Measuring Structural Similarity Among Web Documents: Preliminary Results ⋆

Isabel F. Cruz[1], Slava Borisov[2], Michael A. Marks[1], and Timothy R. Webb[1]

[1] Department of Computer Science
Worcester Polytechnic Institute
ifc@cs.wpi.edu, mikem@wpi.edu, twebb@cs.wpi.edu
[2] GromCo, Inc
slava@sitebits.com

Abstract. When we describe a Web page informally, we often use phrases like "it looks like a newspaper site", "there are several unordered lists" or "it's just a collection of links". Unfortunately, no Web search or classification tools provide the capability to retrieve information using such informal descriptions that are based on the appearance, i.e., *structure*, of the Web page. In this paper, we take a look at the concept of *structurally similar* Web pages. We note that some structural properties can be identified with semantic properties of the data and provide measures for comparison between HTML documents.

1 Introduction

The ability to search for documents in the vast digital library that constitutes the World Wide Web is a widely used feature. For example, two of the most popular Web search systems report an astounding access frequency: 40 million page views per day for Yahoo! [12] and 400 million page views per day for AltaVista [5].

Existing search systems generally fall into one of the two categories: indices and directories. Search systems belonging to the former group (such as AltaVista, Lycos, etc.) attempt to index the whole World Wide Web space to give a user the capability to look for specific keywords and their combinations. The other group, directories, are based on Web classification systems allowing users to navigate through a manually composed hierarchy of topics. Examples of systems from this category include Yahoo! and Excite.

Other Web search systems like Lira [2] and Letizia [6] are agent-based and "suggest" Web sites to users after analyzing their profiles. Machine learning algorithms are used to update the agent's knowledge about users' preferences. Agents rely on "feature extraction" to provide a basis for computing the search heuristics. Lira uses the vector space information retrieval paradigm, in which documents are represented as vectors with weights of individual keywords w_i as elements. (This representation assumes the existence of a dictionary vector D).

⋆ Research supported in part by the National Science Foundation under CAREER Award IRI–9896052. URLs: www.cs.wpi.edu/~ ifc and casa.wpi.edu

While keywords are the base for the most popular search engines, when describing informally a Web page, we often use phrases like “it looks like a newspaper site”, “there are several unordered lists” or “it’s just a collection of links”. Unfortunately, no widely used Web search or classification tools provide the capability to retrieve information using such informal descriptions that are based on the appearance, i.e., *structure*, of the Web page.

In this paper, we suggest that information, stored in the form of HTML tags, offers significant information about the nature of a Web page and, therefore, may be used as a complementary source of information in automated search or classification systems. We introduce some types of distance functions that measure structural similarity between Web documents. However, most search systems limit the use of structural information to very simple cases (e.g., AltaVista ranks highly keywords that are close to the beginning of the document, for example in the title of a Web page) or ignore all HTML mark-up information, thus reducing the contents of a document to its textual component (e.g., Lira and Letizia).

There are several conceptual hurdles when considering information provided by HTML tags: first, it is rather difficult to translate such descriptions into formal queries. Second, given a wide range of possible descriptions, it is hard to build an indexing system. Finally, concepts like “long list”, “looks like a magazine” are subjective, and in order to use them, a search system would have to be adaptive. Therefore, while there are more or less straightforward ways to classify Web documents according to their topic (Yahoo! is currently the *de facto* standard), it is very difficult to classify documents according to how they look.

In our paper, we make a preliminary step that can help in developing systems capable of handling queries involving structural similarity analysis. Namely, we consider the problem from the following perspective: given a set of documents, what are the ways of measuring how *similar* (or *different*) these documents are? Can we compare documents to groups of documents? What are the possible distance functions?

The paper is organized as follows: In Section 2 we consider three specific subcategories of distance functions, namely those based on tag frequency distribution (Section 2.1), those based on formulas that describe the internal structure of the documents (Section 2.2), and those based on edit distances or evolution between documents (Section 2.3). Finally, we draw conclusions in Section 3.

