

Semantic Bookmarking for Non-Visual Web Access *

Saikat Mukherjee
Dept. of Computer Science
Stony Brook University
Stony Brook, NY, 11794, USA
saikat@cs.sunysb.edu

I.V. Ramakrishnan
Dept. of Computer Science
Stony Brook University
Stony Brook, NY, 11794, USA
ram@cs.sunysb.edu

Michael Kifer
Dept. of Computer Science
Stony Brook University
Stony Brook, NY, 11794, USA
kifer@cs.sunysb.edu

ABSTRACT

Bookmarks are shortcuts that enable quick access of the desired Web content. They have become a standard feature in any browser and recent studies have shown that they can be very useful for non-visual Web access as well. Current bookmarking techniques in assistive Web browsers are rigidly tied to the structure of Web pages. Consequently they are susceptible to even slight changes in the structure of Web pages. In this paper we propose *semantic bookmarking* for non-visual Web access. With the help of an ontology that represents concepts in a domain, content in Web pages can be semantically associated with bookmarks. As long as these associations can be identified, semantic bookmarks are resilient in the face of structural changes to the Web page. The use of ontologies allows semantic bookmarks to span multiple Web sites covered by a common domain. This contributes to the ease of information retrieval and bookmark maintenance. In this paper we describe highly automated techniques for creating and retrieving semantic bookmarks. These techniques have been incorporated into an assistive Web browser. Preliminary experimental evidence suggests the effectiveness of semantic bookmarks for non-visual Web access.

Categories and Subject Descriptors

K.4.2 [Computers and Society]: Social Issues—*Assistive Technologies for persons with disabilities*; H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia—*Navigation*; I.7.5 [Document and Text Processing]: Document Capture—*Document Analysis*

General Terms

Algorithms, Human Factors

Keywords

Bookmarks, Semantic Partitioning, Assistive Browsing

* Work supported in part by the NSF grants CCR-0311512 and IIS-0072927.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASSETS'04, October 18–20, 2004, Atlanta, Georgia, USA.
Copyright 2004 ACM 1-58113-911-X/04/0010 ...\$5.00.

1. INTRODUCTION

The World Wide Web has become an indispensable aspect of our society. However, the primary mode of interaction with the Web is via browsers designed for visual modes of interaction. This limits access to the Web for an entire community of people with visual disabilities. This target population faces particular difficulties in accessing, scanning, summarizing, and distilling information on a Web page or group of pages.

Assistive devices like screen readers and audio browsers such as Freedom Scientific's JAWS are being used by blind users to access the Web. However, the problem with these systems is that they provide almost no filtering of Web page content to eliminate "noise". The user is forced to arrow down or page down through a single columned presentation of the entire content in a given page including the text, navigational links, and advertisements. This is due to the nature of the way they present a particular Web page whereby the entire content is rendered as a single column regardless of the structural organization of the content in the page. In short, the user encounters considerable information overload when using these kinds of assistive tools.

One interesting approach to address the information overload problem during ad hoc exploratory browsing is to summarize the content. In fact, this important and useful idea was first explored in BrookesTalk [31, 32]. Another idea is to provide the user with the logical structure of a Web page's content, and the user selects which parts of the document to listen to and when to navigate to a new page. Generating such a logical structure would require capturing the semantics of the page's content. Indeed this seminal idea was pursued in some recent papers on assistive browsing [24, 22, 28, 29, 13]. Of course, browsing based on the logical structure can be integrated with summarization to yield still greater reductions in information overload.

Exploratory browsing is used for ad hoc access to Web pages. Quite often users perform repetitive browsing tasks, *i.e.*, frequent and periodic visits to certain predefined pages at a site or groups of sites (*e.g.*, news and educational sites). Efficiency of such tasks is considerably improved by personalization. A simple form of personalization is bookmarking. Bookmarks provide the user with direct access to pages, via a Web browser, that are repeatedly accessed for information.

This problem of creating bookmarks and retrieving information with them, via assistive Web browsing tools, for individuals with visual impairments has been pioneered in [7, 19]. By and large these works propose syntax-driven solutions for this problem. In such solutions the Web page is viewed as a sequence of HTML tags. Creating bookmarks correspond to placing anchors in this sequence around the desired content which typically are sentences/phrases. Retrieving information with a bookmark amounts to identifying



Figure 1: The New York Times front page

these anchors and reading out the content enclosed by them. The serious limitation with syntax-based solutions for bookmarking is that they are very brittle to structural variations in the page. Thus they are difficult to maintain. The maintenance problem is further exacerbated if one has to create and maintain bookmarks for “similar” content from different Web sites.

