5-2000

VAS (Visual Analysis System): An information visualization engine to interpret World Wide Web structure

Tarkan Karadayi

VAS (VISUAL ANALYSIS SYSTEM): AN INFORMATION VISUALIZATION ENGINE TO INTERPRET WORLD WIDE WEB STRUCTURE

A Thesis

by

TARKAN KARADAYI

Submitted to the Graduate School of the
University of Texas – Pan American
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2000

Major Subject: Computer Science

VAS (VISUAL ANALYSIS SYSTEM): AN INFORMATION VISUALIZATION ENGINE TO INTERPRET WORLD WIDE WEB STRUCTURE

A Thesis
by
TARKAN KARADAYI

Approved as to style and content by:

Richard H. Fowler, Ph. D.
Chair of Committee

Xiannong Meng, Ph. D.
Committee Member

Zhixiang Chen, Ph. D.
Committee Member

Jacob Jen-Gwo Chen, Ph. D., P. E.
College Dean

May 2000

Karadayi, Tarkan, VAS (Visual Analysis System): An Information Visualization Engine to Interpret World Wide Web Structure, Master of Science, Computer Science, May 2000, 93pp., 3 tables, 12 illustrations, references, 86 titles.

People increasingly encounter problems of interpreting and filtering mass quantities of information. The enormous growth of information systems on the World Wide Web has demonstrated that we need systems to filter, interpret, organize and present information in ways that allow users to use these large quantities of information. People need to be able to extract knowledge from this sometimes meaningful but sometimes useless mass of data in order to make informed decisions. Web users need to have some kind of information about the sort of page they might visit, such as, is it a rarely referenced or often-referenced page? This master's thesis presents a method to address these problems using *data mining* and *information visualization* techniques.

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. R. Fowler for sharing his knowledge, resources, input and valuable discussions on all aspects of this project. I thank Dr. X. Meng and Dr. Z. Chen for their assistance and comments. I also thank my wife Anna, my sons Taran and Kyle, and my parents Meral and Unal Karadayi for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

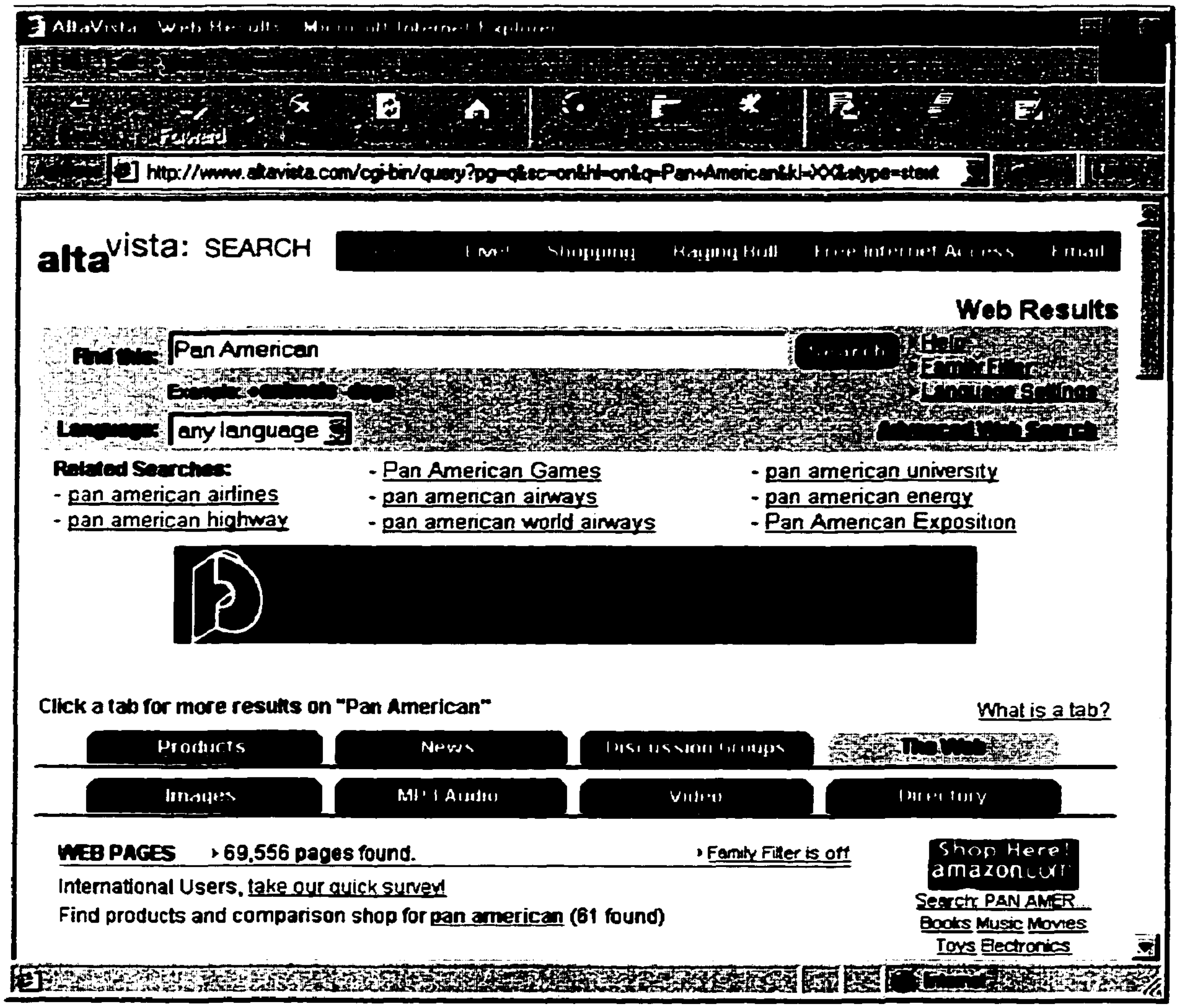
Page

1. INTRODUCTION

Every day the World Wide Web grows by approximately a million pages, adding to the hundreds of millions of pages already on the Web. This tremendous amount of information is linked together by more than a billion annotated connections, also known as hyperlinks.

However, the current network of information lacks organization and structure due to the Web's rapid and uncontrolled growth. In fact, the Web has changed into a global tangle of immense proportion. Individuals with any education, interest or culture can create Web pages in any language or style. Pages can range in size from a few kilobytes to megabytes and contain text, video, animation or sound. How, then, can one extract related pages in response to a specific need for certain information?