2 Distance Functions

In order to measure structural similarities between Web pages, we have to choose distance functions defined on the set of all possible documents. The three types of functions considered below are based on Tag Frequency Distribution Analysis (TFDA), Parametric Functions, and Edit Distances, respectively.

2.1 TFDA Based

Perhaps the easiest way to compute distances between two Web documents is the following: first, eliminate everything except HTML tags. Second, for each HTML tag, compute its frequency (in %) in a document, and summarize the frequencies in Table 1 where FREQUENCY k:1 is the frequency of TAG$_k$ in the first document and FREQUENCY k:2 is the frequency of the same tag in the second document (for tags, not found in the document the corresponding frequency value is 0). The TFDA-based "distance" can then be computed as:

$$d = \sum_{1..k} (F_{k_1} - F_{k_2})^2 * w_k$$

where k is the total number of tags, F_{k_1}, F_{k_2} are the frequency values for the tag T_k in the first and the second document, respectively, w_k is the weight for the kth tag, and $\sum_{1..k} w_k = 1$. Since the frequency parameter is measured in percentages, and the sum of these frequencies for both documents always equals 100, we can infer that the maximum distance between any two documents will never exceed 10,000. This information is helpful if normalization is required.

Tag	Frequency (document 1)	Frequency (document 2)
<TAG$_1$ >	FREQUENCY 1:1	FREQUENCY 1:2
...	...	...
<TAG$_p$ >	FREQUENCY p:1	FREQUENCY p:2

Table 1. Frequency of HTML tags in two Web documents.

This method relies on the assumption that tag frequencies reflect some inherent characteristics of a Web document and correlate with its structure. In other words, the assumption is that those frequencies are not random and they are not the same for documents with different structures. To illustrate this point, we show the data that we obtained from a sample of 50 sites (several hundred Web documents), taken at random from the following Yahoo! categories:

- `arts.artists.personal_exhibits` ("Artists")
- `business.companies.law.firms.immigration` ("Lawyers")
- `education.indices` ("Education")

Table 2 contains the average values of tag frequencies along with their standard deviation values, for the three categories. We can notice the following interesting characteristics of this data:

- Tag <A> is normally used to provide a link to another document. It is only natural that sites from the "Education.Indices" category have more links than others.
- Tag <IMG> is used to insert an image. The fact that artists have almost twice as many images per page than lawyers is perfectly understandable.
- Tag <P> separates paragraphs of text. We can see that the frequency of this tag is significantly higher for lawyers' pages.

Tag	Artists Exp.(Dev)	Lawyers Exp.(Dev)	Education Exp.(Dev)
<A>	7.40 % (3.57)	6.95 % (3.22)	9.29% (5.01)
<IMG>	5.40% (2.70)	2.83 % (2.24)	3.07% (2.52)
<P>	6.73% (4.08)	10.62% (4.54)	6.44% (2.86)
<TABLE>	1.14% (0.80)	0.53% (0.90)	0.43% (0.32)

Table 2. Frequency of some HTML tags in documents from three Yahoo! categories.

– Finally, the differences in the frequencies of <TABLE> tag can be explained as follows: art-related pages often use tables not to present data, but to make the layout more attractive. On the other hand, law-related pages may use tables to present data only when it is necessary. As a result, art-related pages have more tables per document than the pages of the other categories.

Note that the standard deviation for the <TABLE> frequency in the lawyers' pages is extremely high (exceeding the average value). This poses a serious problem – we simply cannot rely on statistical information about a tag's average frequency within a certain category, if the standard deviation value exceeds the expectation itself. Unfortunately, most HTML tags appear to have extremely high standard deviation values. <A>, <IMG>, <P>, <BR>, and <FONT> appear to be the most "stable" tags. On the other hand, we can improve this situation by assigning weight depending on how "predictable" the tag frequency is in its category.