In this paper we propose an alternative solution. We introduce the idea of *semantic* bookmarks. In this approach the user is presented with the logical structure of the Web page organized around an ontology, which consists of semantic concepts relevant to a domain. For example, an ontology for the news domain will consist of concepts such as “Taxonomy News”, “Major Headline News”, etc. Segments of a Web page, corresponding to its logical structure, are classified and assigned the concept names as semantic labels. An assistive browser presents these semantic labels according to the logical structure of the Web page. Upon hearing a semantic label, a user can then instruct the browser to bookmark the concept. When a bookmark is retrieved the Web page is fetched, and its content is *automatically* reorganized into a logical structure and semantically labeled with respect to the ontology. The contents of the segments in this structure, whose labels match the bookmark’s label, are read out to the user. Consequently, as long as the concept instances are correctly identified in the Web page, semantic bookmarks are robust to structural variations. Besides, since ontologies are domain specific the scope of a semantic bookmark extends to all those sites whose content semantics are represented in the ontology.

The rest of the paper is organized as follows. The next section describes the technical aspects underlying semantic bookmarking. In the evaluation section we present preliminary evaluation of our implementation of semantic bookmarking. In the related work section we compare with existing approaches and finally conclude in the discussions section.

2. SEMANTIC BOOKMARKING

Our approach to identifying the logical structure of a Web page rests on inferring the semantics of its content using two processes, namely, *structural analysis* and *semantic labeling*. The former groups elements in a Web page into partitions according to their structural type, spatial locality and semantic similarity, while the latter labels the resulting partitions according to content, relationship with other parts, or type. We now present a technical overview of these two processes.

2.1 Partition Construction and Labeling

In partition construction, we use the HTML tags of Web pages to partition them into semantically-related units, which are organized and then presented to users as a concept hierarchy or partition tree (see Figure 2(b) for an example). Our approach is based on the simple but productive observation that semantically related items in a Web page normally exhibit consistency in presentation style and spatial locality. This is particularly true of content-rich Web sites that update frequently such as news portals, education and e-commerce sites, because these sites are typically maintained using content management software that creates Web pages by populating templates from backend databases.

For example, in the New York Times front page (shown in Figure 1), there is a fixed news taxonomy on the upper left corner. There is also an implicit template for major headline news items. Each of these items begins with a hyperlink labeled with a news headline (e.g., “Supreme Court ...”) followed by the news source (e.g., “By DAVID ...”), followed by an optional timestamp, text summary of the article (e.g., “The Supreme Court ...”), and some pointers to related news (e.g., “Padilla ...”). There are also presentation similarities in the items in the news taxonomy of the New York Times front page (Figure 1). The main items, “NEWS”, “OPINION”, “FEATURES”, etc., are all presented in bold font. All the subtaxonomic items (e.g., “International”, “National”, “Washington”, etc.) under a main taxonomic item (e.g., “NEWS”) are hyperlinks. This kind of consistency in presentation style is reflected in the Document Object Model (DOM) tree of an HTML document. For example, Figure 2(a) depicts a fragment of the DOM tree for the New York Times front page shown in Figure 1. The root-to-leaf sequences of HTML tags for the nodes “NEWS” and “FEATURES” are exactly the same, as are the sequences of HTML tags for the nodes “International”, “Arts”, etc. (font tags with different attributes, e.g., size, are distinguished using different subscripts in Figure 2(a)).

Spatial locality in a Web page and its corresponding DOM tree can also indicate content similarity. For example, when rendered in a browser (see Figure 1), all the taxonomic items for the New York Times are placed in close vicinity occupying the upper left portion of the page. In the corresponding DOM tree all these taxonomic items are grouped together under one single subtree rooted at the table node (see Figure 2(a)). Similarly, all the major headline news items are clustered under a different subtree rooted at the *td* node (shown circled in Figure 2(a)).

Structural Analysis. Consistency in presentation style and spatial locality in Web documents can be identified by looking for recurring patterns in the path structures of the corresponding DOM trees. For example, the root-to-leaf path strings of the news taxonomy items, such as “NEWS”, “OPINION”, “FEATURES”, etc., in Figure 1, which consist of tag names and their associated attributes are all identical. Let us denote the path string $tr \cdot td \cdot table \cdot tr \cdot td \cdot img$ from Figure 2(a) using T_1 , and $tr \cdot td \cdot table \cdot tr \cdot td \cdot a \cdot font_0$ using T_2 . The subtree rooted at *table* in Figure 2(a)