One of the best solutions to this problem is to use search engines. People rely on such search engines that look for specific words or terms. Because of today's technology, large search engines exploit the ability to store and index most of the Web. Such engines can therefore create large-scale indices that allow users to retrieve the set of all Web pages containing a given keyword. Experienced users can effectively use search engines for tasks that can be resolved by searching for closely constrained keywords and phrases. However, these search engines are not applicable for a wide range of important tasks. In particular, a topic of any breadth will usually hold several thousand or million related Web pages, most of them irrelevant (Figure 1). And, usually, a user will be willing to look at only a few of these pages. How, then, can one quickly find only the information he/she needs and trust that it is dependable and authentic? This situation follows from the Web's growing size and lack of structure, as well as the insufficiency of Web search technologies. Even with advanced search engines, the task of filtering and ranking query result sets can be enormous.



http://www.altavista.com/cgi-bin/query?pg=q&sc=on&hl=on&q=Pan+American&kl=XX&stype=stext

Find this: Pan American

Language: any language

Related Searches:
- pan american airlines
- pan american highway
- Pan American Games
- pan american airways
- pan american world airways
- pan american university
- pan american energy
- Pan American Exposition

Click a tab for more results on "Pan American"

WEB PAGES ›69,556 pages found. ›Family Filter is off

International Users, take our quick survey!

Find products and comparison shop for pan american (61 found)

Figure 1
AltaVista search engine results for query "Pan American" returns 69,556 pages.

Users often follow a group of pages via hyperlinks through such search engines result sets. After traversing a number of linked documents on the Web, users feel lost or become disoriented, because they often do not remember how they got there and do not know where they are in this big information space. This is one of the major problems of the World Wide Web. One of the solutions to this problem is to provide users with both local and global views of the information space. Representing these views visually using information visualization technique, helps users quickly determine where they are in this big information space and promote a quick perception and understanding of the hierarchical structure of the World Wide Web.

Researchers attempt to utilize the advances in graphics and visualization technologies in order to improve methods in which users interact both visually and non-visually with information distributed in World Wide Web.

> This process of interaction with the information may start with browsing, continue with digesting and assimilating pieces of information, terminate with generation of new information, and iterate through analyze of pre-existing and new information [4, pg. 123].

Additionally, a large extent users are not directly involved in the development of the World Wide Web and its capabilities. In order to create useful information systems we need to involve users in designing information resources and create user focused interfaces that take user needs into consideration. The ideal is that users would forget that there is a computer separating them from the information.

Using recent developments and improvements in visualization, computer graphics, and storage, we can build improved systems for information navigation, access, and retrieval in which visualization and user interface could play a better role.

1.1. SUMMARY

VAS's approach to providing improvements to filter and interpret World Wide Web information centers on 1) identifying useful data on the Web using data mining techniques, and 2) taking user's needs into consideration by allowing them to interact with the information using dynamic information visualization techniques.

The Visual Analysis System (VAS) provides a user-centered architecture. The system receives a user's query as a keyword and then submits the query to other search engines, such as Alta Vista, retrieving the best matches (usually 200) to serve as a starting point for VAS directed search. Visual Analysis System then starts its own search by following links in these pages to retrieve a second level set. Links in these pages can also be followed to retrieve a third set of pages. Typically, pages retrieved in the second level set contain a large number of links to pages within the previously retrieved sets, as would be expected starting from a single query. As page sets are retrieved, a graph showing connections among pages is formed and displayed. This dynamically formed and displayed graph is the primary means of interaction with the system.

Other researchers [59] have characterized pages as *authorities* (many links to the page), which provide the best source of information on a given topic, and *hubs* (many links from the page), which provide useful lists of possibly relevant pages. Visual

Analysis System first distills a large World Wide Web search topic to a size that makes sense to the human user to provide for identifying the topic's most definitive or authoritative Web pages. That is, not only is a set of relevant pages located, but also those relevant pages of the highest quality are identified. Visual Analysis System examines the fact that the underlying structure of Web contains not only Web pages, but also hyperlinks that connect one page to another.

But just computing and finding authorities and hubs in a given keyword is not enough. Taking advantage of the human's natural pattern-recognition ability by creating visualizations of results, Visual Analysis System displays these authorities and hubs to the user as a dynamic visual summary for interaction. This enables users to get the information they need and reach decisions in a relatively short time. Finally, displaying results with a 2-dimensional view, users can interact with the resulting visualization, navigate through visual space, and select objects easily. This allows users to focus their attention directly on results of the query in an immediate and compelling way.

1.2. LINK TOPOLOGY

There is increasing research effort directed at the combination of textual content and link structure of the Web to improve organizing, visualizing and searching of the WWW. This thesis originates from the problem of searching the WWW. Building on this, it attempts to define a meaningful notion of link structure as a way of analyzing issues in information visualization, data mining, and information discovery.

As noted in [60, pg. 2], "The *link structure* of the World Wide Web represents a considerable amount of latent human annotation, and thus offers a promising starting point for structural studies of the Web."

By examining these interconnections through linked structure of Web, Visual Analysis System defines two types of pages: *authorities* and *hubs* [59]. The former are the best sources of information on a particular topic and the latter are collections of links to those locations. Providing users with authorities and hubs for a topic should enable users to find much of the information they need more efficiently and quickly.

The importance in this thesis is on an investigation of the *link topology* of the WWW, and some fairly pervasive themes that have identified about the structure of hyper-textual communities developed in the Web. The notion of *community* [60] provides a useful perspective from which to view the seemingly haphazard development of the Web's infrastructure.

> The findings of analyzed link structure of WWW are valuable in number of ways. Analysis of the link structure of the WWW suggests that the on-going process of page creation and linkage, while very difficult to understand at a local level, results in structure that is considerably more orderly than is typically assumed. Thus, it gives us a global understanding of the ways in which independent users build connections to one another in hypermedia that arises in a distributed fashion, and it provides a basis for predicting the way in which on-line communities in less computer-oriented disciplines will develop as they become increasingly wired.

> It also suggests some of the types of structured, higher-level information that designers of information discovery tools may be able to provide both for, and about, user populations on the Web [60, pg. 2].

The link structure of World Wide Web can be a valuable source of information regarding the contents of the Web pages. This thesis focus on the use of links for examining the collection of pages relevant to a broad search topic, and identifying the most authoritative pages for such topics using data mining and visualization techniques.

1.3. SEARCHING ON THE WWW

This thesis originates from the problems of identifying pages that are related to a given query, which can be defined as the process of discovering pages using search engines on the WWW. Searching starts with a user supplied query. There is more than one type of query, and the handling of each may require different techniques. Consider for example, the following types of queries:

- Specific queries. E.g., " Does UNIX supports OpenGL?"
- Broad-topic queries. E.g., " Find information about Graphics programming."
- Similar-page queries. E.g., "Find pages 'similar' to cs.panam.edu."

