Combating Information Overload in Non-Visual Web Access Using Context

Jalal Mahmud Yevgen Borodin Dipanjan Das [*] I.V. Ramakrishnan

Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA
{jmahmud, borodin, ram}@cs.sunysb.edu

ABSTRACT

Web sites are designed for graphical mode of interaction. Sighted users can visually segment Web pages and quickly identify relevant information. On the contrary, individuals with visual disabilities have to use screen readers to browse the Web. As screen readers process pages sequentially and read through everything, Web browsing becomes time-consuming and strenuous. Although, the use of shortcut keys and searching offers some improvements, the problem still remains. In this paper, we address this problem using the notion of *context*. Our prototype system, CSurf, embodying our approach, provides all features of a usual screen reader. However, when a user follows a link, CSurf captures the context of the link, processes it with several NLP techniques, and uses it to identify relevant information on the next page. CSurf rearranges the content of the page, so, that the relevant information is read out first. We conducted a series experiments to evaluate the performance of CSurf against the state-of-the-art screen reader JAWS. Our results show that the use of context can save browsing time and substantially improve the browsing experience of visually disabled individuals.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*natural language, Voice I/O*; H.3.3 [**Information Systems**]: Search and Retrieval

General Terms

Algorithms, Design, Human Factors, Experimentation.

Keywords

Web Navigation, Context, Voice Browsing, Screen-Reader, CSurf, User Interface, Information Rearrangement.

1. INTRODUCTION

The Web has become an indispensable source of information. The primary mode of interaction with the Web is via graphical browsers, which are designed for visual interaction. As we browse the Web, we have to filter through a lot of irrelevant data (e.g banners, commercials, navigation bars). Sighted individuals can quickly segment any Web page and identify the information that is most relevant to them. The task becomes complicated for individuals with visual disabilities. Blind people use screen readers [5, 1] to browse the Web. However, screen readers process Web pages sequentially, and provide little or no content filtering, resulting in *information overload.* To address the problem of information overload in non-visual Web browsing, screen-readers often permit to skip blocks of text in the order they appear on the page. Unfortunately, in many cases, users still have to listen or skip through a substantial part of page content before they get to the information. To help users locate the information quicker, a number of screen readers [5, 1] allow keyword searching. However, simple searching has two problems: it works only for exact string matching and it disorients users in case of a wrong match. In both cases users have to start from the beginning of the page. The problem of *information overload* in non-visual Web access still remains. Is it possible to do better than that?

In this paper we describe a technique for context identification and information rearrangement, based on the structural and visual organization of Web pages. We present an algorithm and a system that will help blind users quickly identify relevant information while surfing the Web, thus, considerably reducing their browsing time.

The rest of this paper is organized as follows. In Section 2, we describe the architecture of CSurf, a prototype system that implements context-based browsing. In Section 3 we formalize the definition of context and describe our context processing algorithm. Testing and evaluation of the system appear in Section 4. Related work follows in Section 5. We conclude this paper in Section 6 by describing several directions of further research.

2. SYSTEM ARCHITECTURE

The architecture of CSurf, our context-based browsing system, is shown in Figure 1. Users communicate with the system through the **Interface Manager**. The module uses VoiceXML dialogs to interact with the users and present Web page content. The interface allows keyboard and voice input via our own VoiceXML interpreter [2], and provides both basic and extended screen-reader navigation features, such as shortcuts and corresponding voice commands.