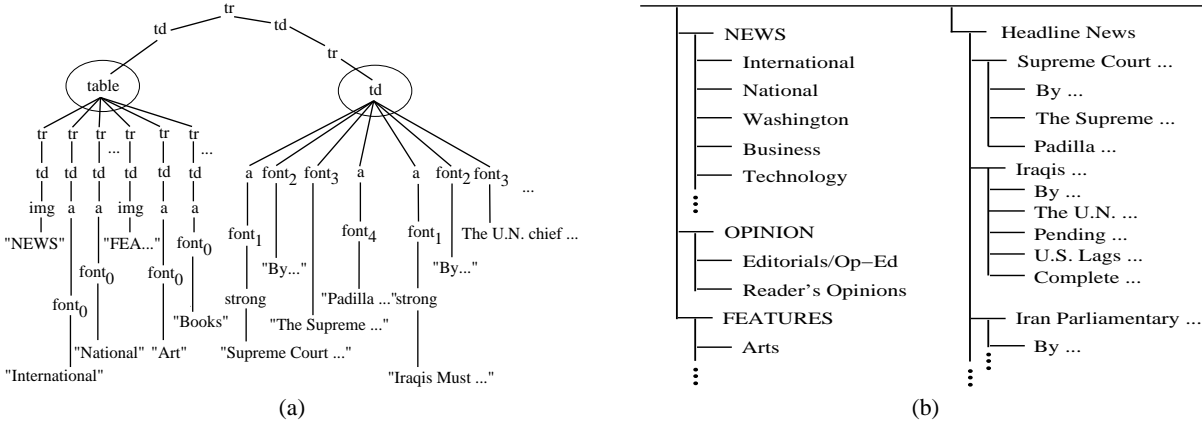


Figure 2: (a) DOM fragment of the New York Times front page (b) Semantic Partition tree of the New York Times front page

has the following string: $T_1T_2T_2 \dots T_1T_2T_2 \dots$. The problem of spatial locality discovery can then be reduced to the problem of *sequential pattern analysis*. For instance, $T_1T_2^*$ (here $*$ denotes Kleene closure) is a sequential pattern that can be observed from the string $T_1T_2T_2 \dots T_1T_2T_2 \dots$, in which T_1 represents a taxonomic item such as “NEWS”, “OPINION”, etc., and T_2^* represents the collection of hyperlinks, such as “International”, “National”, etc. Similarly, the subtree rooted at *td* in Figure 2(a) has the string sequence $T_3T_4T_5T_6T_6 \dots T_3T_4T_5T_6$. Here T_3, T_4, T_5, T_6 denote the the path strings from *tr* to a leaf, in Figure 2(a), containing $font_1, font_2, font_3, font_4$ respectively. Analysis of this string yields the pattern $T_3T_4T_5T_6^*$ which essentially captures an individual major headline news item. Sequential pattern discovery can be performed recursively bottom-up starting from the leaves of the DOM tree of a Web page. Near-leaf patterns correspond to small partitions. Small partitions can be aggregated into bigger partitions for content-rich Web pages.

Semantic Labeling. After partitioning a Web page, we need to assign informative semantic labels to the partitions. The labels act as summaries of the information in each partition. Usually the labels of (small) partitions deep in a partition tree are present in the document itself (e.g., “NEWS”, “OPINION”, etc. in Figure 1). When such a label is present in a document, it is usually the first text item in the partition. Adopting this heuristic results in extracting meaningful labels for many partitions. However, this simplistic heuristic is inadequate for labeling partitions which are aggregated from smaller partitions down the tree. These bigger partitions typically reflect semantic concepts present across pages in the same domain. For instance, the aggregated partition of “NEWS”, “OPINION”, etc in Figure 1 is an instance of a “Taxonomy News” concept. Such an instance is also present in other news portal sites e.g. CNN, Google News, etc. To label these partitions, a domain ontology with a set of concepts and a classifier for every concept to match partitions to it is used. A news domain ontology will typically have concepts like “Taxonomy News”, “Major Headlines”, “Category News” (category specific news), etc. The classifier for a concept exploits the common features of it’s instances across pages in the domain to identify partitions. For instance, the classifier for “Major Headlines” will match those partitions as concept instances which have a hyperlink news headline, a news source, a news summary in text, and some optional pointers in hyperlink to related news. Application of structural and semantic analysis on the New York Times page in Figure 1 generates the semantic partition tree shown in Figure 2(b).

Tight coupling of the structural and semantic analysis yields a robust semantic partitioning system. Experiments on over 100 Web pages collected from a dozen different news e-commerce product portals have demonstrated the accuracy of partitioning and yield of concept instances present in the page to be above 90%. More details about our algorithms and experiments can be found in [20].

2.2 HearSay Pilot

We developed HearSay – a prototype speech-driven Web browser embodying the ideas of structural analysis and semantic labeling for creating the logical structure of Web documents. To enable speech-driven browsing HearSay automatically generates a VoiceXML dialog¹ interface to this logical structure. The current implementation of HearSay uses IBM’s WebSphere toolkit² for VoiceXML interpretation, speech recognition and text-to-speech. Users are allowed to interrupt system prompts at any time. Details of the the HearSay architecture, implementation, and evaluation by both non-blind and blind users appear in [25]. The domains that can be currently handled by HearSay include news, education, and e-commerce. For these domains we have constructed simple ontologies that are used during the labeling process.

