for different users, we could compare the feedback obtained for the same query when a matching user profile was active and when not. For example, the question *What is Apollo?* was selected to be Q_A for a user interested in space, and Q_C for a user who had not specified this interest. The satisfaction with the answers to the question as assessed by the former user was compared to the satisfaction of the latter user to verify whether personalization affects the perception of results.

The choice of three different domains for the queries Q_A, Q_B and Q_C is meant to minimize within-subject bias due to the presence of a previous question related to the same interest category, potentially yielding similar answers. An additional measure to prevent biases with their experience with previous queries was the fact that questions appeared on the task sheets in a random order: hence, for some users the question with role Q_A appeared as the first one, for some others as second or third.

As soon as users issued a query, the top five answers were computed in real time by the QA system; reading level filtering was inactive so that only the user's profile played a role in the experiment.

7.2.3. *Questionnaire*

As soon as the list of five answers was available for each of the three queries, users replied to the following questionnaire:

1) 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

2) For the query results taken *as a whole*:

TEST3: *Finding the information 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 Likert scale answers to the questionnaire provide a qualitative assessment of the effects of the UM, tested at user level to eliminate the nuisance introduced by cross-user evaluation (Chin, 2001). Each questionnaire item relates to a separate factor: first, TEST1 measures the perceived usefulness of each result, and corresponds to the user-centered precision metric applied in personalized IR (Shen *et al.*, 2005). TEST2 measures the perceived relatedness of answer content with respect to the user profile. Relatedness is not a success indicator *per se*, i.e. a result perceived as related to one's own interests need not imply more satisfaction; rather, the former is an indicator of the user's awareness of the system's intent to personalize answers. Thirdly, TEST3 measures user satisfaction with respect to the time taken browsing results. Measuring perceived rather than actual time spent looking for answers is considered to better correlate to user satisfaction in several studies, e.g. (Walker *et al.*, 2000). Finally, TEST4 measures the perception of sensitivity of the system to the profile when answering the query overall, i.e. regardless of individual answers as in TEST2. Again, rather than a success metric, sensitivity is an indicator of awareness of the user regarding the impact of personalization in the system as a whole.

7.2.4. *Results*

Table 1. *Profile evaluation: summary of results (average ± standard deviation). For TEST1 and TEST2, the average of the scores given to the top 5 answers is reported.*

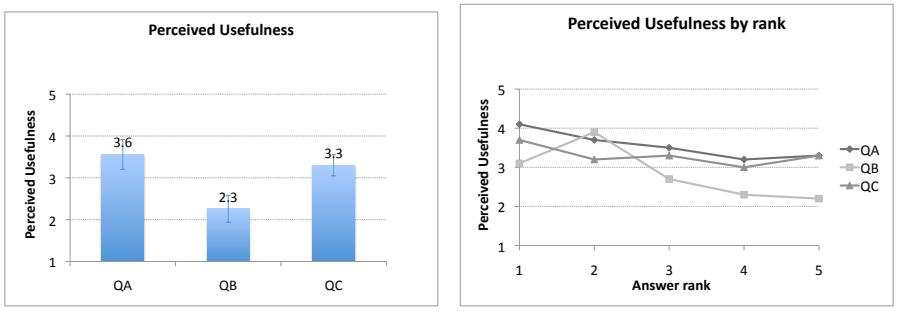
Measurement	Questionnaire item	Q_A	Q_B	Q_C
Perceived usefulness	TEST1	3.6 ± 0.4	2.3 ± 0.3	3.3 ± 0.3
Perceived relatedness	TEST2	4.0 ± 0.5	2.2 ± 0.3	1.7 ± 0.1
Perceived time	TEST3	3.1 ± 1.1	2.7 ± 1.3	3.4 ± 1.4
Perceived sensitivity	TEST4	3.9 ± 0.7	2.5 ± 1.1	1.8 ± 1.2

Answer usefulness (TEST1). The first row of table 1 reports the average and standard deviation of the perceived answer usefulness for each query (answers to TEST1). As depicted in figure 4(a), there is a visible difference between the perceived usefulness of answers to Q_A with respect to those to Q_B. To better evaluate the difference between the three conditions, we carried out a one-way within-subjects analysis of variance (ANOVA) with the average usefulness assessed by users as factor and Q_A, Q_B and Q_C as levels. The $F-$test allowed to reject the null hypothesis by which there is no difference between the three levels when it comes to perceived usefulness ($F = 21.57$, $d.f. = 2, 18$, $p < 0.0001$).