[*]Department of Computer Science, Carnegie Mellon University, dipanjan@cs.cmu.edu

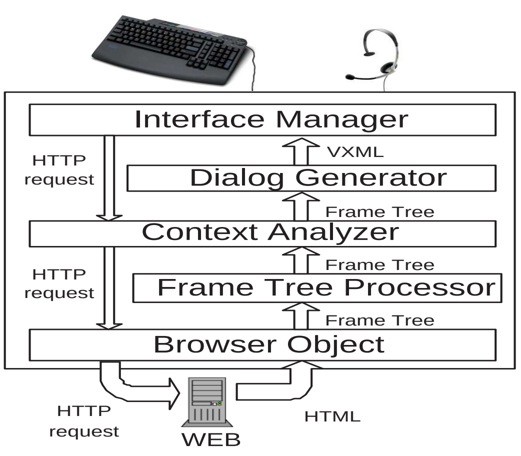


Figure 1: Architecture of CSurf

Context Analyzer is called twice for each Web page access. When the user follows a link, the module collects the context from the current page. When a new Web page is retrieved, the module executes our algorithm to contextualize the page before it is presented to the user.

The **Browser Object** module downloads Web content every time the user requests a new page to be retrieved. The module is built on top of the Mozilla Web Browser coupled with extended JREX Java API wrapper.

Frame Tree Processor extracts the *Frame Tree* representation of the Web page. A *Frame Tree* is a tree-like data structure that contains Web page content and formatting, specifying how a Web page has to be rendered. The module cleans, reorganizes, and partitions the frame tree. Subsequently, Context Analyzer reorders the frame tree before passing it to the Dialog Generator.

The **Dialog Generator** module uses a collection of dialog templates to convert the frame tree into a Voice-XML dialog. The latter is then delivered to the Interface Manager. Some more architectural details appear in [8].

3. CONTEXT ANALYSIS

In this section we formally define the notion of context and describe the phases of the context analysis algorithm.

Context: Given a frame tree of a Web page and a link with its corresponding tree node n_i, *context* is defined as a multiset that includes the text of the link node, along with the text contained in the m sibling nodes $\{n_1, ..., n_{i-1}, n_{i+1}, ..., n_m\}$ of n_i.

Consider an example when the source is the front page of *The New York Times* news Web site, Figure 2 (a). The context of the encircled link is the text surrounded by the dotted line. Figure 2 (b) shows the corresponding frame tree with the link and its siblings.

1. Context Identification and Ranking: The frame tree is searched for all nodes containing the URL selected by the user. The text in and around the link is extracted from the frame tree and stored in a multiset, after all function words have been removed.

We denote this multiset as $S_{context}$ and a member of this set as $W_{context}$. The *Porter's Stemming Algorithm* [11] is used to obtain stems for all words in $S_{context}$. The stems are stored in multiset S_{stem}. We denote the members of S_{stem} as W_{stem}. The words in $S_{context}$ are then ranked (weighted) according to their proximity to the link. Through a series of experiments, we chose the rank (weight) of each W_{stem} to be half the rank of the corresponding $W_{context}$. At this point, the destination Web page is fetched, Figure 3 (a).

2. NLP Based Context Matching: The algorithm matches the words in the multisets to the text in all leaves of the new frame tree, Figure 3 (b), which are then assigned respective weights. The steps are enumerated below:

a. Run a depth first search to identify all leaves $\{Leaf_1, ..., Leaf_n\}$ in the frame tree of the destination page.

b. For each $Leaf_i$:

   i. Set its weight W_i to 0.

   ii. If $Leaf_i$ contains text, tokenize and store the text in a corresponding $List_i$, after removing the function words. Apply *Porter's Stemming Algorithm* to each word in $List_i$ to get $StemList_i$.

   iii. Perform a search of the context keywords from the set $S_{context}$ in $List_i$.

   iv. For each successful context keyword match increment W_i by the weight of the keyword.

   v. Perform a search of the stemmed keywords from the set S_{stem} in $StemList_i$.