In the rest of this paper we exploit and expand the logical structure for automatic creation and retrieval of semantic bookmarks. While semantic bookmarks are applicable to both visual and non-visual access, we will refer to speech-based creation and retrieval of bookmarks as *voicemarking*. We have extended HearSay to include Voicemarking features whose technical details are described next.

2.3 Voicemarking

Personalization is considered an important technology for Web access. The primary objective of personalization technology is to avoid “information overload” when people interact with Web sources. With non-visual Web access, information overload is an even more serious problem since direct translation of Web pages into VoiceXML can result in many repeated and uninteresting utterances. The human eye can filter this “noise” quickly, but audio output is necessarily a more narrow output channel and listening to a lot of irrelevant audio output can be a trying experience.

The present state of the art in personalization is *server-side* personalization through “collaborative-filtering” (e.g., Amazon.com

¹VoiceXML is an XML-based W3C standard for specifying voice dialogs on the Web.

²http://www-306.ibm.com/software/pervasive/voice_toolkit/

and other on-line retailers). Server-side collaborative-filtering uses information from Web server logs to classify users into groups with similar interests. It then creates special presentation templates for each group.

Since audio provides a narrow communication stream, this approach is no longer sufficient. Customization for non-visual access must be augmented with *client-side* personalization strategies to let visually impaired users themselves identify page segments of interest. One way to realize client-side customization is through *voicemarks*. A voicemail is like a bookmark in an HTML browser, but is more precise in terms of which part of the Web page is of interest, and, of course, voicemarks are accessible by voice. Voicemarks can, for example, point to the portions of the CNN Web site that contain information about weather, education, or health. A voicemail might also be temporarily placed in an audio stream during playback in order to enable the listener to switch to another stream and return to the original one later. This latter use of a voicemail is especially valuable for students dealing with courseware on the Web.

Voicemarking is becoming possible through the increasing availability of speech recognition systems, especially in VoiceXML-enabled browsers. We now highlight some of the more interesting extensions to the basic idea of voicemarking, which makes it more useful and flexible.

Creating persistent voicemarks through dialogs. A visually impaired person needs a method to enable listening to Web pages segment-by-segment and mark segments of interest with personal identifiers for later retrieval. A *voicemail* is a triple of the form $\langle VoiceMarkName, URL, ExtractionExpression \rangle$.

VoiceMarkName is chosen by the user and the URL of the current page can be obtained from the voice enabled browser. The problem, then, is how to generate an extraction expression for the HTML segment of interest. Our approach to the problem is to let the system determine the extraction expressions through a combination of the semantic partitioning technique described earlier and the “robustification” algorithms described in [4]. The robustification algorithms can eliminate the need for recomputing extraction expressions due to moderate changes in the structure of Web sites and thus reduce the need for re-using the relatively expensive semantic partitioning algorithm.

In case of non-visual access, the Web page is read to the user segment by segment. The user can voice-mark a segment by giving it a *name* and *anchor phrases*. The anchor phrases can be used to filter out segments that do not contain any of the phrases and thus are probably of no interest to the user.

For example, consider a student who is studying for the SAT and wishes to use the practice math questions at web sites³. The math question changes daily, and appears on a page with quite a bit of other information, including links to other sites and advertisements. The title of the part of the page with math questions is always “Today’s SAT Math Teaser.” So the student can voice-mark this part of the web page by giving it a name, e.g., “math question”, and supplying other anchor phrases, such as “SAT,” “teaser,” and “answer.” The combination of these phrases will eliminate most other information on the page but may still match non-math questions. When the student recalls this page (by saying “math question”), all the matched content will be produced in VoiceXML and the user will have an opportunity to direct the system to drop segments that are of no interest. To make voice-marking more precise, we can also allow the user to supply *negative* examples, i.e., contents that the user does not want to hear.

Our experiments with VoiceXML show that suitable heuristic algorithms can be developed to rank text segments for their relevance to the anchor phrases as well as the negative examples. Based on this information, VoiceXML dialogs can be created that will allow the user to use voice to indicate which segments are to be retained.

Transient voicemarks. Another interesting application of voicemarking is to make voice-enabled contents more interactive. While listening to an audio stream, the user can voicemark various segments, which would enable her to come back to these segments later. This is similar to how we work with books: While reading a paragraph, we might need to look up a reference or a fact and then come back to the main text. The important feature of our voicemarking technique is that it is highly personalized. For instance, the instructor who created the courseware does not need to plan in advance how and where a non-sighted student will need to interrupt the audio stream.