Concentrating on just the first two types of queries for now, we can see that they present very different sorts of obstacles.

The difficulty in handling specific queries is centered, roughly, around what could be called the *Scarcity Problem*: there are very few pages that contain the required information, and it is often difficult to determine the identity of these pages.

> On the other hand, searching using broad-topic queries returns many thousand relevant pages on the WWW as shown in Figure 1. Therefore there is not an issue of scarcity here. Instead, the fundamental difficulty lies in what could be called *Abundance Problem*: the number of pages that could reasonably be returned as relevant is far too large for a human user to digest (Figure 1) [59, pg. 1].

To be able to build effective search methods under these conditions, we need to find a way to filter this big collection of related Web pages to a small set of the most "authoritative" or "definitive" ones.

One of the primary challenges of this project is to accurately identifying authorities in the context of a particular query topic. Therefore the main focus of this thesis is the addressing issues of this notion of authority, relative to a broad-topic of query. There are a couple of complications that arise from notions of authority. First, consider the home page of University of Texas – Pan American, "www.panam.edu". As one can expect that this is one of the most authoritative pages for the query "Pan American". Unfortunately, there are thousands of pages on the WWW that use the term "Pan American" (see Figure 1), and "www.panam.edu" is not the only one that uses the term most often, or most importantly, in any other way that would help using a text-based ranking function. Certainly, there is no special feature that will define the page's authority. Second, consider the problem of searching the home pages of the main WWW search engines. One could begin from the query "search engines", but there is an immediate difficulty in the fact that many of the natural authorities (Yahoo!, Excite, AltaVista) do not use the term on their pages. This is a fundamental and recurring phenomenon. As another example, there is no reason to expect the home pages of Mazda or Chevrolet to contain the term "automobile manufacturers" [59, pg. 2].

> Analyzing the hyperlink structure among WWW pages gives a way to address many of the difficulties discussed above. Hyperlinks encode a considerable amount of latent human judgement, and this kind of judgement is precisely what is needed to formulate a notion of authority. Specially, the creation of a link on the WWW represents a concrete indication of the following type of judgement: the creator of page *p*, by including a link to page *q*, has in some measure conferred authority on *q*. Moreover, links afford us the opportunity to find potential authorities purely through the pages that point to them; this offers a way to circumvent the problem, discussed above, that many prominent pages are not sufficiently self-descriptive [59, pg. 2].

This thesis offers a link-based model for identifying the notion of authority and shows a method that identifies related authoritative WWW pages for broad search topics. The model is based on the relationship that exists between authorities for a topic and those pages that link to many related authorities, which are hubs. There are certain natural types of relationships between hubs and authorities in the graph defined by the link structure [59, pg. 3], VAS utilizes and improves this method to develop an algorithm that identifies both types of pages simultaneously. The algorithm operates on focused subgraphs of the WWW constructed from the output of an index based WWW search engine such as AltaVista. This technique of constructing such subgraphs is designed to produce small collections of pages likely to contain the most authoritative pages for a given topic [59, pg.3].

Also, the visual analysis system developed in this thesis provides an enhanced user-interface for visualization of search results, including graph layout and display of structural information using information visualization techniques. While this project has been implemented on top of existing search services, using static data, its features could easily be combined with any search engine for real time performance and evaluation.

2. RELATED WORK

Browsing through hyperspace without becoming lost, overcoming the rigidity of the World Wide Web, and aggregating and classifying relevant information are important research questions. Visual analysis tools and techniques combine overlapping areas of research: 1) WWW indexing (and searching) and data mining techniques, and 2) information visualization techniques.

2.1. VISUAL ANALYSIS TOOLS AND TECHNIQUES

Visualization has been used routinely in data mining as a presentation tool to generate initial views, navigate data with complicated structures and convey the result of an analysis. Perhaps a stronger visual data mining strategy lies in tightly coupling the visualizations and analytical processes into one data mining tool. Letting human visualization participate in an analytical process and decision-making remains a major challenge. Certain mathematical steps within an analytical procedure may be substituted with human decisions based on visualization to allow the same analytical procedure to analyze a broader scope of information. Visualization supports humans in dealing with decisions that no longer can be automated. [18]

The WebQuery [2] system offers a powerful new method for searching the Web, based on connectivity and content. WebQuery does this by examining links among the nodes returned in a keyword-based query, then rank the nodes, giving the highest rank to the most highly connected nodes. By doing so it finds "hot spots" on the Web that contain information germane to a user's query. There are two overlapping areas of research related to this work: WWW indexing (and searching) and visualization of the WWW. WebQuery explores several techniques for visualizing the information, including 2D graphs, 3D graphs, lists, and "bullseye". For larger data sets it uses 2D cone tree. A cone tree is constructed from the neighbor graph by traversing the structure, depth first, beginning at the root node and adding nodes to the tree in the order that they are visited. With this it provides a visualizing of a graph as a tree. It also uses other visualization techniques; such as, springs and weights. This algorithm and bullseyes algorithm draws the user's attention to the highly ranked nodes.

VANISH [6] is a visualization tool that supports the easy integration of new semantic domains for visualization. It was used to implement an algorithm that supplements the standard content - only searches with the structural information collected by spider. After the query has been post processed via three-step process, an information space is constructed from the neighbor set, suitable for visualization. It has the capability of to visualize information spaces structured as both graphs and trees.

Navigational View Builder [8] is a tool that allows hypermedia designers to create efficient view of the underlying information space. It focuses on how to make complicated hyperspace structure more comprehensible by letting the user view the hierarchy globally and in detail.

T. Bray [9] has generated dynamic VRML representation of the Open Text index database. His work displays sites based on their visibility (number of links into the site), their luminosity (number of links out of the site), and their size.

R. M. Rohrer, J. L. Sibert, D. S. Ebert et al [12], describes a shape based visual interface for information retrieval and interactive exploration that exploits shape recognition. The exploratory system uses procedurally generated shapes coupled with an underlying text-retrieval engine. A visual interface based on 3D shapes enhances traditional text-based queries and summarization. The interface lets users visualize multidimensional relationships among documents and perceive more information than with conventional text-based interfaces.

Another research project binds visual attributes to information attributes to allow important information about retrieved records to be represented visually [17]. It uses bird's eye view of tabular visualization of the search results for the keywords. In this visualization, cubes represent the documents and their color represents the domains. The width, height, and depth of the cubes are mapped to the frequency of the keywords. If the user is actually interested in information and software visualization, he/she should look for documents with width and either depth or height. The user can navigate through the 3D space to look for the interesting documents. Zooming in to parts of the space is also allowed.

