Free-Text Answer Analysis on the Web

Bertrand IBRAHIM, Stephen D. FRANKLIN, Bernard LEVRAT

Résumé. Le World-Wide Web est déjà largement utilisé à des fins éducatives. Cette utilisation est influencée par des facteurs techniques, éducatifs et sociaux, y compris les attentes et les connaissances des éducateurs. Alors que beaucoup d'auteurs de matériel éducatif se contentent d'utiliser les possibilités de transfert à distance du Web, d'autres considèrent que l'éducation nécessite plus d'interaction et désirent créer et présenter du matériel interactif sur le Web. En plus des outils pour simplifier autant que possible la préparation de ce genre de matériel, il est nécessaire de développer une vision des possibilités techniques du Web qui se concentre sur les aspects éducatifs. Cet article énumère d'abord les techniques actuellement utilisées pour l'évaluation de l'apprenant, puis présente une méthode qui permet aux éducateurs d'intégrer facilement de l'analyse de réponse en texte libre dans leurs documents WWW. Diverses possibilités d'implantations sont ensuite décrites.

Keywords: Education, WWW, World-Wide Web, interactive documents, answer analysis, assessment.

Mots-clés : Enseignement, WWW, World-Wide Web, documents interactifs, analyse de réponse, contrôle de connaissances.

1. Introduction

The content of the World-Wide Web is growing at a more rapid rate than any previous information dissemination mechanism, and the tools and techniques available for providing and accessing such content appear to be

✉ Bertrand Ibrahim; Computer Science Department; University of Geneva; 24, rue du Général Dufour; CH-1211 Geneva 4 (Switzerland).
E-mail: Bertrand.Ibrahim@cui.unige.ch

✉ Stephen D. Franklin; Office of Academic Computing; University of California; Irvine, California 92717-2225.
E-mail: franklin@uci.edu

✉ Bernard Levrat; Computer Science Department; University of Geneva; 24, rue du Général Dufour; CH-1211 Geneva 4 (Switzerland).
E-mail: Bernard.Levrat@cui.unige.ch

growing at only a slightly slower rate. The Web is now widely appreciated and deployed within the educational community as a multimedia document delivery mechanism. There is also a growing awareness of its very substantial capabilities for interactive use, coupling the computational and information processing capabilities of local personal computers with those of larger shared-use systems.

It has long been recognized, within the computer-based learning community, that the quality of computer-based educational material was highly dependent on the quality of the assessment of the learner. Indeed, good educational material needs to adapt to learners' misconceptions. One should thus develop and deliver interactive courses which go beyond multimedia course materials to incorporate assessment of student progress.

2. Overview of current assessment techniques used on the Web

As Ulrik Schroeder[1] stated in his "HyperScript—innovative educational use of WWW"[2] position paper at the Teaching and Training Workshop[3] of the Third International World-Wide Web Conference[4], educators are interested in using the Web for the following reasons:

- one can build search and retrieval functions to search through the text of the educational material,
- one can provide cross-referencing within the material, to connect semantically related sections,
- one can provide different levels of details to let the learners focus on those parts they are most interested in,
- one can integrate multimedia material,
- one can include within documents information gathered in real time,
- one can integrate the material with other campus-wide information,
- the learner can interact via forms and server-side programs,
- the educational material can use the wealth of information resources available worldwide.

[1] http://www.isa.informatik.th-darmstadt.de/~uli/ulrik.html

[2] http://vim.ecs.soton.ac.uk/workshop-h-ulrik.html

[3] http://vim.ecs.soton.ac.uk/workshop-h.html

[4] http://www.igd.fhg.de/www95.html

A key point in this list is learner interaction, and educational material that is available on the Web is often weak in that respect. As Schroeder points out, good educational material should provide for active, task-driven and goal-oriented learning and encourage active student participation. We would add to this that the material should provide for adequate help and individualized interaction, based on an extensive assessment of the learner's abilities. Many educators, even those who have produced their own Web documents, do not see construction of such interactive materials on the Web as a realistic possibility for themselves. Part of the reason is that they view the Web as a document delivery mechanism and thus participate only as authors of particular documents. One can move beyond this both by improving understanding of the mechanisms of interactive documents and by using tools which help create such material. We turn first to the latter approach.

