A Comparative Web Browser (CWB) for Browsing and Comparing Web Pages

Akiyo Nadamoto
Communications Research Laboratory
Keihanna Human Info-Communication Research Center
Hikaridai, Seikachyo, Kyoto, Japan
nadamoto@crl.go.jp

Katsumi Tanaka
Kyoto University
Department of Social Informatics,
Graduate School of Informatics
Yoshida Honmachi, Sakyo-ku, Kyoto, Japan
ktanaka@i.kyoto-u.ac.jp

ABSTRACT

In this paper, we propose a new type of Web browser, called the *Comparative Web Browser* (CWB), which concurrently presents multiple Web pages in a way that enables the content of the Web pages to be automatically synchronized. The ability to view multiple Web pages at one time is useful when we wish to make a comparison on the Web, such as when we compare similar products or news articles from different newspapers. The CWB is characterized by (1) automatic content-based retrieval of passages from another Web page based on a passage of the Web page the user is reading, and (2) automatic transformation of a user's behavior (scrolling, clicking, or moving backward or forward) on a Web page into a series of behaviors on the other Web pages. The CWB tries to concurrently present "similar" passages from different Web pages, and for this purpose our CWB automatically navigates Web pages that contain passages similar to those of the initial Web page. Furthermore, we propose an enhancement to the CWB, which enables it to use linkage information to find related documents based on link structure.

Categories and Subject Descriptors

H.5.2 [**User Interface**]: Windowing Systems; H.5.2 [**User Interface**]: Prototyping; I.7.m [**Document and Text Processing**]: Miscellaneous

General Terms

Design, Documentation

Keywords

comparison, Web browser, content synchronization, passage retrieval

1. INTRODUCTION

The Internet continues to grow rapidly, and we now have access to more than 36 million Web sites on the Internet [1]. Many Web sites consist of a vast volume of pages; for example, it is common for a Web site to consist of more than 10,000 Web pages. These Web sites can be classified into similar categories, such as news sites, e-commerce sites, university sites, etc.

The need to view multiple Web pages at one time often arises when we wish to make a comparison on the Web, such as when comparing similar products or news articles from different newspapers. Comparative viewing of multiple Web pages tends to be a tedious task for users since they must open and manipulate a different window for each Web page. A user's scrolling/clicking operations on these windows are independent of each other. For example, suppose that a user wants to buy a notebook PC and wishes to examine two candidates from different vendors. To compare their prices and performance data, for each PC the user will open a Web page containing the price and the performance data. As the number of items to be compared rises, the task becomes increasingly tedious since the *content-synchronization* has to be done manually. Another way to compare data on the Web is to use Web sites that provide comparison services [2][3],in which several items are already arranged for comparative viewing. Unfortunately, such comparison-service sites do not cover the entire range of available items and the user still has to navigate through the comparison information manually.

In this paper, we propose a new type of Web browser, called the *Comparative Web Browser* (CWB), which presents multiple Web pages concurrently in a way that enables the contents of those pages to be automatically synchronized. The CWB can present user-specified Web pages from different Web sites at the same time for comparison (Figure 1). Users can interact with either of the two pages, and any interaction with one of the pages is automatically transformed into a series of interactions with the other page (though not necessarily the same action). In this way, *content-synchronization* between the two pages is achieved. This *content-synchronization* means that the contents of the windows remain similar as the user moves through the information of interest. For example, if a user scrolls through a Web page to see the price of a notebook PC, the second Web page is automatically scrolled to show the price information for the other PC. That is, provided the second Web page contains price information, the CWB will automatically search and navigate through the page to obtain the information. In this case, the interaction (scrolling) for the second page corresponds to a series of different interactions than occurred regarding the initial Web page (i.e., navigating to a different page and scrolling through it).

The *content-synchronization* of the CWB is based on the relevant passage retrieval of a Web page. First, the user identifies a passage of interest from a Web page. Second, the CWB searches for the most similar passage in the other Web page. The first step is done by identifying a passage that appears in the middle area of a browser window. The second step is done through a passage similarity search. This step also includes link navigation from the original Web page or passage retrieval using a pre-generated index of the target Web site pages. Users can freely interact with Web

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.