Narcissus [21] uses techniques of self-organizing systems and virtual reality to generate visualizations through which the user can navigate and manipulate objects in the information space. The system is implemented as a process, which communicates with applications such as, web browsers using KQML: A Knowledge query and manipulation language [20]. This provides a degree of application independence and also allows the system to work concurrently with several, possibly heterogeneous, applications and also allows collaborative working between several users. The Narcissus system provides the user with a window onto their information space and allows the user to navigate through this space and to select and manipulate objects. The user is able to control some aspects of behavior of the objects and can select individual objects and classes of objects, which should be visible.

The Harmony [23] Internet browser provides a number of tightly coupled two and three dimensional visualization and navigational facilities that help provide location feedback and alleviate user disorientation under the Hyper-G Internet information system. Central to the design of Harmony is the concept of location feedback. When a document or collection is visited, its location within the collection structure is automatically displayed in the Session Manager's collection browser by following a hyperlink, or via the local map. This simple technique is a powerful instrument in the fight against becoming "lost in hyperspace"; users can orient themselves with reference to a fixed structural framework. [23] [32] [33]

Web-based visualization lets user customize applications and data representations not originally targeted for each other by dynamically linking Web-based data and visualization applications. For very large data sets, visualization may be the only possible approach. R. M. Rohrer and E. Swing [35] view visualization as an integral component in data analysis. Their development efforts help information analysts find important relationships anomalies, and trends. Their simple interpolative clustering scheme resembles the VIBE system's scheme [36]. VIBE displays several key terms at vertices on a 2D polygon and locates the relevant documents by interpolating between vertices. By using the third dimension, it can provide additional key terms for greater cluster resolution and pack more information into the document cluster space.

Advanced Multimedia Oriented Retrieval Engine (AMORE) [37] is a search engine that integrates image and text search. It has several unique features not present in the current Web search engines. It allows the automatic indexing of both text and image from one or more Web sites. This integration of text and image search is useful in various situations, most importantly searching for Web documents. The combination of text and image search can reduce the number of irrelevant documents that are arrived or increase the number of retrieved relevant documents. Visualization of the search results allows the user to understand the results better. The focus + content view of a Web node helps in understanding the position of the node in the Web site from which it was retrieved.

M. Marchiori [38] presents a novel method for extracting "hyper" informative content from a Web object in contrast with current commercial search engines, which only deal with the "textual" informative content. It also permits focusing separately on the textual and hyper components. It permits improvement of current search engines in a smooth way, since it works on top of any existing scoring function. The method can considerably improve effectiveness and utility for the user of such meta-engines.

Dozens of existing search tools and the keyword-based search model have become the main issues in accessing the WWW. Various ranking algorithms, which are used to evaluate the relevance of documents to the query, have turned out to be impractical. This is because the information given by the user is usually inadequate to give a good estimate. C. Chang and C. Hsu [39] propose a new idea of searching under the multi-engine search architecture to overcome the problems. This include clustering of the search results and extracting of occurrence keywords with the user's feedback, which better refine the query in the search process. It also provides construction of the concept space to gradually customize the search tool to fit the usage for the user at the same time.

The Control [40](Continuous Output and navigation technology with Refinement Online) project at Berkeley explores ways to improve human-computer interaction during data analysis. The project's goal is to develop interactive, intuitive techniques for analyzing massive data sets. It focuses on systems that iteratively refine answers to queries and give users online control of processing, thereby tightening the data analysis process loop. Users can use these techniques in diverse software contexts including decision support database systems, data visualization, data mining, and user interface toolkits. Users can quickly sense if a particular query or mining algorithm reveals anything interesting about the data. They can refine or halt the processing, if necessary, and issue other queries to investigate further.

Clouds [40] is an online visualization technique that provides interactivity on large data sets. It renders records as they are fetched, and it also uses those records to generate an overlay of shaded regions of color (clouds) that estimate missing data.

Dynamic Queries [41] allow users to see an overview of the database, rapidly explore and conveniently filter out unwanted information. Users fly through information spaces by incrementally adjusting a query while continuously viewing the changing results. These issues remain in database and display algorithms, and user interface design, with rapidly displaying and changing many points, colors, and areas, multidimensional pointing, with visual display techniques that increase user comprehension, and integration with existing database systems using Dynamic Queries for Visual Information seeking.

E. Frecon and G. Smith [43] present a virtual reality based application to be used alongside traditional Web browsers. The application aims at providing a structure to the network of documents which have been visited during a browsing session. They use the different dimensions of a 3D space to position representations of documents using different metrics, in which the relationships between the documents are also depicted. WEBPATH [43] is a tool that unobtrusively visualizes a user's trail as they browse the Web, and can be tailored to best suit the users working needs. WEBPATH provides users with a graphical view of their recent activities, enabling them to see how a given document was reached and perform complex searches for previous documents visited.

The Cam Tree [48] visualizes hierarchies as trees of labeled rectangular shapes representing nodes and leaves, interconnected by lines. Each subtree is laid out as a cone with its root at the top of the cone and the children along the cone base. Interactions with Cam Tree include rotation of the tree when a node or leaf has been selected with the mouse. This brings the path to the selected rectangle closest to the user and highlights the rectangles on that path. The Information Cube [48] uses semi-transparent nested cubes to represent leaves or internal nodes. The parent-child relationships are represented by nesting child cubes inside their parent cubes, scaling cubes to enclose the contained cubes. Textual labels are displayed on cube surfaces. Color and transparency level indicates the currently selected cube. In the Information Landscape [48] nodes are represented as pedestal shapes standing on a flat surface with lines connecting pedestals to form a tree. Leaves are represented as box shapes standing on the pedestals. The height of the boxes encodes an attribute such as the size of the data.

A new visualization technique called RDT [49] (Reconfigurable Disc Tree) can alleviate the disadvantages of cone trees significantly for large hierarchies while maintaining its context using 3D depth. In RDT, each node is associated with a disc around which its children are placed. Using discs instead of cones as the basic shape in RDT has several advantages: significant reduction of occluded region, sharp increase in number of displayed nodes, and easy projection onto plane without visual overlapping. It can increase the number of nodes displayed on the screen. VISIT [49] (Visual Information System for reconfigurable Disc Tree) provides 2D and 3D layouts of RDT and various user interface features such as tree reconfiguration, tree transformation, tree shading, viewing transformation, animation, selection and browsing which can enhance the user perception and navigation capabilities.