Natural language assistance. Users will have trouble remembering large sets of voicemarks. To mitigate this concern, we can augment voicemarking with named entity extraction and synonym identification (using a lexical resource such as WordNet [18]). These will permit users to use combinations of voicemark names and anchors, and words related to voicemark names and anchors, to recall voicemarks. For example, if the user wants to recall the math question voicemark described earlier at some later date, she could say any of the phrases “math question,” “math teaser,” or “sat math.” If a user wants to recall a temporary voicemark named “notetakers,” she could say “notetakers,” “notetaking,” “notes,” or even “transcription”.

3. EVALUATION

Voicemarking has been partially implemented in HearSay. In particular, the system permits the users to create and retrieve voicemarks using HearSay. Here we report on our preliminary evaluation of voicemarking in HearSay. We used news portal sites for our study. A Web page in a news portal typically consists of instances of the concepts “Major Headlines”, “Taxonomy News”, and “Category News”. We selected New York Times, CNN, and Google News as our test sites. Instances of the concepts “Major Headlines” and “Taxonomy News” for the New York Times page are shown in Figure 1. The experiments were conducted using four subjects (identified as subjects A, B, C, and D).

We had two objectives in conducting our evaluation. The first was to evaluate the advantage of voicemarks (in a quantitative sense) when compared to simple screen-readers and more sophisticated assistive browsers. Our set-up consisted of the BrookesTalk screen reader, HearSay without voicemarks and HearSay with voicemarks. We did not assume familiarity of the users with any of the three systems. Since voicemarks are mainly used for faster access to content, our experiments focused on the time taken to access a semantic concept with and without voicemarks. So we measured the time taken to access a particular concept instance, in this case “Major Headlines”, for the four subjects across the three sites using our experimental set-up. The results of this experiment are summarized in Figure 3. In Figure 3(a), the bars for BrookesTalk, HearSay (without voicemark), and HearSay (with voicemark) for “Major Headlines” represent the times taken, averaged over the four subjects, to access the beginning of the “Major Headlines” concept instance in the three sites. Observe the considerable reduction in time for HearSay (without voicemarks) compared to BrookesTalk. Such a reduction was uniformly observed for all the three sites. This illustrates the effectiveness of the reorganization of Web pages into its logical structure for efficient browsing. For HearSay (with voice-

³For instance at <http://education.yahoo.com/college/mpotd/>

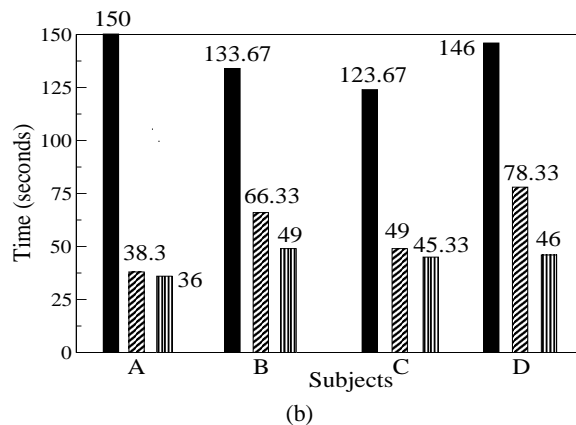
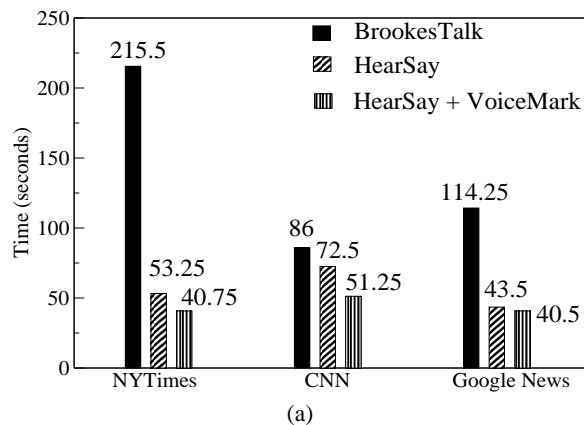


Figure 3: Comparison of BrookesTalk, HearSay, and HearSay with VoiceMark for the Major Headlines news concept (a) in different sites (b) among different users

mark), we assumed that the user had previously voicemarked the “Major Headlines” concept in the three sites. We then measured the time taken to access an instance of this concept via the voicemark. Observe that for all three sites, voicemarking achieves a further reduction in access time. Figure 3(b) (having the same legend as in Figure 3(a)) shows the time taken by each subject to access the “Major Headlines” concept averaged over all the three sites. Observe that for all the subjects HearSay offers a significant reduction in access time over BrookesTalk. Even further reduction is observed when using HearSay with voicemarks. Figures 3(a) and (b) provide empirical evidence that the voicemarking feature of HearSay facilitates efficient access to content.