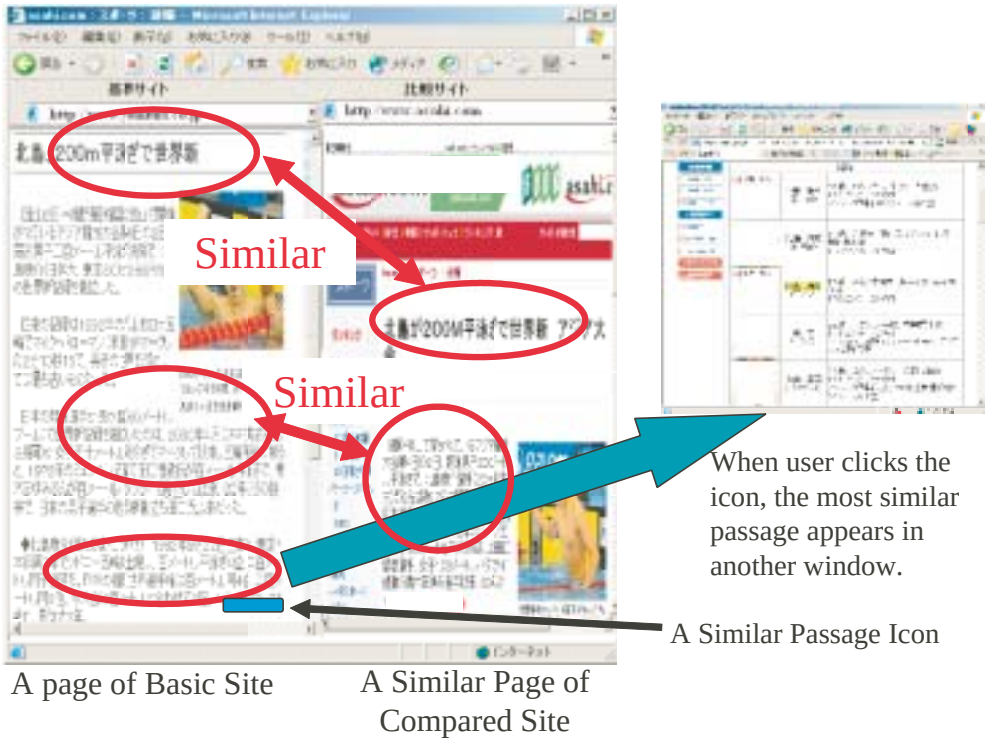


Figure 1: Comparative Web Browsing

pages belonging to a Web site by moving backward or forward, clicking, or scrolling. The CWB automatically generates *content-synchronized navigation* of the other Web site based on the user's behavior.

The CWB is characterized by (1) automatic content-based retrieval of passages from the other Web page based on a passage of the Web page the user is reading, and (2) automatic transformation of the user's behavior (scrolling, clicking, moving backward or forward) on a Web page into a series of behaviors on the other Web pages. Note that a user's behavior (e.g., scrolling) will not necessarily correspond to the same operation on the other Web page; for example, link navigation from a page of the Web site is transformed into searching and presenting the most similar page on the other Web site. This is because our CWB tries to concurrently present "similar" passages from different Web pages, and to do this it must automatically navigate through Web pages to find passages similar to the original passage.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 explains the comparative Web browsing concept, Section 4 discusses the CWB system, Section 5 discusses our experiments, Section 6 explains the link-based CWB for our future work, and we conclude in Section 7.

2. RELATED WORK

To enable comparative browsing of multiple Web sites, Liu et al. [4], [5] recently proposed visualization methods for Web site comparison. For given competitive Web sites, their system merges all the pages of the sites, and then clusters the pages hierarchically according to the feature vectors of the pages. The system presents the clustering result in a tree form, in which the Web site that each page belongs to is indicated by the node color. Their system is useful for browsing through the entire contents of competitive Web sites and grasping the difference between the contents. While our goal is similar to theirs, we took a different approach that is based on the notion of *concurrent navigation of multiple sites with content-synchronization*. For one Web site, users can freely navigate pages according to their particular interest, while similar pages from a second Web site are dynamically retrieved and presented to the user concurrently.

To determine the differences between two Web pages, Seung-Jin et al. [6] proposed the Semantic Change Detection (SCD) algorithm for detecting the *semantic changes* between two bodies of HTML data. Their approach is to transform the HTML data from the two sources into trees and remove the common edges from the two trees. This algorithm aims mainly at detecting important updates made to a single Web page. In contrast, we focus in this paper on the similarities between two Web sites, especially on how the similarities (and differences) can be effectively presented page by page.

Viewing the difference between Web pages, Chen et al. [7] proposed a new HTML differencing tool called TopBlend. TopBlend uses the fast Jacobson-Vo algorithm for page comparison. It finds differentiation of history of Web pages, but our system finds differentiation of passage of Web pages in different Web sites. The TopBlend has two viewing systems, one is the merged view, the other is the two-frame view. The two-frame view looks like the CWB's interface, but it is different from our system to browse only a list of differentiation of two pages, and also two windows' operation is not synchronized.

As a means to find a common structure and generate a schema from multiple semi-structured data, Goldman and Widom [8] proposed the DataGuide algorithm. Multiple semi-structured data is merged in an Object Exchange Model (OEM) graph, and common edges are merged to generate a summary of the multiple semi-

structured data. The DataGuide algorithm makes it possible to grasp the common structure of multiple semi-structured data, but it is still insufficient to show the similarity (and differences) among the contents of Web pages.

For commercial purposes, much attention has been directed towards services providing comparison evaluation information on Web sites. Gomez [2], for example, provides users with comparison information regarding Web sites. The information is collected onto a score card and matched to user needs. In the area of e-commerce, several systems have been developed to extract product price information from the Web, and to provide a comparison service regarding product prices. For example, Doorenbos et al. [9] developed the shopbot price comparison system that is used in Excite. These services, though, are basically manually operated.