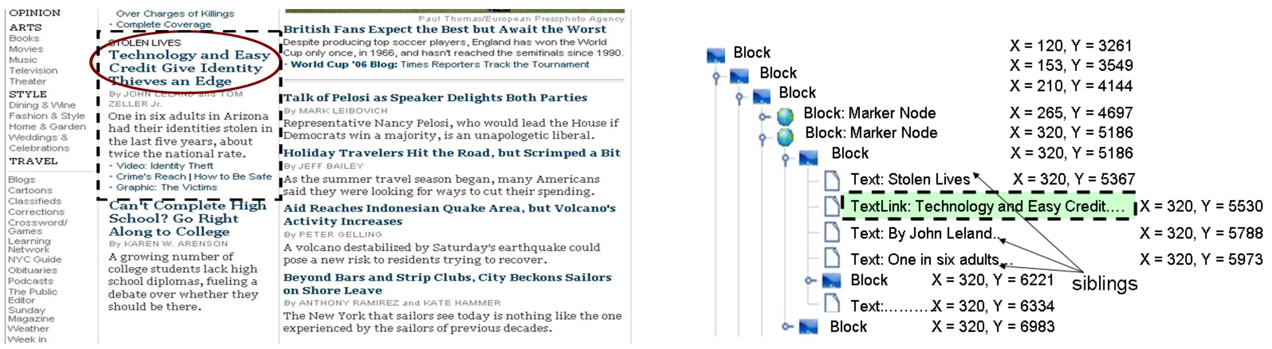
   vi. For each successful match of the stemmed keyword increment W_i by the weight of the stemmed keyword.

   vii. Compute *Lexical Similarity* (that uses inverse of *Edit Distance*) as described in [6] between each word of the set $S_{context}$ and the words in $List_i$. Multiply each similarity value by the weight of the context word and add this result to W_i.

   viii. Repeat the above step for each word of the set S_{stem} and the words in $StemList_i$.

Continuing our example in Figure 3, the frame tree of the destination page will be searched for all words occurring in the context of of the link from the source page. By the end of this step, the leaf nodes, marked with clear-box icons in Figure 3 (b), will have accumulated their weights. The weights $W : n$ express the relevance of each leaf-node with respect to the context gathered from the source Web page. Higher weight implies greater relevancy.

3. Block Ranking and Rearrangement: We propagate the weights from the leaves of the frame tree up to a certain node/block, which we identify as a parent of semantically related nodes. We define such node as *Marker Node*. To identify *Marker Node*, we initiate a depth first search from the root of the frame tree and recursively merge the geometrically aligned nodes. Web pages are usually organized in such a way that semantically related information is grouped together. Thus, the geometric alignment can help identify blocks containing semantically related information. For example, in Figure 2 (a), all Web page objects in the area surrounded by the dotted line have the same geometrical alignment and are otherwise related. The corresponding


OPINION
ARTS
Books
Movies
Music
Television
Theater
STYLE
Dining & Wine
Fashion & Style
Home & Garden
Weddings & Celebrations
TRAVEL
Blogs
Cartoons
Classifieds
Corrections
Crossword/ Games
Learning Network
NYC Guide
Obituaries
Podcasts
The Public Editor
Sunday Magazine
Weather
Week in

Over Charges of Killings
• Complete Coverage

STOLEN LIVES
Technology and Easy Credit Give Identity Thieves an Edge
By JOHN LELAND and TOM ZELLER Jr.
One in six adults in Arizona had their identities stolen in the last five years, about twice the national rate.
• Video: Identity Theft
• Crime's Reach | How to Be Safe
• Graphic: The Victims

Can't Complete High School? Go Right Along to College
By KAREN W. ARENSON
A growing number of college students lack high school diplomas, fueling a debate over whether they should be there.

Paul Thomas/European Pressphoto Agency
British Fans Expect the Best but Await the Worst
Despite producing top soccer players, England has won the World Cup only once, in 1966, and hasn't reached the semifinals since 1990.
• World Cup '06 Blog: Times Reporters Track the Tournament

Talk of Pelosi as Speaker Delights Both Parties
By MARK LEIBOVICH
Representative Nancy Pelosi, who would lead the House if Democrats win a majority, is an unapologetic liberal.

Holiday Travelers Hit the Road, but Scrimped a Bit
By JEFF BAILEY
As the summer travel season began, many Americans said they were looking for ways to cut their spending.