Overview diagrams are one of the best tools for orientation and navigation in hypermedia systems. Navigational View Builder [8] is a tool that allows the user to interactively, creates useful visualizations of the information space. It uses four strategies to form effective views; these are binding, clustering, filtering, and hierarchization. These strategies use a combination of structural and content analysis of the underlying space for forming the visualizations.

DaVIME [50] (Data Visualization, Indexing, and Mining Engine) is a software architecture that performs data visualization, indexing, and mining in an integrated environment. It presents users a unified view of information service. When a user issues an information service request, DaVIME calls upon appropriate software components (agents) to provide the service requested. Depending on the type of service requested, DaVIME might call one or more agents into action to satisfy the user request. It includes Document Explorer [51], which analyzes text information in large document collections and present to the user suitable visualization tools. DUSIE [50] (Dynamic User created Searchable Index Engine) extends the hierarchical indexing schemes currently used in popular browsers to let users build content-based searchable indexes. ParaCrawler is a parallel Web search engine that uses novice ranking and indexing algorithm to provide users with more accurate search results in shorter time.

Document Explorer [51] is a system for visualization of WWW content structure. Visualization, browsing, and query formulation mechanisms are based on document's semantic content. These mechanism complement text and link based search by supplying a visual search and query formulation environment using semantic associations among documents. The user can view and interact with visual representations of WWW document relations to traverse this derived document space.

Document retrieval is a highly interactive process dealing with large amounts of information. Visual representations can provide both a means for managing the complexity of large information structures and an interface style well suited to interactive manipulation. Information Navigator [52] utilizes visually displayed graphic structures and a direct manipulation interface style to supply an integrated environment for retrieval. A common visually displayed network structure is used for query, document content, and term relations. A query can be modified through direct manipulation of its visual form by incorporating terms from any other information structure the system displays. Visualization of these large data structures makes use of fisheye views and overview diagrams to help overcome some of the inherent difficulties of orientation and navigation in large information structures.

Fetuccino [61] provides an enhanced user-interface for visualization of search results, including advanced graph layout, it displays of structural information and support for standards. The research paper proposes two enhancements to existing search services over the Web. One enhancement is the addition of limited dynamic search around results provided by regular Web search services. The second enhancement is an experimental two-phase paradigm that allows the user to distinguish between domain query and a focused query within the dynamically identified domain.

J. E. McEneaney [67] proposes methods that lead to a more direct representation and analysis of user movement in hypertext and to empirically explore the relationship of resulting measures to performance in a hypertext search task.

The Web as a graph: measurements, models, and methods [71] research paper describes two algorithms that operate on the Web graph, addressing problems from Web search and automatic community discovery.

SPHINX [75] is a Java toolkit and interactive development environment for Web crawlers. Unlike other crawler development systems, SPHINX is geared towards developing crawlers that are Web-site specific, personally customized, and relocatable. It allows site-specific crawling rules to be encapsulated and reused in content analyzers, known as classifiers.

2.2. WEB SEARCH AND DATA MINING TOOLS AND TECHNIQUES

A popular technique for finding information on the WWW is to use one of the content-based search tools such as AltaVista. These search tools attempt to index the entire web via its content, where they define content to be the words in a page. However, they can never completely achieve this goal because the Web is a constantly changing database. The search tools provide users with the ability to search the index, returning pointers, URLs, to pages that match the query words to a greater or a lesser degree. The problem in querying is one of term choice, commonly called the vocabulary problem. It arises because there are many ways of expressing similar ideas using natural language.[5]

MOMSpider [7] is a spider constructed specifically to combat the problem of consistency and maintenance in evolving Web structures. Clever [1] is a search engine that analyzes hyperlinks to uncover two types of pages: authorities, which provide the best source of information on a given topic, and hubs, which provide collections of links to authorities. The research group developed an algorithm to compute the hubs and algorithms called HITS (Hyperlink Induced Topic Search) algorithm. Beginning with a search topic, specified by one or more query terms, the HITS algorithm applies two main steps; a sampling component, which constructs a focused collection of several thousand Web pages likely to be rich in relevant authorities; and a weight propagation component, which determines numerical estimates of hub and authority weights by an interactive procedure. Pages with the highest weight returns as hubs and authorities for the search topic.

In addition to finding hubs and authorities, hyperlinks can be used to categorize Web pages. Hyperlinks contain high quality semantic clues to a page's topic. But exploiting this link information is challenging, because it is highly noisy. HyperClass [10] embodies one approach to this problem, making use of robust statistical models such as Markov using random fields together with a relaxation labeling technique.

The methodology of influence weights from citation analysis relates to a link based search method developed by S. Brin and L. Page [11]. They used this method as the basis for their Google Web Search engine. Google first computes a score, called the PageRank, for every page indexed. Given a query, Google returns pages containing the query terms, ranked in order of these pages' Page Ranks. It focuses on Authorities pages. It is something of hybrid between the keyword and human-annotation approaches. It uses its own crawler called Googlebot, which searches for hyperlinks. It looks for Web pages that hyperlink to other pages that are deemed relevant to the topic, based on text-matching and other techniques. Google ranks such pages highly and to return them in response to a search query.

The Smart [16] information retrieval system was developed by Salton at Cornell University. Many of today's information retrieval systems build on ideas from Smart. It supports real-word queries, simple enough to understand, well tested for comparing results. It uses an underlying vector space model in which all documents, queries and results are defined as weighted vectors of word occurrence. This allows for easy comparisons such as in document similarity scoring.

Internet Fish Construction Kit [42] is a tool for building persistent, personal, dynamic information gatherers for the WWW. It incorporates deep structural knowledge of the organization and services of the Web. It differs from current resource discovery tools in that they are introspective, and are also capable of on-the-fly reconfiguration, modification, and expansion. Dynamic reconfigurations and expansions permits IFISH to be modified to take advantage of new information sources or analysis techniques, or to model changes in the user's interests, as they wander the Web.

There have been several approaches to ranking pages in the context of hypertext and the WWW. In work predating the emergence of the WWW, Botafogo, Rivlin, and Schneiderman [55] worked with focused, stand alone hypertext environments. They defined the notions of index nodes and reference nodes. An index node is one whose out-degree is significantly larger than the average out-degree, and a reference node is one whose in-degree is significantly larger than the average in-degree. They also proposed measures of centrality based on node-to-node distances in the graph defined by the link structure.

Carriere and Kazman [2] proposed a ranking measure on WWW pages, for the goal of re-ordering search results. The rank of a page in their model is equal to the sum of its in-degree and its out-degree; thus, it makes use of a 'directionless' version of the WWW link structure.