Dean and Henzinger [10] proposed a Web-page-based search system where the input to the system search process is the URL of a page rather than a set of query terms. Their algorithms to find related web pages use only the connectivity information from the Web (i.e., the links between pages), not the page content or usage information. In our system, the passage is a basic unit of retrieval. For one Web site, when a page is presented in the browser window, the passage positioned in the middle area of the window is regarded as a query, and similarity-based retrieval is done for the other Web site. Currently, our similarity search for pages or passages is done using the vector space model and passage-feature vectors.

Oyama and Tanaka [11] proposed a topic-structure-based search technique for Web similarity searching. A collection of query keywords is automatically transformed into a hierarchical structure in which each parent node denotes a theme and the child nodes denote the content of the theme. Through the role of the keywords in the structure, the system finds a page in which a title contains the thematic keywords and the text under the title contains the content keywords. This technique is used in our system to extract the feature vector of each page.

3. BASIC CONCEPTS OF COMPARATIVE WEB BROWSING

The notion of the *comparative Web browsing* consists of the following constructs (also see Figure 1).

- **Basic and Compared Web sites**
  We assume that a user wishes to browse two Web sites, say X and Y. The user actually interacts with one Web site, which we call the *basic Web site* X. The interaction with the other Web site (Y) is automatically simulated by the system. We call Web site Y the *compared Web site*.

- **Browsing Multiple Pages Simultaneously**
  Our system basically presents two (or more) pages (one from X and the other from Y) concurrently. This allows a user to browse the contents of the two pages and compare them.

- **Content-based Synchronization**
  Whenever a page of the basic Web site is presented in the browser window, the system automatically presents a similar page from the compared Web site in the same browser. This is called *content-based synchronization.*

- **Transformation of Interactions on One Site**
  Basically, each user interaction (scrolling, clicking, navigation, moving forward or backward, etc.) is done on the basic Web site, and is automatically transformed into an interaction (or a series of interactions) on the compared Web site such that the content-base synchronization is achieved. For example, link navigation from a page of the basic Web site is transformed into searching and presenting the most similar page on the compared Web site. Note that users generally don't have to operate the compared Web site in any way.

Suppose that the user browses a page x_i of Web site X. The system automatically searches for the page y_j of Web site Y that is most similar to page x_i. When the user scrolls through page x_i and a passage (e.g., a paragraph) appears in the middle part of the browser window, the system automatically scrolls the window for page y_j so that the most similar passage appears in the middle area of the corresponding window. In this way, we propose the CWB presents multiple similar Web pages concurrently.

Figure 2 shows an example illustrating the concept of comparative Web browsing.

Anchor clicking

Figure 2(a) shows two pages, one a page from a Japanese news site (e.g., the Yomiuri site [13]) in the left sub-window, and the other a page from another Japanese news site (e.g., the Asahi site [12]) shown in the right sub-window. Here, the Yomiuri site is selected as the basic Web site, and the Asahi site is the compared Web site. When the user clicks an anchor in the left page, a new page is shown and the most similar page from the other site is also shown in the right sub-window. As a result, in this example similar news articles are shown. This transformation is *page-level content synchronization*.

Scrolling a page

Each page consists of multiple passages (paragraphs). When we browse through a long Web page, we usually scroll up or down the page to read it. The CWB is also capable of *passage-level content synchronization*. That is, for the passage that appears in the center of the basic Web site sub-window, the system automatically searches for the most similar passage in the compared web site. If the most similar passage is contained within the original page, the system automatically scrolls through that page so that the most similar passage will appear in the right window. Figure 3 shows this automatic scrolling. Suppose paragraph B and paragraph 3 are similar. When the user scrolls through the basic page so that paragraph B is shown in the center of the window, the system scrolls through the right page so that paragraph 3 automatically comes to the center of window. (Figure 2(b) shows an example of the automatic scrolling of the compared site page.) When there is no similar passage, the CWB tries to retrieve similar passages from other pages.

Forward and Backward Navigation

When the user navigates forward or backward in the basic Web site sub-window, our system displays the next (or previous) page of the basic Web site, and also shows the most similar page in the compared Web site (see Figure 2(c)).

Highlighting Common Words

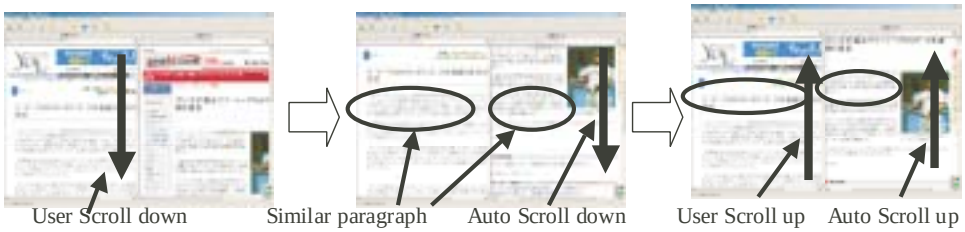
When using the CWB to browse through two pages, the user could find it difficult to find the corresponding paragraphs. To avoid this problem, the user can select a word in the basic page, and our system will then highlight all appearances of that word in the compared page; this is much like the highlighting of words in the Google cache (see Figure 2(d)).