The first step beyond simple document delivery on the Web is the use of interactive forms[5], and there are an impressive number of very good explanations[6] of this capability. Most of these explanations, however, fall short of educators' expectations in two respects. First, they presume that the creation of a CGI-compliant[7] program to handle the form input is a task the reader can reasonably undertake. Second, educationally effective feedback to the learner must often be based on input other than the learner's most recent response. In both these cases, though, providing tools and explanations of those tools oriented to the needs and understanding of educators can overcome these obstacles.

One way educators can incorporate interaction with the learner in the documents they author is through the use of special tools that allow for the automatic handling of answers to questions within Web documents. For instance, David A. Wheeler's[8] *Mklesson*[9] downloadable tool lets one add multiple choice questions at the end of a document, with a different link attached to each possible answer. While the merit of multiple-choice test is debatable, it is certainly true that these tools lower the technological

[5] http://www.w3.org.hypertext/WWW/MarkUp/html-spec/html-spec_8html1#SEC72

[6] http://www.stars.com/Vlib/Providers/Forms.html

[7] http://hoohoo.ncsa.uiuc.edu/cgi/overview.html

[8] http://lglwww.eplf.ch/Ada/Tutorials/Lovelace/dwheeler.html

[9] http://lglwww.epfl.ch/Ada/Tutorials/Lovelace/userg.html

threshold for producing Web-based materials by those instructors who value such testing and that these tools open the door to other uses of the Web as well.

A more educationally ambitious and effective collection of interactive techniques is facilitated by Neal M. Holtz's[10] *Tutorial Gateway*[11], which lets one include questions involving multiple choice, true/false, single numeric, and single algebraic expression answers. Numeric and algebraic answers require form support in the WWW browser and the answer analysis involves a gateway program that will return an appropriate URL based on the answer.

It is however a common misconception that going beyond these answer handling techniques is much too difficult to be worth integrating in small scale educational material. Indeed, in spite of the fact that many educators recognize that certain objectives can only be reached by using open questions and free text answers, most people who tackle the problem of handling the learners' answers to questions consider that free text answer analysis is too complex a task to be manageable with current software technology, usually referring to artificial intelligence techniques to do lexical, syntactic and semantic analysis of the user input.

We will demonstrate further in this paper that reasonably simple techniques can achieve assessments protocols that do not allow the learner to guess likely answers from the questions that are formulated, thus ensuring a more reliable assessment of the learner's knowledge.

3. The Answer Analysis module

People tend to forget that rather simple pattern-matching techniques can be used to achieve free-text answer analysis with very reasonable effectiveness. For instance, the research group on computer assisted learning[12] at the Computer Science department of the University of Geneva has developed, and used in stand-alone CBL programs, pattern matching tools

[10] http://www.civeng.carleton.ca/People/Faculty/Neal_Holtz/Bio.html

[11] http://www.civeng.carleton.ca/~nholtz/tut/doc.html

[12] http://cuiwww.unige.ch/eao/www/CBL.papers/CBL.group.intro.e.html

(Franklin *et al.*, 1985) that make it rather easy to specify, in a symbolic formalism, criteria for various anticipated answer categories. The "rules"[13] used to specify answer analysis are stored separately from the actual educational material to ease translation and maintenance.

This answer analysis uses simple string comparison to determine if the user input (representing the student's response to a question) matches an expected response category as specified in the rules written by the designers of the material. These rules follow a syntax that can be easily mastered by teachers participating in the original design, or by those who may have to modify the rules to handle situations overlooked in the original design. The approach is easy enough to use to allow the educators to remain in control of all the various aspects that decide of the pedagogical value of the educational material.

Basically, the rule formalism allows the designers to specify combination of words that should or should not be found in the learner's answer. Lists of synonyms, as well as partial words can, also be provided. A detailed description is available in the on-line documentation[14]. We will therefore only examine here a simple example that is used to check whether a learner answered something equivalent to "yes":

```
_YES_|_YEP_|_OK_|_RIGHT_|_FINE_|_YEA|_OF COURSE_|_SURE_|and|not|_NO_|_NOT_|N'T_
```

In this notation, all the words that should be matched against the learner's answer are in upper case, the underline character stands for any character that is not a letter or a number, the vertical bar separates alternatives (equivalent to an **or** operator), and "**|and|not|**" is an operator indicating that none of the alternatives that follow should be found.