Aid Reaches Indonesian Quake Area, but Volcano's Activity Increases
By PETER GELLING
A volcano destabilized by Saturday's earthquake could pose a new risk to residents trying to recover.

Beyond Bars and Strip Clubs, City Beckons Sailors on Shore Leave
By ANTHONY RAMIREZ and KATE HAMMER
The New York that sailors see today is nothing like the one experienced by the sailors of previous decades.

Block X = 120, Y = 3261
Block X = 153, Y = 3549
Block X = 210, Y = 4144
Block: Marker Node X = 265, Y = 4697
Block: Marker Node X = 320, Y = 5186
Block X = 320, Y = 5186
Text: Stolen Lives X = 320, Y = 5367
TextLink: Technology and Easy Credit.... X = 320, Y = 5530
Text: By John Leland.. X = 320, Y = 5788
Text: One in six adults... X = 320, Y = 5973
Block X = 320, Y = 6221
Text:......... X = 320, Y = 6334
Block X = 320, Y = 6983
siblings

(a) Source Page (b) Source Frame Tree

Figure 2: Context Identification and Ranking

node will be identified as a marker node in frame tree. In Figure 2 (b) marker nodes are denoted by circular icons.

The following steps describe the Block Ranking procedure:

a. Starting with one level above the leaf nodes, propagate the weights of the leaves up in the frame tree.

b. For a block node m, having n children, calculate its weight:

$$W_m = \sum_{i=1}^{n} (n - i + 1) * W_{child_i}$$

where $i = 1, ..., n$, and W_{child_i} denotes the weight of the ith child of the block node m.

c. Do not propagate the weights beyond the marker nodes.

d. Store all marker nodes in a list for further processing.

We observed that the first node in any block is usually more important than the subsequent nodes. We use this observation in multiplying the weight W_{child_i} of each child node in a block m by $(n-i+1)$. Thus, with each new child node, we reduce its contribution to the total weight of the block m. Then, the frame tree is reorganized based on the weights of the marker nodes, so that the most relevant block of information is placed first. The marker node, expanded in Figure 3 (b), happens to have the most weight, which makes it the most relevant block. The section of the Web page, corresponding to the selected marker node, is shown in Figure 3 (a). The frame tree is rearranged in such a way, that the Interface Manager will first read the most relevant block, which, in our case, is the article about the identity theft. Having finished with the article, the Interface Manager will continue reading the rest of the Web page content.

4. PERFORMANCE

To measure the performance of our system, we experimentally compared the CSurf Web browser against the state-of-the-art JAWS screen-reader. In our experiments we also conducted preliminary qualitative evaluation of CSurf with blind and low-vision users at Helen Keller Services for the Blind (HKSB), Hempstead, NY. The evaluators were experienced computer users proficient with JAWS.

In our testing, we used twenty four Web sites spanning four content domains: *news*, *books*, *consumer electronics*, and *office supplies*. We did 5 navigations on each Web site. The performance testers were asked to measure the total time[1] taken to reach the relevant information using our browser. For baseline comparison, they were also asked to carry out the same experiments with the JAWS screen-reader using the same set of Web pages. We allowed the use of JAWS shortcut keys to skip blocks of text and accelerate browsing.

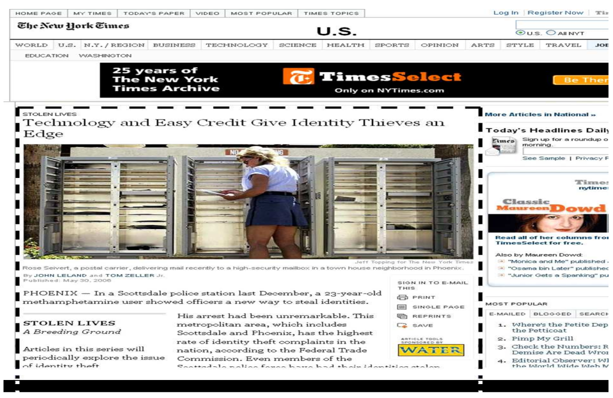
Following the testing, we had CSurf evaluated at HKSB. The evaluators noted that context-based browsing, although not always accurate, is a substantial improvement over regular browsing using screen-readers. The evaluation demonstrated that the use of context can save browsing time and substantially improve browsing experience of visually disabled Internet users. The instructors at HKSB are ready to switch over to CSurf for Web browsing and start teaching their students to use it, as soon as we have a stable version implementing all major functionalities of JAWS screen-

