A Treemap-Based Result Interface for Search Engine Users

Shixian Chu, Jinfeng Chen, Zonghuan Wu, Chee-Hung Henry Chu, and Vijay Raghavan

The Center for Advanced Computer Studies, University of Louisiana at Lafayette
P.O. Box 44330, Lafayette, LA 70504-4330, U.S.A.
{sxc1825,jxc5466,zwu,cice,raghavan}@cacs.louisiana.edu

Abstract. Search engines and metasearch engines on the World Wide Web typically display representative information such as the result document titles and snippets in a one-dimensional ranked list. Alternatively, a few others such as Clusty and Kartoo cluster their results so that users can, to a certain extent, interact with documents, keywords and/or clusters. The number of documents that can be effectively presented in one screen is usually limited to between 10 and 20. In this paper, we propose a method based on treemap visualization that substantially improve the information compactness. In addition, it provides a few unique post-search interaction methods that enable a user to manage a large number of results. An online prototype system is built. Various experiments are designed and done to evaluate the efficiency, effectiveness and usability.

Keywords: Information visualization, web search interface.

1 Introduction

Search engines and metasearch engines such as Google (www.google.com), Yahoo! (www.yahoo.com),Windows Live (www.live.com), and many others essentially have the same one-dimensional rank list-based search result interface. Taking Google as an example, in a result page for the query "treeemap", the information of 10 out of more than 1.5 million results, including the title, url and snippet, were displayed on the first page (Fig. 1). Only the first 6 or 7 results can be seen in a 1280 by 800 sized window of the Web browser, if the user does not click the mouse to scroll down. Obviously, the user has to click the "next page" button to get more results, if the first 10 results cannot satisfy him or her.

A number of other search engines, such as Clusty (www.clusty.com) (Fig. 2), Kartoo (www.kartoo.com), and Ask (www.ask.com), cluster the results into different categories before presenting them to the users. The clustering helps users if they need to narrow their search to a more specific domain. Fundamentally, however, there are still only a small number of results that can be seen within one screen.

The current search engine result display methods have a few common limitations, such as:

- The presentation often makes it hard for users to find desired information, especially if it is not within the top-ranked search results.

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part I, HCII 2007, LNCS 4557, pp. 401–410, 2007.

- Users may not be able to collect enough relevant materials from the first few screens of search results, while it is a tedious, and sometimes impractical, job to go through pages of scrolled lists.Many returned results scattered in the list may be not of interest to the user while he/she still has to scan through them.

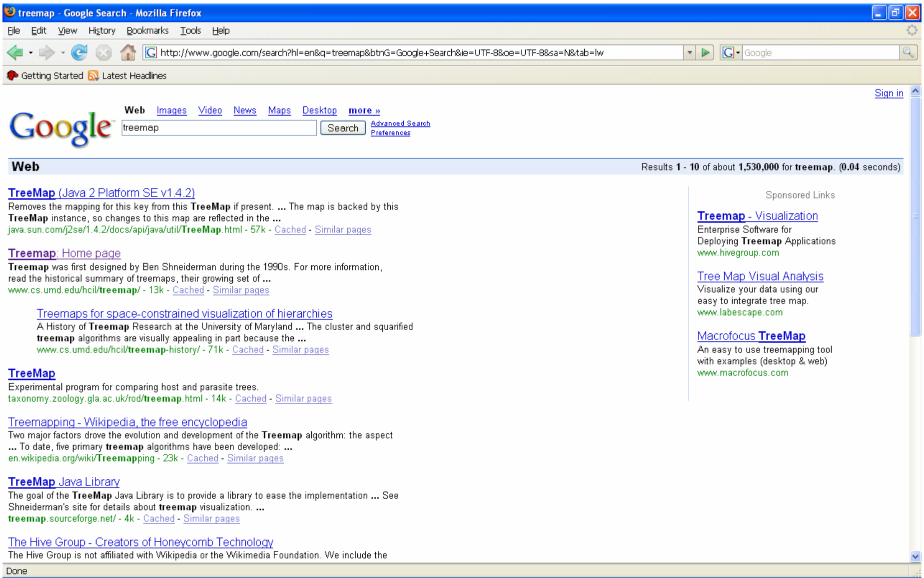

Fig. 1. A screen shot of Google's search result page

These limitations make it difficult for a user to manage large amounts of information returned from search engines, though such information is already retrieved and available for use.