Brin and Page [11] have recently proposed a ranking measure based on a node-to-node weight propagation scheme and its analysis via *eigenvectors*. Specifically, they begin from a model of a user randomly following hyperlinks: at each page, the user either selects an outgoing link uniformly at random, or jumps to a new page selected uniformly at random from the entire WWW.

Frisse [56] considered the problem of document retrieval in singly authored, stand-alone works of hypertext. He proposed basic heuristics by which hyperlinks can enhance notions or relevance and hence the performance of retrieval heuristics. Specifically, in his framework, the relevance of a page in hypertext to a particular query is based in part on the relevance of the pages it links to.

Small and Griffith [57] use breadth-first search to compute the connected components of the undirected graph in which two nodes are joined by an edge if and only if they have a positive co-citation value. Pitkow and Pirolli [58] apply this algorithm to study the link-based relationships among a collection of WWW pages.

Kleinberg [59] developed a set of algorithmic tools for extracting information from the link structures of such environments, and reported on experiments that demonstrate their effectiveness in a variety of contexts on the World Wide Web. The central issue he addresses within his framework is the distillation of broad search topics, through the discovery of authoritative information sources on such topic. He proposes and tests an algorithmic formulation of the notion of authority, based on the relationship between a set of relevant pages and the set of hub pages that join them together in the link structure.

D. Gibson, J. Kleinberg, and P. Raghavan [60] developed a notion of hyperlinked communities on the WWW through an analysis of the link topology. By invoking a simple, mathematically clean method for defining and exposing the structure of these communities, they were able to derive a number of themes, communities can be viewed as containing a core of central authoritative pages linked together by hub pages. They exhibit a natural type of hierarchical topic generalization that can be inferred directly from the pattern linkage.

MetaCrawler [64] is a fielded Web service that represents the next level up in the information 'food chain'. It provides a single, central interface for Web document searching. Upon receiving a query, MetaCrawler posts the query to multiple search services in parallel, collates the returned references, and loads those references to verify their existence and to ensure that they contain relevant information. It also serves as a tool for comparison of diverse search services.

Harvest [66] is a system that provides an integrated set of customizable tools for gathering information from diverse repositories: building topic-specific content indexes, flexibly searching the indexes, widely replicating them, and caching objects as they are retrieved across the Internet. The system interoperates with WWW clients and with HTTP, FTP, Gopher and NetNews information resources.

Focused Crawler [68] is a hypertext resource discovery system. The goal of a focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics. The topics are specified not using keywords, but using exemplary documents.

Rather than collecting and indexing all accessible Web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the Web.

D.Gibson, J. Kleinberg, and P. Raghavan [69] describe a novel approach for clustering collections of sets and its application to the analysis and mining of categorical data. Their approach is based on an iterative method for assigning and propagating weights on the categorical values in a database table; this facilitates a type of similarity measure arising from the co-occurrence of values in the dataset.

O. Zamir and O. Etzioni [70] articulate the unique requirements of Web document clustering and reports on the first evaluation of clustering methods in this domain. W. W. Cohen and W. Fan [73] describe a method for learning general, page-independent heuristics for extracting data from HTML documents.

Efficient crawling through URL ordering [76] studies in what order a crawler should visit the URLs it has seen in order to obtain more important pages first. Obtaining important pages rapidly can be very useful when a crawler cannot visit the entire Web in a reasonable amount of time. The paper defines several importance metrics, ordering schemes, and performance evaluation measures for this problem.

HyPursuit [80] is a hierarchical network search engine that exploits content-link hypertext clustering. It clusters hypertext documents to structure a given information space for browsing and search activities. This content-link-clustering algorithm is based on the semantic information embedded in hyperlink structures and document contents.

2.3. VISUALIZATION TOOLS AND TECHNIQUES

Robertson [19] identifies four processes that need to be supported by appropriate visualizations: Sense making (building an overall understanding of the information), Design, Decision making (building a decision and a rationale for that decision), and Response tasks (finding information to respond to a query). MITRE Corporation [4] developed an enhancement of NCSA Mosaic that allows the user to view the hyperspace depicted as a visual "tree" structure. In addition to viewing, users can "jump" from one document to another by pointing and clicking the mouse without having to go back resource by resource or page by page. While browsing, users often would like to view the names of documents and how they are linked to each other without actually opening and reading each document.

Procedural modeling techniques use algorithms and code segments to dynamically generate, abstract, and encode model detail. These approaches provide a flexible, dynamic framework in which a small set of parameters can control the generation of detailed models. Examples of procedurally generated shapes include fractal surfaces, implicit surface models (blobbies), and grammar-based models (L-systems). Procedural shapes are attractive, since they abstract model detail and yet allow fine algorithmic control. [13]

Ebert et al [14][15], outlined several examples of procedural shapes used as glyphs for multidimensional data visualization. These glyphs use procedurally generated shapes based on fractals, superquadrics, and implicit surfaces for text-visualization.

Vion-Dury et al [22] report work using a function of an object's name to generate a distinct polyhedral shape for the objects. This is potentially useful, particularly when the name carries some semantic information, which might lead to related objects having common physical features.

The most well known work in the area of 3D-information visualization is that of Card et al [19][24] at Xerox PARC on the Information Visualiser and 3D/Rooms. They allow users to interactively explore workspaces modeled as three-dimensional rooms. Particular nodes (data sets) are visualized in rooms of their own, and doors lead from one room to another in a manner similar to hyperlinks. The Information Visualiser provides 3D presentations for linear and hierarchically structured information: the perspective wall [25] and cone tree [26] respectively. The wall has a large front section, and left and right sides, which tail off into the background. Information can be slid along the wall to bring it into focus on the front section. The cone tree can be rotated to bring interesting parts to the front and pruned to remove non-relevant information. VizNet [27] also uses a cone tree representation for hierarchical information, but provides an additional spherical representation for associative relationships. Lower level objects are displayed on lower level spheres.

SemNet [28] was an exploratory system, which represent knowledge bases as directed graphs in 3D. Labeled rectangles (nodes) were connected by lines or arcs. The 3D layout has the advantage over 2D layouts that the nodes of an arbitrary graph can be positioned so that no arcs intersect.

Serra et al [29] discuss the use of 3D object hierarchies with attached multimedia documents. Each component in the 3D-object hierarchy may be combined into a concept node with text, image, and video documents. Links may be made from these text, image and video documents to other concept nodes.

Smith and Wilson [30] describe a prototype system based on Hypercard and Virtus Walkthrough (a 3D visualization system), in the context of an academic departmental information system. They enabled users to interactively explore a 3D model of the department: when they approached within a certain distance of a source anchor, it automatically triggered to display a corresponding text document.