4. THE COMPARATIVE WEB BROWSER SYSTEM

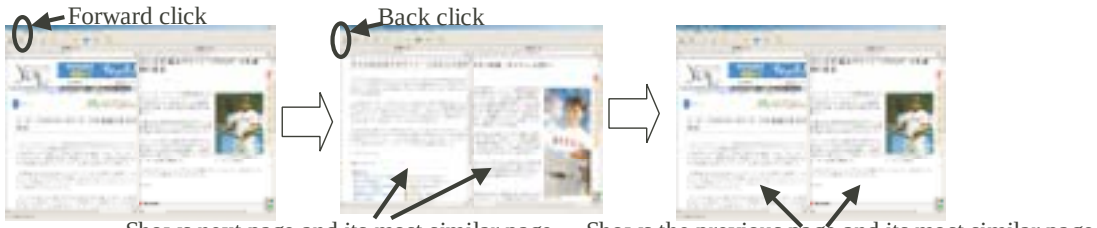
4.1 System Overview

The CWB identifies search keywords in the basic page in the basic Web site, and then automatically searches for a similar page and passage in the compared Web site by using a passage-based similar-

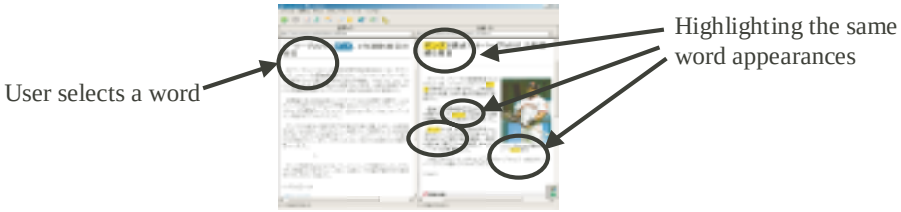
(a) Anchor clicking with content-synchronization



(b) Scrolling a page with content-synchronization



(c) Forward and Backward Navigation



(d) Highlighting Common Words

Figure 2: Example of the Interface

ity search. The CWB uses an index database, which is made from all the Web pages of multiple Web sites, in the server. When the user interacts with the basic Web site's window, the CWB operates the compared Web site's window synchronously and concurrently. Figure 4 shows our CWB system architecture.

The CWB system overview is as follows:

1. In the preprocessing phase, the CWB automatically extracts search keywords and creates a keywords database with regard to user-specified Web sites.
2. The user specifies the URLs of the basic Web site and the compared Web site, and the CWB shows the top page of each Web site. At this time, the two sites are assumed to be categorized into similar fields.
3. The user clicks an anchor or scrolls in the basic page.
4. When the user clicks an anchor of the basic page, the CWB calculates the degree of similarity of each Web page in the compared Web site for the new basic page. Then, it finds the most similar page from the comparison Web site, and shows it in the compared page window.
5. When the user scrolls the basic page, the CWB identifies a paragraph that appears in the middle area of the basic page window. Then, the CWB searches the most similar paragraph in the compared page, and scrolls automatically the compared page so that the most similar paragraph may appear in the middle area of the compared page window. If the CWB fails to find a similar paragraph, it searches for a page that has a paragraph with high similarity from the compared Web site .
6. Repeat steps (3) to (5).

In Figure 1, the user has specified two news sites as the basic and compared Web sites. The left side shows the basic page from news