In this paper, we present a treemap-based search result visualization solution for search engine users. Treemap is a space-constrained visualization of hierarchical structures. In our work, such a structure is used to represent search results and the relation between phrases and documents. Thus, hundreds of results can be displayed in one screen. Furthermore, a few useful interaction methods that cannot be realized in traditional search engine interfaces can be applied to further empower users' capability to manage a large number of search results.

In addition to providing a general solution for applying treemap in search engine result visualization, our other contributions in this paper include that, we propose two treemap algorithms to enhance user experiences; we also provide our study on performance and usability; finally, we have implemented a Web-based prototype system for interested users to have first-hand experience.

In section 2, we briefly go over treemap, the concept and related prior arts. In section 3, we explain our method in detail. In section 4, we compare our result visualization methods with a few other major search engines, including Google, Clusty, Kartoo. Finally, we conclude in section 5.

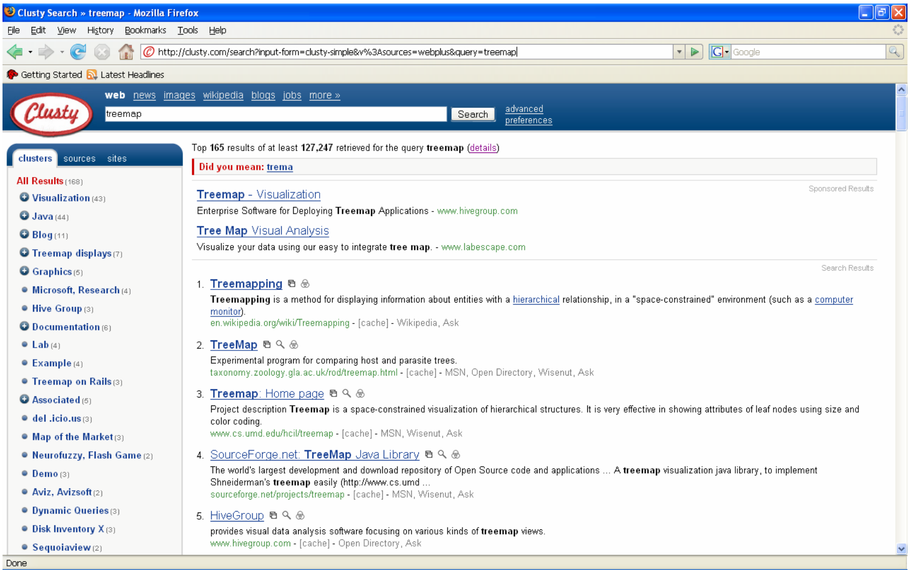

Fig. 2. A screen shot of Clusty's search result page

2 Treemap and Related Work

Treemap is a space-constrained visualization of hierarchical structures. It turns a tree into a planar space-filling map. Typically, in a two-dimensional space, each node is represented by a rectangle; child nodes are arranged to be nested in their parent node's rectangle. The arrangement, size, and color of each rectangle represent different attributes of the node. A treemap can enable users to compare nodes and sub-trees even at varying depth in the tree, and help them spot patterns and exceptions. A lot of studies have been done since Ben Shneiderman invented the treemap in early 1990s. For example, several treemap algorithms, from using rectangles to other polygons and circles, from two dimensional to three dimensional, have been designed; treemap visualization has been integrated into research works ranging from data mining to bio-informatics; powerful open source software has been developed. It has been successfully applied to many industries, from inventory control, project management, to financial analysis and e-commerce. A comprehensive list of publications and applications of treemap is maintained at www.cs.umd.edu/hcil/treemap.

Essentially, treemap visualization helps display an enormous amount of information and reveals underlying patterns in the data. In the context of the Web, treemap has been applied to visualize:

- online news: Newsmap at www.marumushi.com/apps/newsmap/Newsmap,
- websites: del.icio.us being the most popular treemap; and
- the Usenet: Netscan at netscan.research.microsoft.com/treemap/ Netscan.