Experiment Setup This technique was used in an experiment that we conducted in order to explore possible keywords-topic and structure-topic correlations. In our comparison, we referred to a inter-category similarity matrix obtained by Chekuri *et al.* [3]. This matrix was computed using a similarity function based on the analysis of word-frequency vectors. Chekuri *et al.* considered twenty topics that correspond to twenty high-level Yahoo! categories. The matrix they obtained gives information on the correlation between keywords and the topics chosen.

We have sampled the majority of Web pages stored in Yahoo!. For each of the twenty topics, all of the corresponding URLs were processed. Each URL was processed only once even when referenced by multiple categories. A total of more than 664,000 Web pages were retrieved for an average of thirty thousand valid pages per topic (full details are given in [7]). For each tag, both the expectation and the standard deviation values were computed. The distance function we used is as defined above. Weights corresponding to individual tags were chosen to be proportional to the product of the ratios between average value and deviation (with a shift constant ϵ added to the standard deviation to avoid dividing by zero):

$$w_i = A_i^1/(D_i^1 + \epsilon) * A_i^2/(D_i^2 + \epsilon)$$

and then normalized to ensure that $\sum_{1..k} w_k = 1$.

Results and Comparisons The results of this comparison are summarized in two graphs. Both graphs are based on similarity matrices. The graph of Figure 1 (extracted from [3]) is based on the inter-category similarity matrix obtained by Chekuri *et al.*; the graph of Figure 2 is based on the inter-category similarity

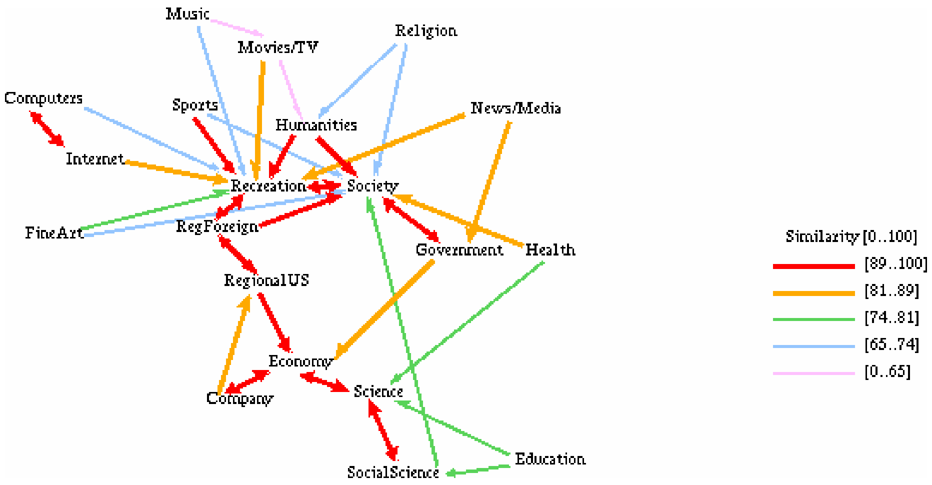


Fig. 1. Nearest neighbors of the categories in the experiment by Chekuri *et al.*

matrix that we obtained and depicts the same connections of Figure 1. In this figure, the shade and thickness of the edges encode the percentile dissimilarity between two adjacent categories.

Despite the difference in methods (the first graph demonstrates connections between *keyword frequencies* and topics, while the second graph is about correlations between values based on *tag frequencies* and topics), there are a number of common dependencies: strong connections between "recreation" and "society" and between "regional US" and "economy". Some connections that were the second strongest in the first graph appear also as the second strongest in the second graph: "news/media – government", "society – health", "movies/TV – recreation". At the same time, some of the strongest connections in the first graph that are not the strongest in the second graph are still among the second strongest: "society – regional foreign", "science – social science", "sports – recreation", "regional foreign – regional US", and "computers – internet".

A couple of strong connections in the first graph now appear much weaker: "economy – science" and "humanities and recreation". The latter, in particular, appears to confirm one's expectation that the two categories may not be closely related. On the other hand, we uncovered interesting (and not that easy to explain!) strong connections, such as "social science – internet" and "economy – sports".