To further investigate our findings, we performed the paired t-test on the usefulness judgments for the two different questions related to the user's profile, i.e. Q_A (answered by taking the profile into account) and Q_B (answered by ignoring the profile). The test revealed a significant difference between Q_A and Q_B ($p = 0.02$). This highlights that users consider the answers to questions related to their profile more useful when such profile is taken into account in the answer extraction process. Furthermore, the t-test comparing the average answer usefulness for the two cases where profile filtering was inactive, corresponding to queries Q_B (related to the user's profile) and Q_C (unrelated to the user's profile), returned another significant p-value (0.03). This suggests that with a baseline system (i.e., no personalization), users tend to find answers to generic questions more useful than answers to questions related to their own profile. Perhaps, users are less "demanding" as far as answers concern an unfamiliar domain. However, if we perform a Bonferroni adjustment of α to account for the fact that we made two comparisons (i.e. $Q_A - Q_B$ and $Q_B - Q_C$), we note that the above p-value of 0.03 is just above $\frac{\alpha}{2} = 0.025$, denoting a weak significance.

An interesting observation is suggested by plotting answer usefulness as a function of rank (figure 4(b)): when answers are produced by taking user profile into account (question Q_A), the top answer is rated as highly useful by users, while usefulness gradually decreases with answer rank; this trend does not manifest for the answers produced by ignoring the profile (Q_B and Q_C). This suggests that indeed not only the answers to Q_A are regarded as more useful *on average* by the users, but also that the system is able to rank useful answers in a higher position. However, the Friedman test performed on ranked values did not highlight a significant difference, hence from the collected data it cannot be concluded that the ranking of personalized answers differs from standard ranking. Such a finding could partly be due to the "accidental" presence of a number of useful replies in the second rank of Q_C's answers in the specific experiment we conducted.

Answer relatedness (TEST2). To analyze the answers to TEST2, which assessed the perceived relatedness of answer to the current profile, we computed the ANOVA table on the data summarized in the second row of Table 1 and visualized in figure 5(a). Again, we used Q_A, Q_B and Q_C as levels, this time with answer relatedness as the independent variable. The results showed an overall significant difference ($F = 45.14$, $d.f. = 2, 18$, $p < .0001$), confirming that the answers to the three queries do not share the same relatedness distributions. Furthermore, to assess whether answers to profile-



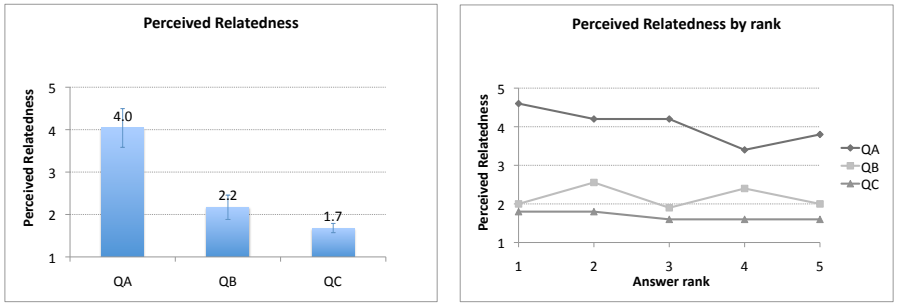
(a) average of the top five answers (b) average for each answer rank

Figure 4. *Profile evaluation: perceived system usefulness to profile.*

related questions (Q_A and Q_B) obtained by ignoring the users' profile (Q_B case) were perceived as significantly less related to their interests than those obtained using their own profile (Q_A case), we computed the paired t-test between Q_A and Q_B, which returned a significant difference ($p < 0.01$). As expected, the paired t-test between Q_A and Q_C revealed an even lower $p-$value (< 0.001) as indeed in the Q_C case both question and answers have nothing to do with the user's profile. In contrast, the difference between Q_B and Q_C was not significant according to the t-test, as indeed in both cases the User Model profile played no role in the selection of results.