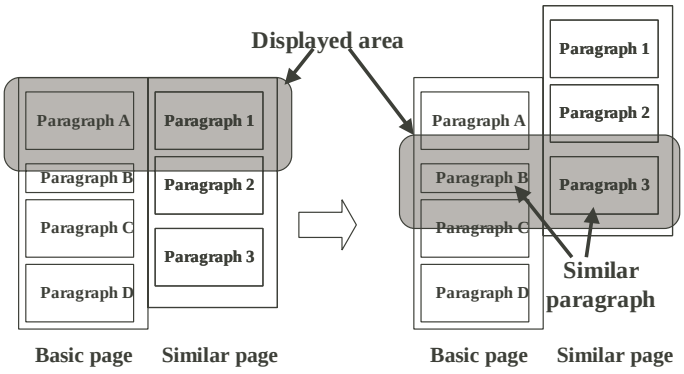


Figure 3: Scrolling a Page with Passage-level Content Synchronization

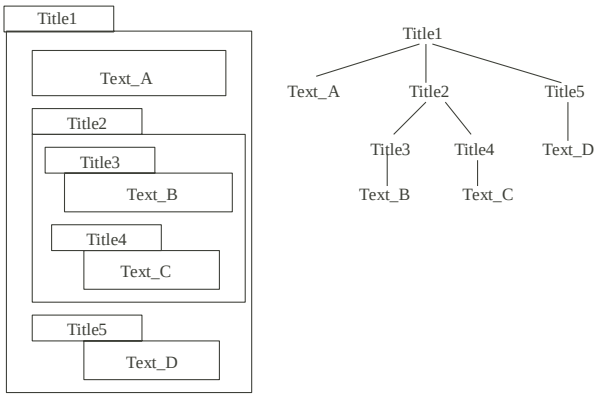


Figure 5: Structure of a Web Page

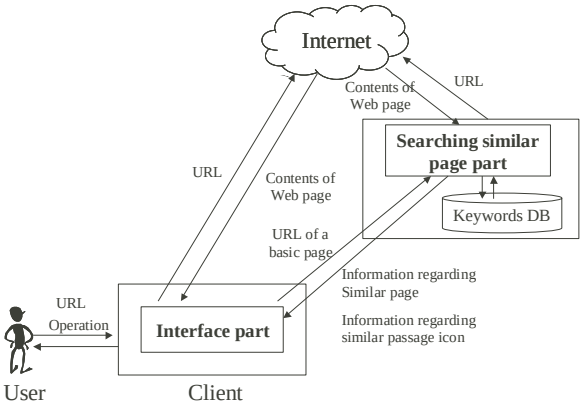


Figure 4: System Architecture of the Content-Based CWB

site A, while the right side shows a similar page from news site B. When the user goes to a news page in news site A describing a world record in swimming, the CWB gets feature vector information of that page from the keywords database. Next, the CWB searches for a similar news page in news site B that refers to the world record in swimming. The CWB concurrently displays the page from news site A in the left window and the news page from news site B in the right window. However, the page from news site A also contains a description of the career of the athlete who set the world record, but a similar description of the athlete's career is on a different page in news site B. In this case, the CWB puts the similar passage icon under the content regarding the athlete's career in the original news page from news site A. When the user clicks the icon, the CWB opens another window and browses through the page containing the content regarding the athlete's career from news site B.

Content that is irrelevant for comparing Web pages will often be included in the pages. News sites, for example, include many advertisements and similar frame content. We use a pattern data file, which contains a wide variety of Web site patterns, to filter out such unnecessary content. If the user specifies a Web page that doesn't have a pattern data file, the CWB evaluates the entire contents of that page.

4.2 Creating the Keywords Database

As preprocessing, the CWB extracts search keywords in advance at the server side and creates the keywords database because the CWB has to be able to do the similarity-based retrieval in real time. The keywords database has a paragraph table, a subject keyword table, a content keyword table, and a nouns and proper nouns table for each Web page of each Web site. If there is no database for a user's specified Web site, the CWB updates the keywords database in real time. A database for a Web site which has about 1000 Web pages can be created in about 3 minutes.

In this section, we explain how keywords from Web pages can be found to create the keywords database.

Oyama and Tanaka [11] proposed a way to classify user-specified keywords into subject keywords and content keywords. We use this method to extract the subject and content keywords from Web pages. A Web page typically includes a title and several paragraphs. These paragraphs often have subtitles indicating the nature of their content. In other words, many Web pages have a hierarchical structure (see Figure 5). We extract the search keywords in a Web page by following the hierarchical structure of the page. The words included in a title or subtitle become subject keywords, because these words are reliable indicators as to the nature of the content. The words in the content headed by a title or subtitle become content keywords.

The procedure to extract search keywords is as follows:

1. Make a tree structure of a Web page by using structure tags. These structure tags - Hn, P, $BLOCKQUOTE$, DIV, VL, OL, DL, and $TABLE$ - are used to create the Web structure. These structure tags separate the content into paragraphs.

2. Compute the word frequency (tf) of nouns and proper nouns in a Web page.

3. Compute word vectors based on word weights.
   If all nouns and proper nouns have the same weight, numbers and numerical classifiers will have high weights. The weights we assigned to each part of speech in our experiment are shown in Table 1. The word vector is equal to the word frequency multiplied by the word weight.

4. Finding a title and subtitles.
   We don't use a $< title >$ tag to find a title and subtitles, because a $< title >$ tag isn't always a title content in a Web page. For example, the Asahi news site has the same sentence in a $< title >$ tag in each page. A title or subtitle consists of words or a sentence enclosed by structure tags.

Table 1: Weight given to Nouns

part of speech	weight
proper nouns	3.0
number	0.1
numerical classifier	0.1
general nouns	1.0
other nouns	0.9

Furthermore, titles are written in larger characters than the other content of a Web page and/or the characters are emphasized. A word or sentence enclosed by a $< Font >$tag or a $< H >$tag, where the last word is a noun or proper noun, is considered a title candidate. A title is generally found at the top of a Web page and is the shallowest node and furthest left node in the tree structure. Subtitles are titles candidate other than the title. Title and subtitles have a nest structure (see Figure 5). In the case of Figure 5, Title1 is a title, and from Title2 to Title5 are subtitles.