4. Adaptation to the Web

In spite of its simplicity, this approach gives astonishingly good results with carefully chosen questions and well thought-out rules for expected answers. Its integration to WWW-based material can be done in various ways. The tools based on this mechanism are meant to be used by document

[13] http://cuiwww.unige.ch/eao/www/FreeTtext/answeris/html

[14] http://cuiwww.unige.ch/eao/www/FreeText/answeris.html

authors who want to include, in their documents, input fields that the document reader (the learner) can fill in, and that will lead to different other documents based on the (free text) input that the learner has given.

We have currently implemented two approaches involving server-side computing. Both approaches assume the use of a simple form with a text input widget for the learner input. The form contains the various analysis criteria and corresponding URLs in hidden fields. This information will be used by a cgi-bin script on an HTTP server to do the pattern matching and select the appropriate URL to send back a "Location:"[15] reply to the WWW client that will, in turn, display the corresponding document.

Authors do not need to know how to put forms in their documents, as our tools will do it automatically for them. The author can specify the criteria that will be matched against the user input to classify this input in different categories. All the possible input that match a given category will trigger the same behavior, *i.e.* will lead to the same document.

In this model, we are making the assumption that a document will usually contain only one such text input form or, if the document contains more than one form, only one will be used at a time, leading to one of many possible successor documents, based on the category of the user input.

The two approaches we have implemented involve on-line form-based tools to automatically create such a user input form and embed it in an HTML document. An initial prototype[16] assumed that all the information necessary to create the answer form (with hidden fields containing the criteria and the corresponding URLs) would be entered via an on-line form[17]. This was easy to put in place, but was not very convenient for maintenance, as the content of the various fields of the form cannot be saved by the designers for later modification and thus have to be re-entered every time the designers decide to change the analysis criteria, unless the filled form is still available within the history cache of the Web browser.

The second prototype[18] assumes that the criteria are prepared in a www-accessible file[19] and that a form[20] is mainly used to give the URL of

[15] http://hoohoo.ncsa.uiuc.edu/docs/cgi/out.html

[16] http://cuiwww.unige.ch/eao/www/FreeText/Info.html

[17] http://cuiwww.unige.ch/eao/www/FreeText/DeveloperForm.html

[18] http://cuiwww.unige.ch/eao/www/FreeText/Info2.html

[19] http://cuiwww.unige.ch/eao/www/FreeText/TempCrit.txt

[20] http://cuiwww.unige.ch/eao/www/FreeText/DeveloperForm2.html

the original document (in which an answer form must be embedded) and the URL of the criteria file. This makes it easier to update and maintain the criteria for a whole set of documents with embedded answer forms. When an educator uses this second prototype, providing the two URLs and the symbol used as a placeholder, the server fetches both the original document and the criteria file, finds the analysis patterns in the latter, replaces, in the original document, the placeholder with the form that the learner will see and that contains all the pattern matching information in hidden fields. The resulting document is then returned to the educator's browser, ready to be saved locally (e.g. using the File/Save As browser menu item) and integrated into a set of educational documents. When a learner uses the resulting document and fills in the form, the same pattern matching program is activated as with the first prototype.

5. Client-side processing

As powerful an educational tool as server-side computing can be, it is not without its limitations. Network bandwidth and cumulative system load on a server required to meet non-trivial computational demands from multiple simultaneous requests can easily turn "interactive" into "slow batch processing." The situation is aggravated by the comparison with the speed of local processing and by the importance of predictable-time-to-respond in an important and extensive range of educational applications. Educators can always install the pattern matching software on their local WWW server, to be less dependent on some remote machine over which they have no control, but this raises portability issues that are not necessarily obvious to solve.

An alternative to the server-side pattern-matching approach described above is to use client-side computing to do all the pattern matching directly on the learner's machine. Similarly to the server-side approach, the pattern matching information would be embedded in the document displayed to the learner, but this time as parameters to an applet that will handle both the learner's input and the pattern matching. This solution would still require access to a remote server the first time the applet is loaded, but hopefully not the following times, assuming that the applet-capable browser caches it locally and reuses it whenever this applet is used within other documents.