The second objective was to evaluate the effect of voicemarking on retrieving specific news items embedded within concept instances. Specifically we were interested in evaluating the efficiency of accessing text present inside arbitrary concepts by voicemarking. Our four subjects were each asked to answer three questions that dealt with retrieving a specific phrase from two different concept instances across two different sites. These three questions were graded from easy to hardest in terms of the difficulty in answering them. The questions were:

- (Easy) Write down the items in the New York Times “Category News” concept instance which contain the phrases “George Bush” or “John Kerry”.
- (Medium) Write down the items in the New York Times “Major Headlines” concept instance which contain the phrases “George Bush” or “John Kerry”.
- (Hard) Write down the items in both the New York Times and CNN “Category News” concept instances which contain the phrases “George Bush” or “John Kerry”.

The difficulty in answering such questions increases with text length. Also, correlating information from two different sites, for instance New York Times and CNN, is a challenging task. The time required to answer each of these questions using HearSay without voicemark and HearSay with voicemarks was measured for every subject. For measuring the time taken when using HearSay with voicemark, it was assumed that the subject had previously voicemarked the “Major Headlines” and “Category News” concepts in New York Times and CNN. Figure 4 shows the time taken, averaged over the four subjects, to answer these three questions. In the figure, column 1 shows the time taken to answer the questions using

Question	HearSay (sec)	HearSay+VM (sec)	Time Reduction (%)
Easy	546	521	4.58
Medium	935	870	6.95
Hard	1228	1112	9.44

Figure 4: Time taken to access different news concepts via voicemarks

HearSay without voicemarks, column 2 shows the time taken using HearSay with voicemarks, and column 3 measures the percentage reduction in time when using HearSay with voicemarks. Observe, as expected, the time taken to answer increases with the hardness of the questions. Note also that the effectiveness of voicemarking, as measured by the percentage reduction in answering time, increases with the hardness of questions. In fact, for a blind user, voicemarking is most effective for retrieving information from diverse sources. Our experiment seems to empirically demonstrate the utility of semantic voicemarking for efficiently searching arbitrary text in concept instances present within the same and different Web pages.

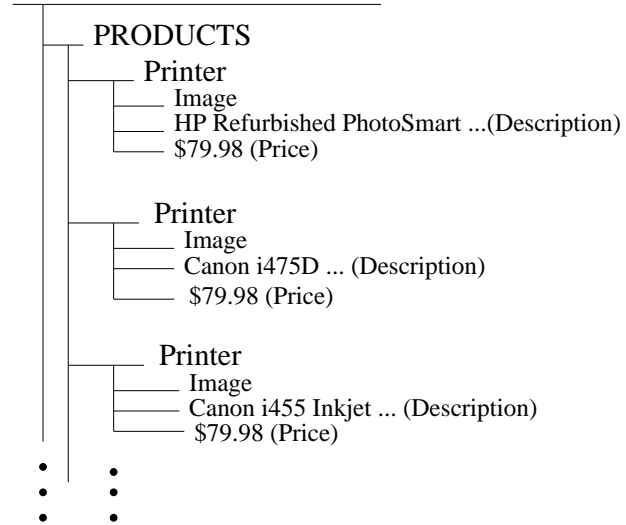
Recall that HearSay’s rendition into audio of the semantic concept instances present in a page follows the (hierarchical) logical structure of the Web content. Thus, it takes more time for HearSay to access semantic concepts that are deeper in this hierarchy. For instance, from Figure 2(b), it takes more time for HearSay to access the “Business” item than the “News” item. Thus the reduction in time achieved using HearSay with voicemarks over HearSay without voicemarks would be greater for semantic concepts lower down in the tree. Our current ontology for news domain is shallow and so the efficiency differences using HearSay without voicemarks and HearSay with voicemarks is not that big.

4. ACCESSING TABLES WITH HEARSAY

The use of HTML tables for presenting content is widespread in the Web. While being an excellent means for arranging information, tables provide the sight-impaired with additional challenges for navigating and understanding the content. Consequently, there has been a lot of research in improving the accessibility of tabular content for the visually challenged [1, 26, 8, 22, 23, 21]. The lack of comprehensible aural rendition of table content in initial approaches [1, 24, 21], focusing on content linearizing, prompted researchers to incorporate contextual reinforcement techniques for



(a)



(b)

Figure 5: (a) OfficeMax Printer Products Web page and its (b) Semantic Partition Tree

a better rendition. While such techniques [8, 26, 23, 28] can be applied to a wide variety of tables with a very high degree of comprehension, they usually require content providers to annotate the content. This dependence on content providers has limited their wide applicability despite being a powerful technique.

Recall from Section 2.1 that the partitioning technique in HearSay is able to group together semantically related content for a wide class of HTML pages. In the context of HTML tables, this usually enables the entire content in the table to be associated in a single partition. Moreover for well structured tables, common in content-rich Web sites, every individual row's content is also grouped into separate partitions inside the corresponding table's partition. When the content in the table can be identified by a domain ontology, it becomes possible to associate semantic labels to individual rows and columns. These semantic labels can then be used as further context while aurally rendering the table's content. In particular, the start of every row and the semantics of columns can be incorporated into the aural rendition. Thus, given a domain ontology, for a large class of tables HearSay can be used to provide a more comprehensible aural rendition than simple content linearizing and yet not depend on individual content providers for annotation.