In this paper, we apply it to search engines for visualization of their retrieved results.

3 Treemap Based Search Engine Result Visualization

Beyond a presentation in a ranked list, mechanisms that take advantage of the human capacity to process visual information can enable a user to manage a large amount of results. From the earlier Cat-A-Cone [1] and Card Visualization [2] methods, to the more recent Clusty and Kartoo interfaces, many attempts have been made and considerable improvement is achieved. Cat-A-Cone used a Cone Tree [3], which is a three-dimensional visualization technique that makes effective use of the available screen space while displaying the entire data structure at the same time. However, as the 3-D cone tree is projected onto a 2-D plane in the browser, overlap inevitably occurs which obscures the view of the rear items [4].

Presentation of search results to a user needs to consider information at three levels [5]. At the set level, the user should be able to see trends and hot spots; at the web site level, the user should see the structure or parts of the web; at the document level, the user should know whether to focus on a specific page. Many developed systems, such as [6-8], are based on the focus+context approach. The context of the information is typically the relevance of the retrieved web page to the query [9].

Clusty groups search results in different categories; each category is identified by a key phrase as the label. By clicking on a label, the corresponding documents are displayed. In some sense, Clusty method can be seen as an extension of Card visualization. Kartoo utilizes a graphic interface and a proprietary algorithm. A few result documents and key phrases are shown in a "contour map". When the mouse hovers over an object (i.e. a document or a key phrase), curve lines are drawn to link related objects. However, it appears that this method can only manage a few tens of records and is not scalable.

3.1 A General Tree-Map Based Solution

The basic idea of our method is: Firstly, search result documents will be clustered into hierarchical groups (based on content, title and/or snippet). The hierarchical cluster structure is essentially a tree with each node being a cluster or a sub-cluster. Each node has its "name", which is a key phrase extracted from the documents within the cluster. Secondly, we map the tree into clustered treemap. Each cluster in the treemap represents a cluster of documents. Each document is represented by a block in the treemap cluster, with a name (the title of the document, for example). The size of a block is used to indicate result relevancy. The bigger the block, the more relevant (to the query) the corresponding documents. Features of blocks, including colors and brightness, are used to indicate the file type, domain, date, location, size and other document properties. Finally, such a treemap will be shown to users. Users can also conveniently customize the block feature-document property mapping through additional functions provided at the interface.

Comparing to existing result displaying/visualizing techniques, the advantages for our method are as follows.

1. It allows users to comfortably take advantage of natural intuition, spatial cues and perception at the interface. Logical cluster structure of the search results can be expressed clearly; Relationships among phrases and docs are also clear; Comprehensive information and trend can be presented—for example, with just a

glance, the user can get the sense of how many results are there in each domain or how many document are located in a given country.

2. Also, our method can meet the two stringent requirements of the online web search:
   a. Information compact — hundreds of results can be shown in one result page;
   b. Network and computation efficient — In our implementation, the size of a result page with two hundred search results is around 75k to 100k bytes. In average, one Google result web page of 10 search results is around 32k bytes. A Clusty result page of 20 clustered search results is about 150k bytes. Also, the treemap algorithm does not require intensive computation.
3. It is interesting.

4 Experiments and Evaluations

There are many aspects of a visualization system that can be tested for its performance. In [10], the factors considered for the success of visual structures are: (i) target user group, (ii) type and size of result sets, and (iii) the nature of task. By varying these factors, the performance of an interface can be judged by: (i) the task completeness or effectiveness, (ii) the task performance time or efficiency, and (iii) the user subjective acceptance or satisfaction. A scenario-based evaluation of the interface used in [11] varies the user group to measure the time to complete a task, the perceived precision, and users' subjective reactions. We have designed several tests to evaluate the performance of search result visualization methods. More specifically, the following aspects were assessed:

- Time to find the most relevant result
- Capability of providing the macro-view of search results
- Maximum Topological distance
- The Index of ease of use
- The Interestingness Index

Each of the following sub-section addresses one of the aspects listed above. In most tests, we compare our treemap-based method with those used by a few other major search engines (Table 1). Note that our search engine gets results information and uses cluster information from SRC MSN; it subsequently uses the centralized treemap to visualize the organized information. In our experiments, we assumed that the quality and coverage of these five search engines are virtually equal to each other, for the ease of comparison.