The File System Navigator (FSN, or Fusion) written by J. Tesler and S. Stransnick at Silicon Graphics [31] visualizes a Unix file system as an information landscape. Directories are represented by blocks laid out on a plane, their height representing the cumulative size of the contained files. User can fly over the landscape, taking it in as a whole, or swoop down to a specific directory.

N. Gershon [34] developed a system which permits automatic generation of maps to any specified number of generations. This facilitates, hyperspace structure browsing without forcing users to read the documents. It also enables one command printing of all linked documents. With this system user can interactively and visually modify the structure of hyperspace, creating a personal space.

WWW3D [44] takes a radical approach in presenting the structure of Web documents, it integrates the display of Web documents and structural information about the portion of the Web that the user has explored into a single 3D structure. Documents are represented within the space as spheres. The content of the Web page is depicted on the inside of the sphere as 3D icons. When a user selects a link from within a page, the new page is loaded, and represented by a new sphere. The two spheres are then connected with an arrow to visually represent the hyper-link between them.

WebBook [45] is a three-dimensional web browser that allows multiple pages to be viewed simultaneously and organizes the pages into books. Books can order pages according to different filters such as relative URL books, search reports, and book books.

WebMap [46] is a browser extension that shows a 2D graphical relationship between pages. Each page is represented by a small circle that can be selected to display the actual page. Links between pages are colored to indicate whether the destination document has already been read or whether it is located on a different server.

PadPrints [47] is a prototype zooming Web browser within a multi-scale graphical environment. Instead of having a single page visible at a time, multiple pages and links between them are depicted on a large zoomable information surface. Pages are scaled so that the page in focus is clearly readable with connected pages shown at smaller scales to provide context.

3. STATEMENT OF THE PROBLEM

The Web is becoming a universal place for human knowledge and culture which has allowed sharing ideas and information on a scale never seen before. The publicly indexable World Wide Web as of 1999 contains about 800 million pages, encompassing about 6 terabytes of text data on about 3 million servers [82] [99]. One of the reasons for it's success is a standard user interface, always the same no matter what computer system is used to run the interface. Because of this standard interface, the user is not bothered with details of different operating systems, network protocols, and server locations. In addition, any user can create his or her own Web pages and make them point to any other one without any limitations. This is an important feature because it transforms the Web into a new communication and publishing medium, which can be accessible by everybody.

> One sometimes hears that the Internet characterized as the world's library for the digital age. This description does not stand up under even casual examination. The Internet and specially its collection of multimedia resources known as the World Wide Web was not designed to support the organized publication and retrieval of information, as libraries are [65, pg. 1].

The Web has evolved into what one can describe as a disorganized information space where collective output of the world's digital printing presses are stored. This information space contains not only books, and papers but also data, pictures, sound, video, and animation.

In summary, the Web is not a digital library. However, if it is to progress and succeed as a new method of communication, we need something similar to traditional library services to organize, access and protect networked information. Yet, the Web will not be similar to a traditional library, because the Web contains data that is contents are more generally separated than a standard collection. As a result, the librarian's categorization and selection abilities must be combined with the computer scientist's skills to computerize the task of arranging and collecting information. Combining these two professions abilities and perspectives allow this new medium to remain viable.

One of the main challenges that computer technology faces at this time is to organize information on the Web. In theory, software that automatically classifies and indexes collections of digital data can address these challenges. Because, automating information access has the advantage of directly exploiting the rapidly dropping costs of computers and avoiding the high expense and delays of human indexing.

But the rate of the web's growth has been and continues to be exponential. (Table 1) [102]. Measuring the Internet and in particular the Web, is a difficult task due to its highly changing nature. Today, there are more than 40 million computers in more than 200 countries connected to the Internet, many of them hosting Web servers [63].

The estimated number of Web servers ranges from 2.4 million according to NetSizer [97] (November 1998) to over 3 million according to the Netcraft [98] Web survey (October 1998).

Results Summary			
Month	# of Web Sites	% .com Sites	Hosts per Web Server
Jun-93	130	1.5	13,000 (3,846)
Dec-93	623	4.6	3,475 (963)
Jun-94	2,783	13.5	1,095 (255)
Dec-94	10,022	18.3	451 (99)
Jun-95	23,500	31.3	270 (46)
Jan-96	100,000	50	94 (17)
Jun-96	230,000	68	41
Jan-97	650,000	52.6	NA

Table 1
Growth of Web Between 1993 to 1997

During February 1999, random Internet Protocol (IP) addresses tested for a web server at the standard port by NEC Research Institute. There were currently about 4.3 billion possible IP addresses; some of these are unavailable while some are known to be unassigned. After testing 3.6 million IP addresses, survey found a Web server for one in every 269 requests, leading to an estimate of 16 million web serves in total [99]. For comparison, Netcraft [98] found 4.3 million web servers in February 1999 based on testing known host names. The estimate of 16 million servers is not very useful, because there are many web servers that would not normally be considered part of the publicly indexable Web. These include servers with authorization requirements (including firewalls), servers that respond with a default page, those with no content (sites 'coming soon' for example), web-hosting companies that present their home page on many IP addresses, printers, routers, proxies, mail servers, CD-ROM servers and other hardware that provides a web interface. NEC survey [99] manually classified all servers and removed servers that are not part of the publicly indexable Web. Their resulting estimate of the number of servers on the publicly indexable Web as of February 1999 is 2.8 million [99].

To estimate the number of indexable Web pages, survey [99] crawled all the pages on the first 2,500 random web servers. The mean number of pages per server was 289, leading to an estimate of the number of pages on the publicly indexable web of about 800 million. Many sites have few pages, and a few sites have vast numbers of pages, which limits the accuracy of the estimate [99]. The true value could be higher because of very rare sites that have millions of pages, or because some sites could not be crawled completely because of errors. The mean size of pages was 18.7 kilobytes or 7.3 Kbytes after reducing the pages to only the textual content (removing HTML tags, comments and extra white space) [99]. This allows an estimate of the amount of data on the publicly indexable Web: 15 terabytes (Tbytes) of pages, or 6 Tbytes of textual content [99]. Survey also estimated 62.8 images per web server and a mean image size of 15.2 Kbytes, leading to an estimate of 180 million images on the publicly indexable web and a total amount of image data of about 3 Tbytes [99].

Other estimations were made by sampling 0.1% of all Internet numeric addresses obtaining about 2 million unique Web sites or by counting domain names starting with "www" which in July 1998 were 780,000 according to the Internet Domain survey.