Figure 5(a) shows a printer products page in the Web site of OfficeMax. The products are presented in a HTML table. Observe that every individual printer is characterized by the three attributes: (a) an image of the printer, (b) a text description of the printer, and (c) the price of the printer. The attribute price is characterized by the symbol "\$" while the printer description usually has the keywords "Printer", "HP", "Canon", etc. A printer product can be characterized by the presence of the description and price attributes. A simple printer ontology can be easily created based on these keywords. Using such an ontology in combination with HearSay results in the semantic partition tree of the OfficeMax page shown in Figure 5(b). Aural rendition of this page is made more comprehensible by knowledge of the beginning of a printer, and of the semantics in the text of the description and price columns of each individual printer. Observe also that, in this particular example, the table does not even contain the column names (description, price, etc). The use of domain ontologies coupled with HearSay's partitioning enables the semantic annotation of content which leads to comprehensive aural rendition of tables.

5. RELATED WORK

Assistive technologies for Web accessibility have spurred substantial research and development activities. On the browser front, several research projects devoted to browser-level support for accessibility emerged. They came in response to the need for adapting Web browsers for accessibility support—an issue raised in various works [2, 12]—and to address the ineffective performance of existing screen readers demonstrated in experimental studies [6, 10]. For example, the work of Asakawa et al. [1, 21], which was integrated in the IBM Home Page Reader, allows navigation at the level of links, i.e., move from link to link, words, or page elements. Dillin's work converts HTML into a MS-NOTEPAD style of presentation [5]. Other efforts include the pwWebSpeak browser that offers speech output through HTML interpretation, the BrailleSurf system that allows filtering of HTML into Braille [11], the Census Internet Browser which uses Windows-eyes for Internet browsing [9], the BrookesTalk system which uses text abstracting techniques to facilitate "scanning" of web pages [31], the Audio Hallway program [27] and NLP based summarization systems for Web page access [32]. There has also been some work on software engineering methodologies for developing non-visual interfaces [30], and development of applications for Web-based course delivery for students who are visually impaired (e.g., [14, 15, 16]). But all of the above works transform Web documents for accessibility using low-level structural properties of HTML (tag-type, etc). In contrast, using structural analysis we generate higher-level partitions based on aggregating semantically related elements.

Techniques resulting from early efforts at bookmarking for non-visual Web access were incorporated in the DAISY system [19] and the IBM HomePage Reader [1]. In the DAISY system, which renders digital books into audio, users could place "bookmarks" at a reading position and use these bookmarks to jump to arbitrary positions in the text. However, only phrases could be bookmarked and as a result this facility could be used only for static documents whose content does not change over time. In contrast, the IBM HomePage Reader allowed users to "fast-forward" to a desired section of a document by speeding up the audio. The primary problem with fast-forwarding was the effort required to reach the desired position in the text since speeding up the audio can result in going

beyond or falling short of that position. More recently, [7] describes a method for bookmarking sentences in Web pages and searching for updated content in a page via these bookmarks. However, these bookmarks are built from the sequence of HTML tags in the page. The primary limitation of these syntax-based approaches is their brittleness to structural variations in the Web page. In contrast semantic bookmarking as embodied in HearSay is more resilient to such variations. Besides their scope extends across Web sites whose content can be captured by the semantics of a domain specific ontology. This not only results in efficient retrieval of bookmarked content from multiple Web sources but also simplifies their maintenance.

The idea of creating a logical structure for organizing Web content to facilitate more comprehensible navigation is the basis of the research work in [29, 13, 23]. Specifically, [29] segments a page into regions where each region matches a pre-defined layout pattern associated with a particular semantics. These layout patterns, or annotations, are manually created per site with an Annotation Editor. The work in [13] describes a framework to manually annotate the content w.r.t a schema representing the task a user wishes to accomplish. These annotation rules are also site specific. The ASTER system [26] of Raman, permits people who are blind to listen to both structured (LaTeX) and unstructured (ASCII text) documents, as well as manually define their own document reading rules (e.g. to skip all but the headings for a general overview). The work in [23] describes a manual annotation framework, based on contextual graphs, that allows for very flexible navigation between the segments in a page. The fundamental difference between these works and HearSay is the degree of automation in creating the annotations. Recall that, in HearSay, structural analysis is totally automatic while the ontology used in semantic analysis is built once for a domain. Thus, HearSay offers a higher degree of scalability over a large collection of Web sources and associated Web documents.