Table 1. Search engines and their URLs used in our experiments

Search Engine	URL
Google	http://www.google.com
SRC	http://rwsm.directtaps.net/
Kartoo	http://www.kartoo.com
clusty	http://www.clusty.com
Treemap-based	http://lincweb.cacs.louisiana.edu:8080/treemap

4.1 Time to Find the Most Relevant Result

In this test, we compare how much time is needed for a user to find his/her desired results. There are two types of queries that a user can send, viz. implicit and explicit.

Implicit Queries refer to queries that a user's intentions cannot be easily identified from the query itself or a user does not know what to anticipate from search results. To guide testers to send proper implicit queries, we adopted the description at [12] and divide a query into the following 5 sub-categories:

1) A query that has Ambiguity. For example, "Java" is an ambiguous query as it may mean a programming language or coffee.

2) A query that has many sub-topics. Many query terms contains sub-topics, such as a query one of our testers sent: "Argentina", its sub-topics include its history, travel, business, and so on.

3) A query for finding unknown fact of peoples. As an example, one of our tester sent "bill Clinton".

4) A query for finding unknown relationship of peoples. When a query is two person's name, find out their relationships. For example, "Bush Blair". (Note: None of our testers chose such kind of queries)

5) A query that is a question, to find out the possible answers. For example, one of the testers sent queries such as "smallest animal" and "fastest animal".

Explicit Queries refer to queries that have key terms that can hopefully pinpoint the specific desired result documents. For example, a test sent a query "Distinctive image features from scale-invariant key points" to find the paper that has the query string as its title.

Testers were required to send different implicit queries and explicit queries to each search engine, and record the time it takes to find desired information from results. If it takes the user more than 3 minutes to find satisfying information, please mark time to be infinite

Table 2. The number of clicks it takes to find satisfactory results from a search engine

	# of Clicks	Google (# of queries)	Clusty (# of queries)	SRC (# of queries)	Kartoo (# of queries)	Treemap (# of queries)
Implicit Queries (32 queries in total)	1	*13*	*1*	*0*	*11*	*26*
	2	*5*	*21*	*19*	*2*	*2*
	3	*2*	*7*	*7*	*10*	*2*
	4+	*8*	*2*	*4*	*4*	*1*
	NSRF*	*4*	*1*	*2*	*5*	*1*
Explicit Queries (12 queries in total)	1	*12*	*12*	*12*	*10*	*12*
	2	*0*	*0*	*0*	*1*	*0*
	3	*0*	*0*	*0*	*0*	*0*
	4+	*0*	*0*	*0*	*1*	*0*
	NSRF*	*0*	*0*	*0*	*0*	*0*

* NSRF = No Satisfactory Result Found

Table 3. The time (in seconds) it takes to find satisfactory results from a search engine

	Time	Google	Clusty	SRC	Kartoo	Treemap
Implicit Queries (32 queries in total)	<5s	*10*	*9*	*12*	*6*	*16*
	5s~10s	*6*	*7*	*7*	*5*	*8*
	10s~60s	*7*	*12*	*9*	*11*	*4*
	60s~180s	*5*	*3*	*2*	*4*	*2*
	NSRF*	*4*	*1*	*2*	*5*	*1*
Explicit Queries (12 queries in total)	<2s	*7*	*2*	*0*	*0*	*0*
	2s~5s	*5*	*8*	*11*	*2*	*9*
	5s~10s	*0*	*2*	*1*	*4*	*2*
	10s~180s	*0*	*0*	*0*	*6*	*1*
	NSRF*	*0*	*0*	*0*	*0*	*0*

* NSRF = No Satisfactory Result Found

Due to complexity of the test, and limitations on time and resource, we could only perform the test at a relative small scale. Six (6) volunteers have participated in this test. Each one was given time to practice using the five search engines until they are familiar with the interfaces, each of them was asked to send 5~6 implicit queries and 2 explicit queries to each search engine. The time and the number of mouse clicks between that a query is sent and that a satisfactory result is found, are recorded.