Table 1: CSurf Performance.

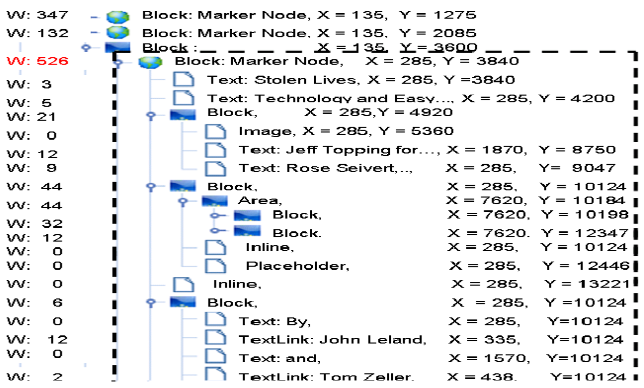
Web Sites	Time Taken (using)			
	CSurf		JAWS	
	μ	σ	μ	σ
NYTimes	25.2	2.4	90.5	5.2
LATimes	21.4	1.3	100.8	6.5
GoogleNews	38.6	2.1	112.5	5.4
YahooNews	42.7	1.1	120.2	7.1
CNN	36.8	1.7	106.8	4.5
BBCNews	34.5	2.3	92.4	6.3
Bloomberg News	41.5	2.9	106.4	3.5
Washington Post	56.9	2.5	90.5	5.2
Amazon	50.4	10.2	145.5	14.4
BN	70.2	13.23	161.5	21.42
AbeBooks	88.2	23.45	182.1	31.12
Khazana	74.4	31.5	211.4	22.5
BlackWell	90.4	21.3	172.1	23.4
OfficeMAX	101.6	23.6	200.2	31.52
OfficeDepot	89.9	15.8	184	25.23
QuillCorp	82.6	19.6	195	31.84
Walmart	106.1	13.7	180	20.2
Shop	91.5	10.4	230	26.13
Staples	75.5	14.5	210.4	36.3
BestBuy	90.5	20.7	190	36.7
Buy	91.3	21.2	176.7	20.5
CompUSA	111.3	20.4	186.78	21.5
BizRate	114.2	23.2	179.8	25.3
ecost	115.3	24.1	220.2	26.1

*All times are in seconds .

[1]Here *total time* represents the period that starts when the user follows a link and ends when our system begins to read the relevant information on the next Web page.


The New York Times
U.S.
25 years of The New York Times Archive
TimesSelect
Technology and Easy Credit Give Identity Thieves an Edge
STOLEN LIVES
A Breeding Ground
More Articles in National »
Classic Maureen Dowd
MOST POPULAR
WATER