Similarly to the case of TEST1, the average perceived relatedness of the results to Q_A was high for the top answer and slightly decreased with the rank; in contrast, for the other related question Q_B, where the profile was ignored, relatedness was lower and irregular; finally, for the unrelated question Q_C, answer relatedness was lowest and "stable" across ranks (see figure 5(b)). However, the difference was again not significant according to the Friedman test; this suggests that, since the personalized version of YourQA gives priority to the relevance of answers with respect to *queries* and uses relevance with respect to user profiles as its secondary criterion, the higher average relevance of answers obtained using the profile is not sufficiently fine-grained to be measured across the ranks.

Time spent looking for answers (TEST3). The average time perceived by users as necessary to locate suitable answers is plotted in figure 6(a). It can be noted that, for questions related to the users' profile, the time required to locate answers was perceived to be slightly shorter when the personalization component was active: Q_A obtained an average of 3.1 ± 1.1, while Q_B obtained 2.7 ± 1.3. However, for question Q_C, which was unrelated to their profile, users found the time of interaction even shorter, which might suggest that they spent less time investigating results to questions they considered uninteresting. Indeed, the $F-$test executed by using the ANOVA table



(a) average of the top five answers (b) average for each answer rank

Figure 5. *Profile evaluation: perceived system relatedness to profile.*

results rejects the null hypothesis corresponding to "no difference in browsing time" ($F = 6.73$, $df = 2, 16$ [9], $p = 0.008$).

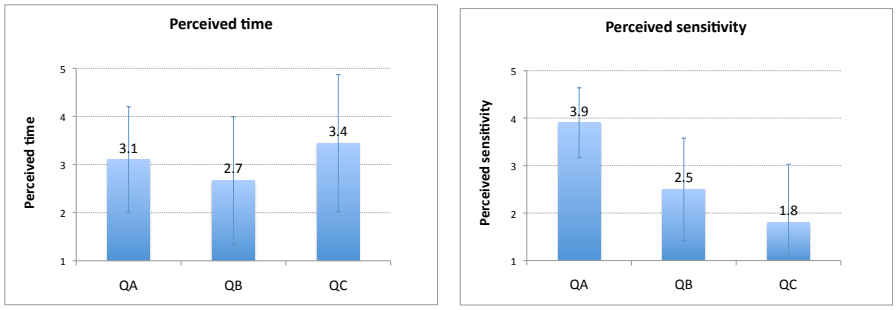
However, by taking a closer look at the results, the only significant difference is the one between Q_B and Q_C, indicating that when the profile is inactive during answer extraction, users take more time browsing results to questions related to their profile (Q_B) than those relating to some other user's profile (Q_C); this may suggest that the user is more interested in what the system returns in the former case. Interestingly, the shorter browsing time of profile-related query results when the profile is ignored (Q_B) is not significantly different from the case when it is used (Q_A) and this result may have occurred by "accident". In summary, we believe that the time question is problematic as the perception of time is influenced by a variety of factors, such as familiarity with the system and personal interest towards the results; for this reason, we reckon that these findings do not provide a sufficiently clear picture about personalization efficiency.

Profile sensitivity (TEST4). To interpret the results of TEST4, it must be noted that each user's task sheet clearly required them to assume during the experiments that their interests were exclusively the ones specified during the initial step of topic elicitation. This was to reduce the occurrence of biases from other interests that users may not have specified in advance. The results for perceived sensitivity computed for each query, plotted in figure 6(b), show a difference between the answers to question Q_A (3.9 ± 0.7) and those to question Q_B (2.5 ± 1.1) and Q_C (1.8 ± 1.2): such a difference is significant according to the ANOVA table ($F = 40.09$, $d.f. = 2, 18$, $p < 0.0001$).

Furthermore, the paired t-test between Q_A and Q_B underlines a significant difference ($p = 0.01$), denoting that users do perceive the higher sensitivity of results tailored to their own needs. The difference between Q_A and Q_C is even more significant ($p < 0.001$): this was expected as in the case of Q_C, not only the user profile is

9. Only 9 of the 10 subjects provided feedback to TEST3 for all three questions.

inactive but the question itself is unrelated to the profile. Finally, there is no significant difference between Q_B and Q_C: indeed, both queries are addressed by ignoring the user profile, making such a finding also expectable.