More importantly, we have determined structural analogies which were not as strongly present in the former experiment: "music – movies/TV", "music –

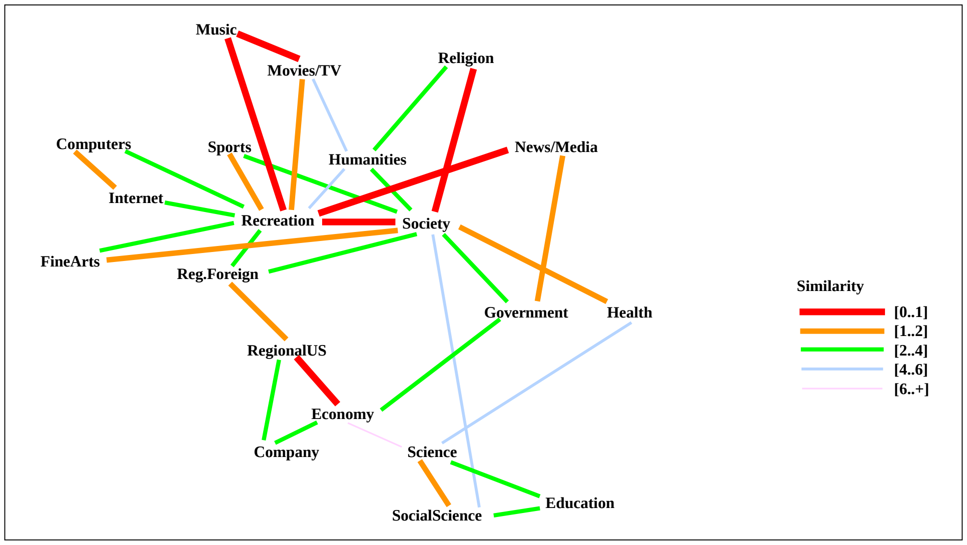


Fig. 2. Connections in the experiment by Chekuri *et al.* as determined by our inter-category similarity matrix.

recreation", "society – entertainment", "recreation – news/media", and (from the complete inter-category similarity matrix) "humanities – social science". These connections seem rather natural given the content of the documents they may contain.

Conclusions The TFDA-based approach has the advantage of being relatively simple. All calculations can be carried out quickly. Also, this method allows for the comparison of Web pages to other pages, Web sites and even entire categories, since it is very easy to compute the corresponding tables for a set of documents. This is helpful in building a classifier: to determine the category a particular Web page belongs to, we simply need to compare its table to the tables of all groups and pick the one with the minimal distance.

On the other hand, one obvious drawback is that the distribution analysis based function will take exactly the same value even if we rearrange all tags contained in the documents being compared! Clearly, changing the order in which tags appear in documents changes the logical structure of the document and, therefore, it should at least affect the distance value.

Finally, we would like to mention one possible extension to this method: the tag frequency table may be augmented with values of other variables such as the ratio between number of tables and cells or combined frequencies of tags. Provided that the corresponding weight parameters are chosen carefully, this simple modification will give us more information on a Web page and, eventually, a more accurate estimate for the distance between documents.

2.2 Formula Based

Another group of functions we consider requires a more sophisticated setup. Suppose we have a set of documents (or groups of documents) we would like to compare to each other. In order to find structural similarities, we should first extract the internal structures. The first step of the formula-based method is to choose a proper data representation for all documents/groups of documents. Then, we need to select a set of functions whose values will eventually be combined in the resulting formula. Finally, when the representation and parametric functions are chosen, we construct a formula (or a set of rules) that maps the values of parametric functions on the appropriate data structure to the "final" value of the distance.

Summarizing, the formula-based method utilizes the following algorithm for computing the distance:

Step 1: Translate documents (or groups of documents) into the chosen data representation.
Step 2: Compute the values of parametric functions.
Step 3: Following a set of rules or a formula, compute the resulting distance value.