(a) Destination Page


```
W: 347   Block: Marker Node, X = 135,  Y = 1275
W: 132   Block: Marker Node. X = 135.  Y = 2085
         Block :                X = 135,  Y = 3600
W: 526     Block: Marker Node,    X = 285, Y = 3840
W:  3        Text: Stolen Lives, X = 285, Y =3840
W:  5        Text: Technology and Easy..., X = 285, Y = 4200
W: 21        Block,         X = 285,Y = 4920
W:  0          Image, X = 285, Y = 5360
W: 12          Text: Jeff Topping for..., X = 1870,  Y = 8750
W:  9          Text: Rose Seivert,..,     X = 285,   Y=  9047
W: 44        Block,                       X = 285,   Y = 10124
W: 44          Area,                      X = 7620,  Y = 10184
W: 32            Block,                   X = 7620,  Y = 10198
W: 12            Block.                   X = 7620.  Y = 12347
W:  0          Inline,                    X = 285,   Y = 10124
W:  0          Placeholder,               X = 285,   Y = 12446
W:  0        Inline,                      X = 285,   Y = 13221
W:  6        Block,                       X  = 285,  Y =10124
W:  0          Text: By,                  X = 285,    Y=10124
W: 12          TextLink: John Leland,     X = 335,    Y=10124
W:  0          Text: and,                 X = 1570,   Y=10124
W:  2          TextLink: Tom Zeller.      X = 438.    Y=10124
```

(b) Destination Frame Tree

Figure 3: Most Relevant Block Identification

reader. HKSB have also agreed to provide their power users with visual disabilities for pre-release beta testing.

5. RELATED WORK

The work described in this paper has broad connections to research in non-visual Web access, information rearrangement in Web pages, and contextual analysis.

Blind users access the Web using screen-readers, such as JAWS [5] and IBM's Home Page Reader [1]. BrookesTalk [12] makes summaries of Web pages to address the *information overload* problem in *non-visual Web access*. CSurf departs from these systems in scope and approach. Specifically, it helps to find the relevant information quickly on on following a link. *Information Rearrangement in Web Pages* typically relies either on rules [9] or logical structures [10]. Our system automatically captures contextual information and re-arranges the content. *Contextual Analysis* for non-visual Web navigation is not a well-studied problem. The system in [3] uses the context of a link to get the preview of the next Web page before following a link. In contrast, we aim to help visually disabled users quickly identify relevant information *after* following a link. A poster with our initial ideas on context-based browsing and some preliminary testing [7] appears in the proceedings of ASSETS2006. Here, we extend our preliminary work and give a detailed context algorithm improved with several NLP techniques. We also report evaluation at Helen Keller Services for the Blind [4].

6. CONCLUSION AND FUTURE WORK

In this paper, we have described the design and implementation of CSurf, a context-directed non-visual Web browsing system. We presented the results of our quantitative performance evaluation and the qualitative evaluation at the Helen Keller Services for the Blind. The results show that context-based browsing can reduce information overload in non-visual Web browsing. A massive beta testing and evaluation will be performed following the first official release of our voice browser. We also identify several potentially useful areas for further research. If search keywords are treated as context, our algorithm can be easily extended to allow smart searching within a Web page. We are currently researching summarization techniques to further save the browsing time. We are also investigating the feasibility of applying machine learning algorithms and statistical models to context identification and ranking.

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] Y. Borodin. A flexible vxml interpreter for non-visual web access. In *ACM Conf. on Assistive Technologies (ASSETS)*, 2006.

[3] S. Harper, C. Goble, R. Stevens, and Y. Yesilada. Middleware to expand context and preview in hypertext. In *Assets '04: Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, 2004.

[4] `http://www.hellenkeller.org`.

[5] `http://www.freedomscientific.com/`.

[6] D. Lin. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, 1998.

[7] J. Mahmud, Y. Borodin, D. Das, and I. Ramakrishnan. Improving non-visual web access using context. In *ASSETS*, 2006.

[8] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[9] T. Raman. Audio system for technical readings. *PhD Thesis, Cornell University*, 1994.

[10] H. Takagi and C. Asakawa. Transcoding proxy for nonvisual web access. In *ACM Intl. Conf. on Assistive Technologies (ASSETS)*, 2000.

[11] C. van Rijsbergen, S. Robertson, and M. Porter. *New models in probabilistic information retrieval.* British Library, 1980.

[12] M. Zajicek, C. Powell, and C. Reeves. Web search and orientation with brookestalk. In *Proceedings of Tech. and Persons with Disabilities Conf.*, 1999.