5. Fixing subject keywords.
   Subject keywords are nouns and proper nouns from a title or subtitle. The CWB searches for subject keywords through a breadth-first search of the tree structure. If all words in a title or subtitle are search keywords, too many subject keywords will be generated. That is, the maximum number of subject keywords is fixed depending on their word vectors that exceeds the threshold (α). The title keywords (T_i) and subtitle keywords (STx_k) are considered the subject keywords ($inTitle$), where i is the number of title keywords, x is the number of subtitle keywords, and k is the number of subject keywords. $inTitle$ is defined as
   $inTitle = (T_i, ST1_j, \cdots, STx_k)$

6. Fixing content keywords
   Sentences other than those in a title or subtitles make up the content of a Web page, and contain the content keywords. That is, content keywords $inText_i, i \in \{1, \cdots, n\}$ are obtained from each paragraph of the basic page. The CWB then searches for content in the compared Web site that is similar to each part of the basic page. $inText_i$ are nouns and proper nouns whose word vectors exceed a threshold (β). If the word vectors of $C_i, i = 1, 2, ...n$ exceed β, $inText_i$ is defined as
   $inText_i = (C_0, C_1, ..., C_n)$
   $inText_i$ is ranked from the highest word frequency to the lowest.

For the case shown in Figure 5, T_i would be words contained in Title1, and STx_j would be words contained in Title2 to Title5. $inText_1$ would be words contained in Text_A, $inText_2$ the words in Text_B, $inText_3$ the words in Text_C, and $inText_4$ the words in Text_D.

In this way, the CWB obtains subject and content keywords from each Web page and creates or updates the keywords database.

4.3 Searching Similar Pages and Similar Passage Pages

The CWB searches a similar page from the compared Web site by using the passage-level feature vectors consisting of the subject keywords and content keywords. To synchronize the operation on two browser windows, the CWB computes the similarity-degree at a passage-level in the Web pages. The similarity-degrees between passages are computed using the Euclidian distance between the keyword feature vectors. For the CWB, the meaning of a passage is a paragraph, and a similar page is considered the one with the most similar paragraphs.

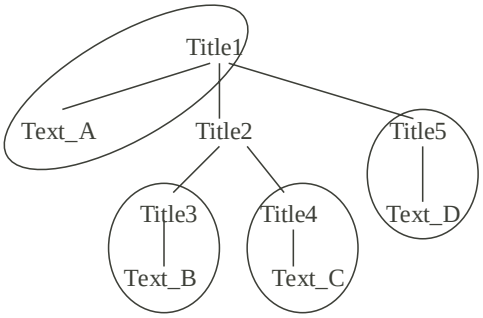


Figure 6: Web Page with Subtitles

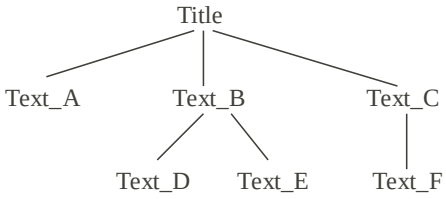


Figure 7: Web Page without Subtitles

A paragraph of a Web page is a node of the Web page tree structure of the Web page, and we search for a particular paragraph node of the tree structure. The CWB searches for similar titles and subtitles by using the subject keyword vector, and searches for similar content by using the content keywords vector, respectively. A Web page without subtitles, however, differs significantly from one with subtitles, and the CWB searches differently for each case.

Case 1. Web page having subtitles

In this case, the Web page is structured in such a way that child nodes contain the content under each title or subtitle (see Figure 6), and a subtitle and its child node can be treated together as a single entity.

We search for similar passages as follows:

1. Searching for a similar title and/or similar subtitles in the compared Web site.
   The CWB computes the similarity-degrees of the title and/or subtitles through a breadth-first search because the title and subtitles are within a nested structure. If the similarity-degree of a title and/or subtitles is higher than the threshold (γ), the title and/or subtitles are regarded a similar title and/or similar subtitles, and the contents of the title and subtitles are considered similar contents. That is, the contents become similar paragraphs.

2. Searching for similar contents in the compared Web site.
   The CWB computes the similarity-degree of content by using a subject keyword vector for the contents, except for the similar title and subtitle. If the similarity-degree of content is higher than the threshold (γ), the content is considered a similar paragraph.

Table 2: Relevance Ratio for Each Number of Keywords (precision %)

number of Subject keywords	number of Content keywords	Relevance ratio
5	7	32
	10	37
	15	30
	20	32
7	7	37
	10	52
	15	58
	20	52
10	7	52
	10	42
	15	30
	20	31

Case 2. Web page without subtitles
In this case, the Web page is considered a non-structured Web page (Figure 7). The root node is a title, and all other nodes contain content. The CWB computes the similarity-degrees between all contents by using a subject keyword vector, except for the root node.

In this way, the CWB searches for similar paragraphs in the Web pages of the compared Web site, and the Web page that has the greatest number of similar paragraphs becomes the similar page. If multiple Web pages are found as similar-page candidates, the CWB selects the page with the shallowest node and the left-most node in the link tree of the compared Web site as the similar page.

There are many cases, however, where similar contents on a basic Web page extend over multiple Web pages in the compared Web site. In this case, after the CWB determines the similar page, it estimates the difference between the basic page's paragraph and the similar page's paragraph. If the basic page's paragraph has no similar paragraph in the similar page, the CWB searches for a similar paragraph in the other Web pages of the compared Web site. A Web page, which has contents different than those of the basic page and similar page, then becomes the similar passage page. If multiple Web pages are candidates to become the similar passage page, the one with the shallowest node and farthest left node in the link tree of the compared Web site is selected as the similar passage page.