From tables 2 and 3, we could clearly see the advantage of treemap method when the search query is implicit. In 26 out of 32 queries, only 1 click is needed to get the satisfactory results, while the best in the other four competitors only can do this in 13 queries (Google). Similarly, in 26 out of 32 queries, it took less than 10 seconds to find results while the record of the best in 4 competitors is 19 queries (SRC).

For explicit queries, Google has advantages in terms of number of clicks and time needed over others. However, the performance of treemap method is still comparable with the other three search engines.

4.2 Capability of Providing the Macro-view of Search Results

As mentioned in previous sections, most current search engines can only effectively display a limited number of search results, while our treemap-based method can display hundreds of results on the computer screen. As more information is available, it provides a better basis for a user to get the macro view of his/her search. In Table 4, we summarize the capability of the search engines in terms of the macro view of search results.

Google lists results in a ranked list. The other four search engines cluster results in different categories. Such categorization often helps a user to understand the search better. From the different sizes of clusters, the different color code of documents based on the document publication date, document type, domain of the website, or other features, Treemap has a unique property, viz. color coding, which can clarify the distribution based on some feature of results. Among all search engines, only through the treemap-based method can a user can get the answers of questions such as in which domains the searched topic was most popular (Figure 3), or in which years the searched topic was most active, with just at a glance of the result page.

4.3 The Topological Distance Between Related Result Documents

Clustering help similar documents be close to each other on the screen. When information in one result document is not sufficient, the user can easily locates the related document in the neighborhood. Among the five methods, treemap displays related documents in a cluster in the most compact manner. We borrow the concept of *topological distance* to illustrate this feature.

Take the first cluster in Fig 3, which is labeled by the keyword "industrial", as an example; it has 10 documents d_0, d_{12}, d_{24}, …, d_{96}. In the treemap visualization, any two documents are separated by essentially more than 2 documents. For example, between d_0 and d_{96} , there are d_{12}, d_{80} or d_{60} and d_{71}. So, we can say the maximum topological distance between two related documents is 2. However, in case of SRC or Clusty, the clustered results are lined in a ranked list. Thus, the maximum topological distance between two related documents is 8, as between d_0 and d_{96} , there are the other 8 documents in the clusters listed. Since there is no clustering, Google is worse in this aspect. Usually, Google display 10 results a page, that virtually prevent the d_0 on page 1 and d_{96} on page 9 from appearing together. Even though Google could be customized to show 100 documents in a page, between d_0 and d_{96} , there will be some other 95 documents listed. We note that Katoo was not included in the comparison because its proprietary atypical visualization method makes it difficult to apply the topological distance to its interface.

4.4 Ease of Use and Interestingness

After they conducted experiments described in 4.1, the 6 volunteers were asked to rank the five search engine result visualization methods based on their degree of ease of use and that of interestingness. In both cases, we use a 7-point scale: 1 means very easy to use/very interesting and 7 means very difficult to use/very boring.

Table 4. Macro-view of search results

	Google	Clusty	SRC	Kartoo	Treemap
#Results shown on a screen	<10	<10	<10	20 ~30	>100
Are results categorized based on content?	No	Yes	Yes	Yes	Yes
Can a distribution of common features be shown?	No	No	No	No	Yes