Next, we give an example to demonstrate this method. In this example, we measure distances based on structural differences in unordered lists.

Example

Step 1: Document representation. There are different sorts of HTML tags. Some of them (<B>, <I>, <U>, <FONT>, etc.) only affect the way some fragment of the text is presented, without changing the document's format. Other tags actually induce the structure, and they are of the most interest to us. We choose to break the "structurally important" into the following groups:

- <TABLE>, <TR>, <TD>, <TH>
- <DL>, <DD>
- <OL>, <LI>
- <UL>, <LI>
- <CENTER>, <P>, <BR>

In this example, we will look at the group consisting of the following tags: <UL>, <LI> and </UL>.

First, we process the input documents, eliminating all tags, except <UL>, </UL> and <LI>. The result is two (or more, depending on how many documents we are comparing) strings of tags, containing sequences of <UL>, </UL> and <LI>. These sequences are not random. In fact, they can be specified by the following grammar (provided that the HTML code of the input documents is syntactically correct).

```
R → (E)*
E → < UL > M < /UL >
M → < LI > M
M → E
M → empty_string
```

Our representation will be a string, consisting of '(', ')' and non-negative numbers. With every <UL> encountered, the conversion algorithm will increment the counter value and catenate a '(' character to the string and the element counter will be set to zero. With every closing tag (< /UL>), it will output the counter value and a ')'. At the end, the counter value will be output again. This algorithm produces strings such as 0(5(3)1)0.

This means that the original ("code") string contained 5 <LI> tags at the first "<UL> level", then 3 <LI> tags at the second (nested) "<UL> level", then another tag <LI> at the first level again. It can be easily inferred that the maximum depth of the entire structure was 2 (at any character position in this string, the depth can be computed as the difference between the number of '('s and the number of ')'s placed to the left). We can also compute the average length of unordered lists at each level (which is 3), and obtain the maximum and the minimum values (5 and 1).

Representations like this are simple enough to allow for a variety of parametric functions (some examples will follow) to be defined on them, yet they seem rich enough to contain interesting information about the original structures.

Step 2. Parametric function values. Given a string of this form, how much can we say about the internal structures of a document? Parametric functions are supposed to measure some characteristics of these structures and give values that are used in the resulting formula. Here are some examples:

- Len(S): Length of the string
- Lef(S): Number of '('s
- Sum(S): Sum of all numbers
- Ave(S): Average of all numbers
- PMSeq(S): Number of sequences of the form '(*number*)'

Step 3. Distance value. Once we have computed the values of all the chosen parametric functions for both documents (documents had been converted into the data representation of our choice), we can look at the differences and produce an answer. If we happen to know something about the representation (or we have conducted some sort of statistical experiment), then we may know, for example, that the differences in Sum's are less important than the differences in Lef's. One option is to build a formula with weights assigned to all parametric functions so that we can simply sum all the (squared) differences:

$$dist = w_{len} * (Len(S_1) - Len(S_2))^2 + w_{plu} * (Lef(S_1) - Lef(S_2))^2 + \\ w_{ave} * (Ave(S_1) - Ave(S_2))^2 + w_{sum} * (Sum(S_1) - Sum(S_2))^2$$

Other possibilities include algorithms that compute the distance based on some other criteria.

This example demonstrated that "building" formula-based distance functions can be almost as easy as using the TFDA method. On the other hand, the performance of such functions will depend on how "good" was the choice of all three components (data representation, parametric functions and the resulting formula or algorithm).

2.3 Evolutionary (Edit-distance Based)

The third approach is based on edit distances. The *edit distance* between two documents is defined as the minimum number of operations from a fixed set (usually consisting of deletion, insertion and substitution) required to transform one document into another. It has been shown that this distance is a useful measure of evolutionary distance [8].

Wang *et al.* suggest the use of edit distances for HTML documents represented as parse trees [9]. Their system, *TreeDiff*, is a counterpart of the UNIX command *diff*.