5. EXPERIMENTS

We did two types of experiments, one concerning the number of subject keywords and content keywords, and the other considering the precision ratio when finding similar pages.

The number of subject keywords and content keywords
The CWB extracts subject keywords and content keywords in the basic Web page, and creates the keywords database. The number of keywords affects the searching of similar pages, because the CWB searches similar Web pages by using search keywords in the keywords database. If there are too many or too few keywords, the correct answer rate of the search results will drop. It is thus important to determine the optimal number of keywords. In our experiment regarding the number of subject and content keywords, we used 1000 pages from the Asahi newspaper site [12] and the Yomiuri newspaper site [13]. We did the experiment by changing the subject keywords or content keywords. The results of the experiment are shown in Table 2.

Table 3: Precision Ratio (%)

Base site	Comparison site	Precision ratio
Asahi	Yomiuri	58
Yomiuri	Asahi	60
Asahi	Mainichi	51
Mainichi	Asahi	50
Yomiuri	Mainichi	62
Mainichi	Yomiuri	56

Our results indicate that the best number of subject keywords is 7 and the best number of content keywords is 15.

Precision ratio when finding similar pages
We also experimentally examined the precision ratio when finding similar pages to test the effectiveness of our passage similarity search method. In this experiment, we used about 200 Web pages from each of three newspaper Web sites, the two sites used in our first experiment, plus the Mainichi newspaper site [14]. Our results are shown in Table 3. In this experiment, we used the optimal number of keywords obtained in our first experiment. We considered a correct result as finding a similar page, and not finding a similar page when the comparison site did not have a genuinely similar page. The average precision ratio was 56%.

We found that incorrect results were most likely when, for example, a major news story involved several countries and various people and governments. In such a case, the news stories are written from different perspectives that depend on the news sites. Incorrect results also occurred when there were several reports involving different people with a common name (such as Smith). Because a person's name is a proper noun, which would be heavily weighted, the system searches among many different news pages to find similar pages.

6. LINK-BASED CWB

Our CWB searches for similar content by comparing Web pages contents. However, users often want to compare Web pages with a similar structure. For example, a user may want to compare the research database of one research center's Web site with that of another research center. Most research center Web sites include member items, research content items, publication list items, and so on, but these contents depend on each research center. That is, although these Web sites have similar structures and items, they don't necessarily have similar contents. Thus, a user cannot find similar pages by using our CWB. To overcome this problem, and make our CWB more convenient, we are developing a CWB based on link navigation (i.e., a link-based CWB system).

The link-based CWB will search not a similar content Web page but a similar links based on anchor texts. This means link navigation is used to search similar Web pages. The link-based CWB is a light program because it compares only anchor texts. It works only on the client side, and its interface architecture is the same as in our initial CWB system. Figure 8 shows the system architecture of the link-based CWB.

The link-based CWB operates as follows:

1. The user specifies the URLs of the basic Web site and the compared Web site. The CWB shows the top page of each Web site or the user's specified Web pages.