(a) perceived browsing time (b) perceived sensitivity to profile

Figure 6. *Profile evaluation: perceived time and sensitivity (average of top 5 answers).*

The results of TEST4 complete the outcome of TEST1 and TEST2, illustrating that not only users found personalized answers generally more related to their profile and more useful than "standard" answers, but also that they felt aware of the system's adaptivity to their profile.

8. Conclusions

We present an efficient and light-weight method to personalize the results of a Web-based Question Answering system based on a User Model representing the reading level and interests of individual users. We show how the User Model components can be estimated automatically and fairly unobtrusively from the user's documents and how they can be used to filter and re-rank the answers to their queries.

Moreover, we introduce a user-centered evaluation methodology for personalized Question Answering that independently assesses the effects of the two main User Model components: reading level and personal interests. The results of our experiments show, on the one hand, the accuracy of the language modelling techniques used for reading level estimation; on the other, a statistically significant improvement is measured when filtering answers based on the users' profile in terms of both perceived answer usefulness and profile relatedness.

In conclusion, although the application of User Modelling for personalizing a Web system is not new, this work describes to our knowledge the first fully implemented and evaluated application of such a technique to Question Answering. The proposed approach is by far not a final solution to the problem of personalizing a natural language application. In fact, we argue that significant progress would be brought by the

ability to dynamically update and refine User Models based on previous information-seeking history.

A first step towards a more dynamic User Model is the analysis of the interaction logs obtained by the system, upon which key-phrase extraction may be performed to update the user's interests. Indeed, previous work highlights the contribution of dialogue interfaces to the construction of User Models in the field of intelligent tutoring: for instance, in the collaborative learning environment described in (Linton *et al.*, 2003), a chat tool is used for keyword extraction and to infer the evolving level of knowledge of each student. Similarly, it seems worthwhile to explore interaction logs of the conversational version of YourQA (Quarteroni and Manandhar, 2009) to obtain the user's reading level and interests.

Acknowledgements

Part of this work has been supported by the University of York under the supervision of Suresh Manandhar. The author would like to thank Helen Petrie for her valuable suggestions in the design of profile evaluation. Last but not least, the attentive and patient work of anonymous reviewers greatly improved the quality of this work and should be acknowledged.

9. References

Alfonseca E., Rodriguez P., "Modelling Users' Interests and Needs for an Adaptive Online Information System", *Proceedings of User Modelling*, vol. 2702 of *LNAI/LNCS*, Springer, 2003.

Ardissono L., Console L., Torre I., "An adaptive system for the personalized access to news", *AI Communications*, vol. 14, n° 3, p. 129-147, 2001.

Belkin N. J., Croft W., "Information Filtering and Information Retrieval: Two Sides of the Same Coin?", *Communications of the ACM*, vol. 35, n° 12, p. 29-38, 1992.

Carroll J., Minnen G., Pearce D., Canning Y., Devlin S., Tait J., "Simplifying text for language-impaired readers", *Proceedings of EACL*, p. 269-270, 1999.

Chin D. N., "Empirical Evaluation of User Models and User-Adapted Systems", *User Modeling and User-Adapted Interaction*, vol. 11, n° 1-2, p. 181-194, 2001.

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

D'Avanzo E., Magnini B., Vallin A., "Keyphrase Extraction for Summarization Purposes: The LAKE System at DUC-2004", *Proceedings of the Document Understanding Conference (DUC)*, Boston, MT, USA, 2004.

Frank E., Paynter G. W., Witten I. H., Gutwin C., Nevill-Manning C. G., "Domain-Specific Keyphrase Extraction", *Proceedings of IJCAI*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 668-673, 1999.

Fry E., "A readability formula that saves time", *Journal of Reading*, vol. 11, n° 7, p. 265-71, 1969.

Inui K., Fujita A., Takahashi T., Iida R., Iwakura T., "Text simplification for reading assistance: a project note", *Proceedings of the ACL 2003 Workshop on Paraphrasing: Paraphrase Acquisition and Applications*, p. 9-16, 2003.

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

Kobsa A., "Generic User Modeling Systems", *User Modeling and User-Adapted Interaction*, vol. 11, p. 49-63, 2001.

Kwok C. T., Etzioni O., Weld D. S., "Scaling question answering to the Web", *Proceedings of WWW*, 2001.