The evolutionary method is elegant and has applications in different areas. However, we have detected that in some cases the edit distance between two HTML documents and their appearance seem uncorrelated, as shown by the following examples.

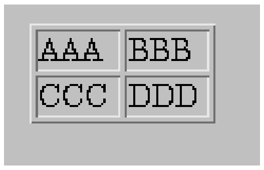
Example 1 This example illustrates that the transformation between two HTML documents may require several operations, even in the case where the layout of the documents may not differ substantially. In Figure 3 we show the HTML source code for two tables (we removed all tag parameters such as BORDER=1 and COLSPAN=2), while the tables appear almost identical.

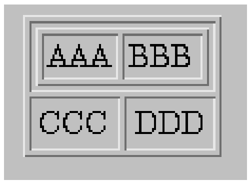
Example 2 In this example we show that two HTML documents whose source code is very similar can differ substantially in terms of their layout. For example, a simple deletion or insertion of a tag may considerably change the way the document is presented. The two simplified HTML code fragments of Figure 4 differing by a single tag (<TR>) demonstrates this point.

3 Conclusions and Future Work

While most Web search and classifier tools rely on keyword-based distances between documents, we consider Web documents strictly in terms of their structural (i.e., HTML tag) component and look at three different methods for defining similarity distances between documents. The simplest of the methods (*TFDA-based*), which is strictly based on the frequency of HTML tags (and independent of their occurrence pattern within the document), gave encouraging results when compared with a method that is based only on keywords [3]. Our approach not only corroborates the strong similarities found between categories of documents

```
<HTML>                <HTML>
<TABLE>               <TABLE>
                      <TR>
                      <TD>
                      <TABLE>
<TR>                  <TR>
<TD>AAA</TD>          <TD>AAA</TD>
<TD>BBB</TD>          <TD>BBB</TD>
</TR>                 </TR>
                      </TABLE>
                      </TD>
                      </TR>
                      <TR>
                      <TD>
                      <TABLE>
<TR>                  <TR>
<TD>CCC</TD>          <TD>CCC</TD>
                      </TR>
                      </TABLE>
                      </TD>
                      <TD>
                      <TABLE>
                      <TR>
<TD>DDD</TD>          <TD>DDD</TD>
</TR>                 </TR>
</TABLE>              </TABLE>
</HTML>               </HTML>
    (a)                   (b)
```





(a) (b)

Fig. 3. HTML code for two tables and resulting tables.

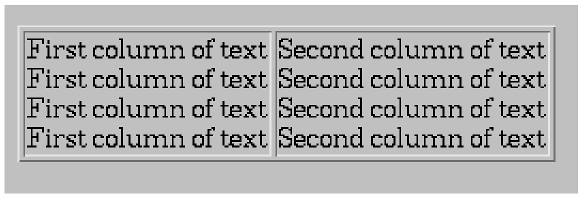
in [3], but also discovers other relevant relationships. Another method, that we call *formula-based* takes into account the placement and sequence of HTML tags in documents. Although slightly more complicated than the previous method, it could yield even more interesting results, particularly because the methods that are used can be tailored by the user. Finally, we consider *evolutionary* methods, similar to those used in discovering patterns in molecular genetics. Such methods are elegant and powerful, provided that the edit distances on which they are based are correlated with the appearance of the HTML documents.

```
<HTML>

<TABLE>
<TR>
<TD>
First column of text<BR>
First column of text<BR>
First column of text<BR>
First column of text<BR>
</TD>

<TD>
Second column of text<BR>
Second column of text<BR>
Second column of text<BR>
Second column of text<BR>
</TD>
</TABLE>

</HTML>
```

(a)