Implementing this client-side approach mostly consists in translating the pattern matching code into whichever programming language appears to be supported by the widest range of Web browsers. We are currently working on an implementation, based on the Ada to Java[21] bytecode compiler[22] from Intermetrics, that will be compatible with Java-enabled browsers. The cgi-bin script that currently substitutes a placeholder with an answer input form in a document, will simply be updated to embed an applet tag instead. The educators will not need to see any change in the way they prepare their documents, compared to the existing prototypes mentioned earlier.

6. Comparison of the two approaches

Even though in most cases it will be more expedient to do the pattern matching on the learner's system, there can be cases where analyzing the answers on a server may make more sense. A centralized approach makes it indeed easier to collect statistics and keep a database of user profiles (assuming that the learners enter their name before they can start using the educational material). On the other hand, the client-side approach makes it easier to keep the profile information directly in the learners' disk space, allowing the learners to remain anonymous if they wish so.

Another major difference between the two approaches is the ability to time user input. With the server-side approach, a mechanism could be put in place to evaluate the time elapsed between the moment the document with the answer input form was sent to the learner and the time at which the answer is sent back by the learner, but it would be difficult, using the standard HTTP protocol[23], to modify the document seen by the learner without any learner intervention (the authors are, of course, aware of the *Netscape* server-push technique[24], but it is not a standard use of the HTTP protocol and it is not available with public-domain WWW server software). On the other hand, it would be fairly easy, with the client-side approach, to implement in the user input applet a specific behavior based on the time elapsed since the applet started. The respective advantages and disadvantages of both approaches can be summarized as follows:

[21] http://java.sun.com/

[22] http://www.inmet.com/java.html

[23] http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html

[24] http://home.netscape.com/assist/net_sites/pushpull.html

	Advantages	Disadvantages
server-side pattern matching	– easy to put in place. – easier to maintain a database of learner profiles. – usable with any browser supporting forms.	– more network traffic. – slower response time. – no learner profile if anonymity has to be preserved. – individualization limited to varying the documents that are shown to the learner. – timing cannot be used flexibly.
client-side pattern matching	– less network traffic – instant response time – anonymity can be preserved while maintaining a local learner profile. – individualization can consist in more than just varying the documents that are shown to the learner. – timing can be used easily to vary the learning server as experience.	– more difficult to maintain a database of learner profiles. – more complex to put in place. – requires a browser supporting applets. – may require to lower security protection for applets, unless all educators install a copy of the applet on the same as the educational material.

There is no need to choose between the two approaches. Indeed, it may be that one reverts to a server-based analysis if a client-side one is not possible. Or it may be that one falls back on client-side processing when server response is unacceptable. However, based on the above table, one can see that the two approaches are not completely equivalent and the choice an educator will make will depend on individual factors and the relative importance of the various advantages and disadvantages that we have listed.

7. Conclusion

We have described, in this paper, a method and various implementations that allow educators to embed free-text answer analysis within Web documents. This method allows for better assessment of the knowledge acquisition process, since there need not be any element of answer within the questions that are asked to the learners. This method has successfully been used in the past within stand-alone computer based learning material and we have shown that this method was applicable to today's standard Web technology (forms) as well as to more experimental technology (applets).

Moreover, we have shown that the interface to the developers could be the same regardless of the technological choice.

Balancing technical capabilities and educational realities has never been an easy task. The World Wide Web has opened a range of educational opportunities unprecedented in their scale and scope. Appreciable progress has already been made and, looking ahead to new possibilities, we see the importance of integrating Web technology (and limitations) with the educationally-centered perspective that is created through a focus on specific curricular challenges.

Bibliography

FRANKLIN (Stephen D.), LEVRAT (Bernard): 1985, "Portability of Computer Based Learning Materials across machines and across natural languages", *Computers in Education*, DUNCAN (K.) and HARRIS (D.) eds., (North-Holland: Elsevier Science Publishers B.V.), IFIP, pp. 291–295.