2. CWB divides the shown Web pages into paragraphs by using structure tags.

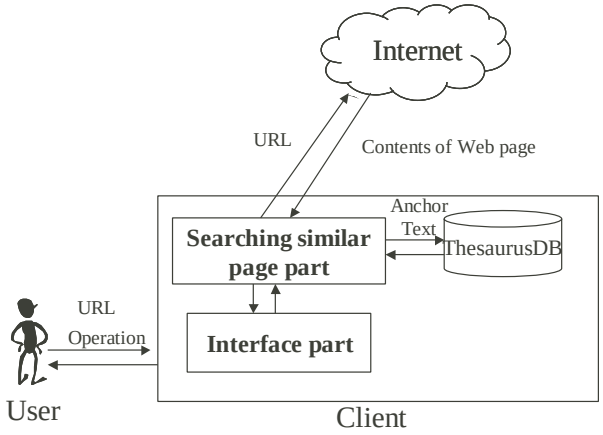


Figure 8: System Architecture of the Link-based CWB

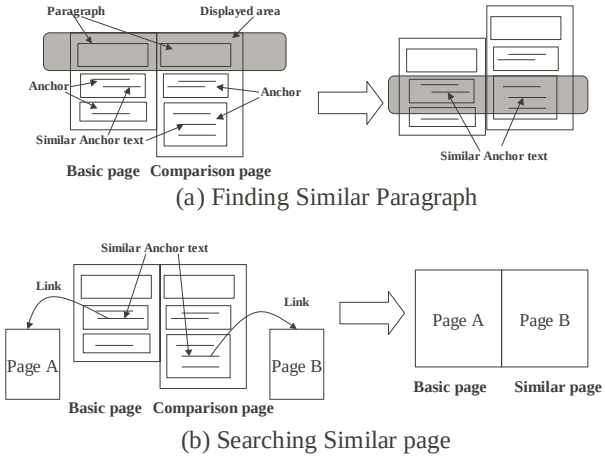


Figure 9: Link-Based CWB Computation Method

3. The CWB finds similar paragraphs in both the basic Web page and the compared Web page.
   When the user scrolls through the basic page, the CWB searches the anchor text in the basic page. If there is an anchor text in the browsed paragraph which is shown in the middle of the basic Web site's window, the CWB then searches for a similar anchor text in the compared Web page and shows a paragraph that includes a similar anchor text in the middle of the compared Web site's window (Figure 9(a)). If there isn't any similar anchor text in the compared page, the system tries to search a similar anchor text in the pages that are linked from or links to the current page. If a paragraph displayed in the basic site window has multiple anchors, and if a comparison page has similar anchors in different paragraphs, the CWB selects the paragraph in the comparison page which has the most anchors and displays this paragraph in the compared Web site window.

4. The CWB searches the similar item.
   When the user clicks an anchor text in the basic Web page, the CWB searches for a similar anchor in the compared Web page. The reason of not using the contents of linked page is that the link-based CWB searches not a similar contents but a similar item. If the CWB can find a similar anchor text in the compared Web page, it automatically shows the top of the linked page from the basic page and the top of the link page in the compared Web page. The compared Web page then becomes a similar Web page (see Figure 9(b)). The same anchor text, however, must always be found in the two Web pages because these two pages are from different Web sites. Therefore, we use a thesaurus database in the link-based CWB.

5. When the user clicks an anchor and navigates to the next basic page, the CWB repeats steps (2) to (4).

Users can thus automatically and concurrently compare Web pages with a similar structure (items) by using the link-based CWB. We plan to implement the link-based CWB for testing in the near future.

7. CONCLUSION

In this paper, we have described our Comparative Web Browser (CWB). The CWB concurrently presents multiple Web pages in a way that allows automatic synchronization of the content of the Web pages. The CWB can display user-specified Web pages from different Web sites at the same time for comparison. A user's control of a Web page is automatically transformed into a series of interactions (which are not necessarily the same actions) with the other Web page to achieve the *content-synchronization* for these pages.

The main characteristics of the CWB are:

- Automatic content-based retrieval of passages from the other Web page based on the passage of the Web page a user is reading.
- Automatic transformation of a user's behavior (scrolling, clicking, moving backward or forward, etc.) on a Web page into a series of behaviors on the other Web page.

Users can thus compare similar Web sites easily, automatically, and concurrently by using the CWB.

8. ACKNOWLEDGMENTS

This research has been partly supported through the Research into Cross-media for Multimedia Data cooperative research project with CRL (Communications Research Laboratory) and Kyoto University.

9. REFERENCES

[1] netcraft homepage
http://www.netcraft.com/survey/

[2] Gomez homepage
http://www.gomez.com

[3] kakaku.com homepage
http://www.kakaku.com

[4] B. Liu, Y. Ma, and P.S. Yu, "Discovering Unexpected Information from Your Competitor's Web Sites", Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2001), San Francisco, CA, August 2001.

[5] B. Liu, K. Zhao, and L. Yi, "Visualizing Web Site Comparisons", The 11th International World Wide Web Conference (WWW2002), Honolulu, Hawaii, May 2002.
http://www2002.org/CDROM/refereed/571/index.html

[6] Seung-Jin Lim and Yiu-Kai Ng, “An Automated Change-detection Algorithm for HTML documents Based on Semantic Hierarchies”, Proc. the 17th Intl. Conf. on Data Engineering (ICDE’01), pp. 303-312, Heidelberg, Germany, April 2001.

[7] Yih-Farn Chen, Fred Douglis, Huale Huang, and Kiem-Phong Vo, “TopBlend: An Efficient Implementation of HtmlDiff in Java”, Proc. the WebNet2000 Conference, San Antonio, Texas, November 2000.

[8] Roy Goldman and Jennifer Widom, “DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases”, Proc. 23rd Intl. Conf. on Very Large Data Bases (VLDB’23), pp. 436-445, August 1997.

[9] Robert D. Doorenbos, Oren Etzioni and Daniel S. Weld, “A Scalable Comparison-Shopping Agent for the World-Wide Web”, Proc. the 1st Intl. Conf. on Autonomous Agents, pp. 39-48, 1997.

[10] Jeffrey Dean, Monika R. Henzinger, “Finding Related Pages in the World Wide Web”, The 8th International World Wide Web Conference (WWW8), Toronto, Canada, May 1999. http://www8.org/w8-papers/4a-search-mining/finding/finding.html

[11] Satoshi Oyama, Katsumi Tanaka, “Web Search Using the Hierarchical Structure of Topics”, Technical Report, IPSJ SIGDBS Technical Report, Kinugawa, Vol. 2002, No. 67 2002-DBS-128, pp. 465-472, July 2002. (in Japanese)

[12] Asahi newspaper site homepage http://www.asahi.com

[13] Yomiuri newspaper site homepage http://www.yomiuri.co.jp

[14] Mainichi newspaper site homepage http://www.mainichi.co.jp