Linton F., Goodman B., Gaimar R., Zarrella J., Ross H., "Student Modeling for an Intelligent Agent in a Collaborative Learning Environment", *Proceedings of User Modelling*, 2003.

Liu Y., Agichtein E., "You've got answers: towards personalized models for predicting success in community question answering", *Proceeding of HLT '08*, Association for Computational Linguistics, Morristown, NJ, USA, p. 97-100, 2008.

Magnini B., Strapparava C., "Improving User Modelling with Content-Based Techniques", *Proceedings of User Modelling*, vol. 2109 of *LNCS*, Springer, 2001.

Mairesse F., Walker M., "Trainable Generation of Big-Five Personality Styles through Data-driven Parameter Estimation", *Proceedings of ACL*, Columbus, OH, USA, 2008.

McLaughlin G., "SMOG grading: A new readability formula", *Journal of Reading*, vol. 12, n° 8, p. 693-46, 1969.

Miller B., Albert I., Lam S., Konstan J., Riedl J., "MovieLens Unplugged: Experiences with a Recommender System on Four Mobile Devices", *Proceedings of HCI*, 2003.

Miller G. A., "WordNet: a lexical database for English", *Communications of the ACM*, vol. 38, n° 11, p. 39-41, 1995.

Moschitti A., Quarteroni S., Basili R., Manandhar S., "Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification", *Proceedings of ACL*, Prague, Czech Republic, 2007.

Paik W., Yilmazel S., Brown E., Poulin M., Dubon S., Amice C., "Applying natural language processing (NLP) based metadata extraction to automatically acquire user preferences", *Proceedings of K-CAP '01*, ACM, New York, NY, USA, p. 116-122, 2001.

Pitkow J., Schuetze H., Cass T., Cooley R., Turnbull D., Edmonds A., Adar E., Breuel T., "Personalized search", *Communications of the ACM*, vol. 45, n° 9, p. 50-55, 2002.

Porter M., "An algorithm for suffix stripping.", *Program*, vol. 14, n° 3, p. 130-137, 1980.

Quarteroni S., Advanced Techniques for Personalized, Interactive Question Answering, PhD thesis, The University of York, 2007.

Quarteroni S., Manandhar S., "Designing an Open-Domain Interactive Question Answering System", *Natural Language Engineering*, vol. 15, n° 1, p. 73-95, 2009.

Shen X., Tan B., Zhai C., "Implicit user modeling for personalized search", *Proceedings of CIKM*, ACM Press, New York, NY, USA, p. 824-831, 2005.

Simmons R. F., "Answering English questions by computer: a survey", *Communications of the ACM*, vol. 8, n° 1, p. 53-70, 1965.

Stock O., "ALFRESCO: Enjoying The Combination of NLP and Hypermedia for Information Exploration", *Proceedings of the AAAI Workshop on Intelligent Multimedia Interfaces*, p. 197-224, 1993.

Teevan J., Dumais S. T., Horvitz E., "Personalizing search via automated analysis of interests and activities", *Proceedings of SIGIR*, ACM Press, New York, NY, USA, p. 449-456, 2005.

Thai V., O'Riain S., Davis B., O'Sullivan D., "Personalized Question Answering: A Use Case for Business Analysis", 2006.

Virvou M., Moundridou M., "Student and Instructor Models: Two Kinds of User Model and Their Interaction in an ITS Authoring Tool", *Proceedings of User Modelling*, vol. 2109 of *LNCS*, Springer, 2001.

Voorhees E. M., "Overview of the TREC 2003 Question Answering Track", *Proceedings of TREC*, 2003.

Walker M. A., Kamm C., Litman D., "Towards Developing General Models of Usability with PARADISE", *Natural Language Engineering Special Issue on Best Practice in Spoken Dialogue Systems*, 2000.

Williams S., Reiter E., "Generating Readable Texts for Readers with Low Basic Skills", *Proceedings of ENLG-2005*, p. 140-147, 2005.

Witten I. H., Paynter G. W., Frank E., Gutwin C., Nevill-Manning C. G., "KEA: Practical Automatic Keyphrase Extraction", *Proceedings of Digital Libraries*, p. 254-255, 1999.

Zhang Y., Koren J., "Efficient bayesian hierarchical user modeling for recommendation system", *Proceedings of SIGIR'07*, ACM, New York, NY, USA, p. 47-54, 2007.