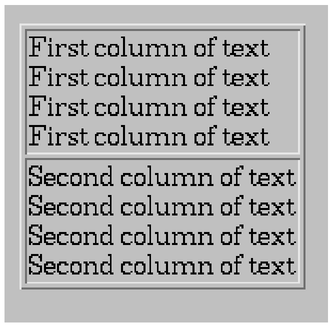
```
<HTML>

<TABLE>
<TR>
<TD>
First column of text<BR>
First column of text<BR>
First column of text<BR>
First column of text<BR>
</TD>
<TR>
<TD>
Second column of text<BR>
Second column of text<BR>
Second column of text<BR>
Second column of text<BR>
</TD>
</TABLE>

</HTML>
```

(b)



First column of text First column of text First column of text First column of text	Second column of text Second column of text Second column of text Second column of text

(a)



First column of text First column of text First column of text First column of text
Second column of text Second column of text Second column of text Second column of text

(b)

Fig. 4. HTML code for two tables and resulting tables.

In spite of the promising results that we obtain, we do not advocate that the distance between documents should be based only on the structure of the document (no more than we believe that it should be based only on keywords). Instead, similarity between documents (and their subsequent use in classification and search) should be based on a variety of similarity measures. One of the challenges resides in providing good user interfaces (e.g., for browsing), with which the user can incrementally specify different kinds of similarity parameters, as opposed to specifying everything at once. Existing user interfaces that could be extended in this way include the *facet-space interface* [1] and AMIT [11]. The interfaces in [4, 10] use constraints to specify the layout of new documents. A similar approach could be used to retrieve documents with a given structure.

Another issue for future research involves the indexing of the Web space according to document layout characteristics. Possibilities include associating these characteristics with document categories (e.g., Yahoo! categories), or having them included in the metadata associated with entire sites or individual documents, which, in turn, could be queried with more "traditional" search engines like AltaVista.

Acknowledgements We would like to thank the referees for useful comments.

References

1. R. B. Allen. Retrieval from Facet Spaces. *Electronic Publishing*, 8(3):247–257, 1996.
2. M. Balabanovic, Y. Shoham, and Y. Yun. An Adaptive Agent for Automated Web Browsing. Technical Report CS-TN-97-52, Stanford University, February 1997.
3. C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. Web Search Using Automatic Classification. Technical report, Stanford University and IBM Almaden Center, December 1996. theory.stanford.edu/people/wass/publications/Web_Search/.
4. I. F. Cruz and W. T. Lucas. Delaunay^{MM}: a Visual Framework for Multimedia Presentation. In *IEEE Symposium on Visual Languages (VL '97)*, pages 212–219, 1997.
5. Digital Equipment Corporation. Digital's AltaVista Search Unveils Largest and Freshest Web Index. www.altavista.digital.com/av/content/pr101497.htm.
6. H. Lieberman. Letizia: an Agent that Assists Web Browsing. In *Proc. of the International Joint Conference on Artificial Intelligence*, 1995.
7. M. A. Marks and T. R. Webb. Internet Documents Clustered by Structure. Major Qualifying Project, Worcester Polytechnic Institute, 1997.
8. D. Sankoff and J. B. Kruskal, eds. *Time Warps, String Edits, and Macromolecules: the Theory and Practice of Sequence Comparison.* Addison-Wesley, 1983.
9. J. T.-L. Wang, G. J. S. Chang, G. Patel, L. Rhihan, D. Shasha, and K. Zhang. Structural Mapping and Discovery in Document Databases. In *ACM-SIGMOD Intl. Conf. on Management of Data*, pages 560–563, 1997.
10. L. Weitzman and K. Wittenburg. Automatic Presentation of Multimedia Documents Using Relational Grammars. In *ACM Multimedia Conference*, 1994.
11. K. Wittenburg and E. Sigman. Visual Focusing and Transition Techniques in a Treeviewer for Web Information Access. In *IEEE Symposium on Visual Languages (VL '97)*, pages 20–27, 1997.
12. Yahoo! Inc. Yahoo! Ranked No. 1 Web Site Among Business Users in First-Ever PC Meter Workplace Study. www.yahoo.com/docs/pr/release106.html.