6. DISCUSSIONS

In this paper we described a technique for bookmarking semantic concepts in a Web page whose content is logically structured around a domain ontology. The rendition of this semantic structure into audio enables individuals who suffer from visual impairments to bookmark semantic concepts using audio. We presented preliminary experimental evidence of the efficiency of semantic bookmarking for repetitive browsing tasks.

There are a number of directions for extending and improving semantic bookmarks for non-visual Web access. One can generalize semantic bookmarks to more sophisticated forms of personalization. One such example is a voice-driven personal information assistant for automatically retrieving content from diverse forces including those requiring access to the deep Web. We can further enhance the scalability of HearSay by developing more automated techniques for identifying concepts in Web content via mining ontologies along the lines outlined in [3, 17].

Finally the presentation of data in tables and charts using audio is an emerging research topic [21, 23, 8]. It will be interesting to extend semantic bookmarks to include these kinds of complex HTML structures.

7. REFERENCES

- [1] C. Asakawa and T. Itoh. User interface of a home page reader. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.
- [2] D. Bolnick and G. Freed. New developments in web-based accessible multimedia. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1998.
- [3] H. Davulcu, S. Vadrevu, S. Nagarajan, and I. Ramakrishnan. Ontominer: Bootstrapping and populating ontologies from domain-specific web sites. *IEEE Intelligent Systems*, 18(5), 2003.
- [4] H. Davulcu, G. Yang, M. Kifer, and I. Ramakrishnan. Computational aspects of resilient data extraction from semistructured sources. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 136–144, 2000.
- [5] D. Dillin. Concept of a structural html page browser to support access to www for people with disabilities. *Technical Report, Yuri Rubinsky Insight Foundation*, 1997.
- [6] C. Earl and J. Leventhal. A survey of windows screen reader users: Recent improvements in accessibility. *Journal of Visual Impairment and Blindness*, 93(3), 1999.
- [7] T. Ebina, S. Igi, and T. Miyake. Fast web by using updated content extraction and a bookmark facility. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2000.
- [8] R. Filepp, J. Challenger, and D. Rosu. Improving the accessibility of aurally rendered html tables. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.
- [9] D. Geoffray. The internet through the eyes of windows-eyes. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1999.
- [10] J. Gunderson and R. Mendelson. Usability of world wide web browsers by persons with visual impairments. In *Proceedings of the RESNA Annual Conf.*, 1997.
- [11] D. Hadjadj and D. Burger. Braillesurf: An html browser for visually handicapped people. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1999.
- [12] P. Hendrix and M. Birkmire. Adapting web browsers for accessibility. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1998.
- [13] A. Huang and N. Sundaresan. A semantic transcoding system to adapt web services for users with disabilities. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2000.
- [14] P. Jones. Speech-enabled web access: an instructional case study. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1998.
- [15] L. Kraus. Teaching mathematics to students with physical disabilities using the world wide web. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1998.
- [16] D. Leu and D. Leu. *Teaching with the Internet: Lessons from the Classroom*. Christopher-Gordon Publishers, 1997.
- [17] A. Maedche and S. Staab. Learning ontologies for the semantic web. In *Workshop on the Semantic Web (SemWeb)*, 2001.
- [18] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.
- [19] S. Morley. Digital talking books on a pc: A usability evaluation of the prototype daisy playback software. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.
- [20] S. Mukherjee, G. Yang, and I. Ramakrishnan. Automatic annotation of content-rich html documents: Structural and semantic analysis. In *Intl. Semantic Web Conf. (ISWC)*, 2003.
- [21] T. Oogane and C. Asakawa. An interactive method for accessing tables in html. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.
- [22] E. Pontelli and T. C. Son. Planning, reasoning, and agents for

- non-visual navigation of tables and frames. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.
- [23] E. Pontelli, W. Xiong, D. Gillian, E. Saad, G. Gupta, and A. Karshmer. Navigation of html tables, frames, and xml fragments. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.
- [24] E. Pontelli, W. Xiong, G. Gupta, and A. Karshmer. A domain specific language framework for non-visual browsing of complex html structures. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2000.
- [25] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Intl. World Wide Web Conf. (WWW)*, 2004.
- [26] T. Raman. Audio system for technical readings. *PhD Thesis, Cornell University*, 1994.
- [27] C. Schmandt. Audio hallway: A virtual accoustic environment for browsing. In *Proceedings of UIST*, 1998.
- [28] H. Takagi and C. Asakawa. Transcoding proxy for nonvisual web access. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2000.
- [29] H. Takagi, C. Asakawa, K. Fukuda, and J. Maeda. Site-wide annotation: Reconstructing existing pages to be accessible. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2002.
- [30] G. Weber. Programming for usability in non-visual user interfaces. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.
- [31] M. Zajicek, C. Powell, and C. Reeves. A web navigation tool for the blind. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 1998.
- [32] M. Zajicek, C. Powell, and C. Reeves. Web search and orientation with brookestalk. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1999.