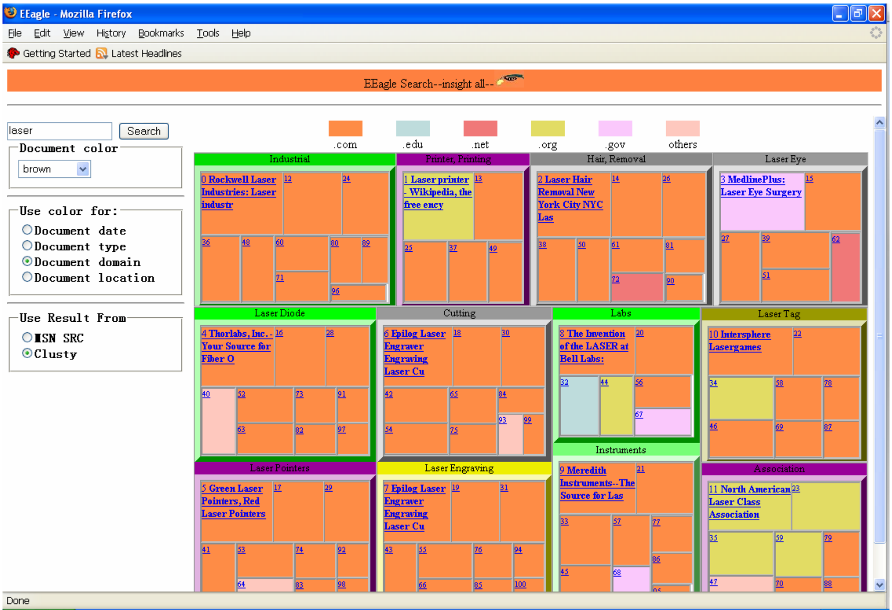

Fig. 3. A screen shot of treemap search result page for the query “laser”

Table 5. Average subjective scores of ease of use and interestingness of search engines

Search Engine	Ease of Use (score: rank)	Interestingness (score : rank)
Treemap	1.83 : 2	2.67 : 1
Google	1.5 : 1	4.5 : 5
SRC	3.16 : 3	4: 4
Clusty	3.33 : 4	3.83 : 3
Kartoo	4.66 : 5	2.83: 2

The average scores are listed in Table 5, which shows that the treemap is in the second place, considered to be quite easy to use, only slightly less so than Google but better than others. As for interestingness, treemap is considered to be the most interesting method among all five search engines.

5 Conclusions

In this paper, we proposed 1) a treemap-based result visualization method for Web search engines, and 2) a set of methods to evaluate the performance of the interface. Our tests show that, in most cases, the treemap method outperforms that of other search engines in information compactness, macro-view capability, efficiency of finding results for implicit query, ease of use and interestingness.

Acknowledgments. This work was supported in part by the Louisiana Governor's Information Technology Initiative.

References

1. Hearst, M.A., Karadi, C.: Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In: Proc. 20th Annual ACM SIGIR Conference, pp. 246–255 (1997)
2. Mukherjea, S., Hara, Y.: Visualizing World Wide Web search engine results. In: Proc. 1999 IEEE Int. Conf. Information Visualization, pp. 400–405 (1999)
3. Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone trees: animated 3d visualizations of hierarchical information. In: CHI '91: Proc. SIGCHI Conference on Human factors in Computing Systems, pp. 189–194 (1991)
4. Engdahl, B.: Ordered and Unordered Treemap Algorithms and Their Applications on Handheld Devices, Master's Degree Project, Stockholm, Sweden (2005)
5. Mann, T.M.: Visualization of WWW-search results. In: Proc. 10th Int. Workshop on Database and Expert Systems Applications, pp. 264–268 (1999)
6. Roberts, J.C., Suvanaphen, E.: Visual bracketing for Web search result visualization. In: Proc. 7th Int. Conf. Information Visualization, pp. 264–269 (2003)
7. Granitzer, M., Kienreich, W., Sabol, V., Dosinger, G.: WebRat: Supporting agile knowledge retrieval through dynamic, incremental clustering and automatic labeling of Web search result sets. In: Proc. 12th IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 290–301 (2003)
8. Cellary, W., Wiza, W., Walczak, K.: Visualizing Web Search Results in 3D. IEEE Computer 37(5), 87–89 (2004)
9. Nguyen, T., Zhang, J.: A novel visualization model for Web search results. IEEE Trans. Visualization and Computer Graphics 12(5), 981–988 (2006)
10. Mann, T.M., Reiterer, H.: Evaluation of different visualizations of Web search results. Proc. 11th Int. Workshop on Database and Expert Systems Applications, pp. 586–590 (2000)
11. Hoeber, O., Yang, X.D.: A comparative user study of Web search interfaces: HotMap, Concept Highlighter, and Google. In: Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intelligence, pp. 866–874 (2006)
12. accessed on 02/15/2007, http://rwsm.directtaps.net/usage.htm