However, since not all Web servers have this prefix, the real number is even higher. Considering that in July 1998 the number of Internet hosts was estimated at 36.7 million, there is about one Web server per every ten computers connected to the Internet [63, pg. 370].

In two interesting articles, Bray and Woodruff studied different statistical measures of the Web. The first study uses 11 million pages while second uses 2.6 million pages, with both sets gathered in November 1995. Their characterization of Web pages is partially reproduced in the following paragraphs. A first question is how many different institutions (not Web servers) maintain Web data. This number is smaller than the number of servers, because many places have multiple servers. The exact number is unknown, but should be more than 40% of the number of Web servers (this percentage was the value back in 1995). The exact number of Web pages is also not known. Estimates at the beginning of 1998 ranged from 200 to 320 million, with 800 million according to NEC research institute in February 1999 [99]. The later study used 20,000 random queries based on a lexicon of 400,000 words extracted from Yahoo! [92]. Those queries were submitted to four search engines and the union of all the answers covered about 70% of the Web. On the other hand, it is estimated that the 30,000 largest Web sites (about 1% of the Web) account for approximately 50% of all Web pages, and an average page has between five and 15 hyperlinks (more than 8 links on average) [63, pg. 370, 371].

Figure 2 [63, pg. 370] gives an approximation of how the number of Web servers and the number of pages have changed in recent years. Between 1997 and 1998, the size of the Web doubled in nine months and is currently growing at a rate of 20 million pages per month.

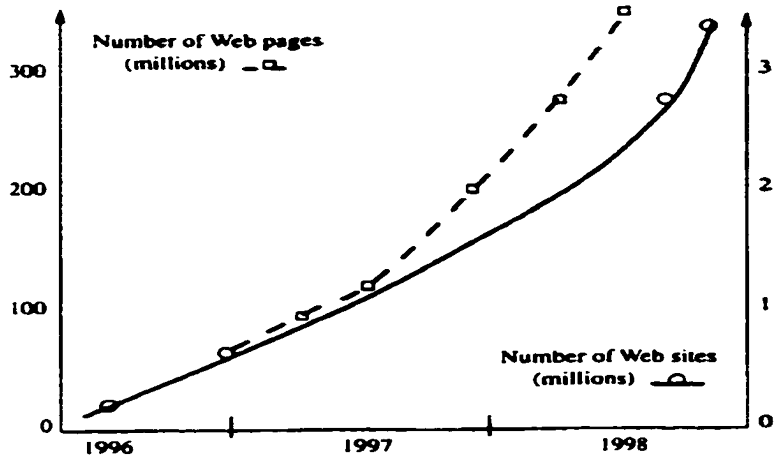


Figure 2

Approximate growth of the Web

4. OVERVIEW OF THE PROBLEM

Despite so much success, the Web has introduced new problems of its own:

- Due to the intrinsic nature of the Web, data spans over many computers and platforms. These computers are interconnected with no predefined topology and the available bandwidth and reliability on the network interconnections varies widely.
- Due to the Internet dynamics, new computers and data can be added or removed easily (it is estimated that 40% of the Web changes every month). There are also dangling links and relocation problems when domain or file names change or disappear.
- The exponential growth of the Web poses scaling issues that are difficult to cope with.
- Most people say that the Web is a distributed hypertext. However, this is not exactly so. Any hypertext has a conceptual model behind it, which organizes and adds consistency to the data and the hyperlinks. That is hardly true in the Web, even for individual documents. In addition, each HTML page is not well structured and some people use the term semi-structured data. Moreover, much Web data is repeated (mirrored or copied) or very similar.
- Approximately 30% of Web pages are (near) duplicates. Semantic redundancy can be even larger.

- The Web can be considered as a new publishing medium. However, there is, in most cases, no editorial process. So, data can be false, invalid (for example, because it is too old), poorly written or, typically, with many errors from different sources (typos, grammatical mistakes, OCR errors etc.). Preliminary studies show that the number of words with typos can range from 1 in 200 for common words to 1 in 3 for foreign surnames.
- In addition to having to deal with multiple media types and hence with multiple formats, there are also different languages and, what is worse, different alphabets, some of them very large.

Finding useful information on the Web is frequently a tedious and difficult task. For instance, to satisfy information need, the user might navigate the space of Web links searching for information of interest. However, since the hyperspace is vast and almost unknown, such a navigation task is usually inefficient. For naive users, the problem becomes harder, which might entirely frustrate all their efforts. The main obstacle is the absence of a well-defined underlying data model for the Web, which implies that information definition and structure is frequently low quality [63, pg. 368, 369].

There are no easy solutions to these problems and not all the problems (such as the different data types and unsatisfactory data quality) are solvable easily by software improvements. In fact, many of these problems should not change (as in the case of language diversity) because they are features part of the human nature.

The second class of problems is those faced by the user during the interaction with the retrieval system. There are basically two problems: 1) how to specify a query and 2) how to interpret the answer provided by the

> system. Without taking into account the semantic content of a document, it is not easy to precisely specify a query, unless it is very simple. Further, even if the user is able to pose the query, the answer might be a thousand Web pages. How do we handle a large answer? How do we rank the documents? How do we select the documents that really are of interest to the user? In addition, a single document could be large. How do we browse efficiently in large documents? [63, pg. 369]

These problems have brought renewed attention in information retrieval and its techniques as promising solution also known as search engines. The overall challenge, despite of the natural problems introduced by the Web, is to give a proper query to the search engine, and retrieve a set of pages that make sense to user.

Today searching for Web sites is one of the most common tasks performed on the Web. About 85% of users use search engines to locate information, and several search engines consistently rank among the top ten sites accessed on the Web [82]. In 1990, researchers at McGill University in Montreal developed Archie, the first Internet search engine. Archie searches the files of Internet FTP servers. Two other early engines search gopher servers: Veronica, developed in 1992 at the University of Nevada; and Jughead, developed in 1993 at the University of Utah [3].

> Today popular search engines are traditionally consisting of three components: the crawler, the index, and the search software. Most search engines use centralized crawler-indexer architecture. Crawlers are programs (software agents) that traverse the Web sending new or updated pages to a main server where they are indexed. Crawlers are also called

robots, spiders, wanderers, walkers, and knowbots. In spite of their name, a crawler does not actually move to and run on remote machines, rather the crawler runs on a local system and sends requests to remote Web servers. Periodically, they dispatch programs to every site they can identify on the Web, each site being a set of documents, called pages, which can be accessed over the network. The Web crawlers download and examine these pages and extract indexing information that can be used to describe them [63, pg. 373].

This process, details of which differ between search engines- may include simply finding most of the words that appear in Web pages or going through very complicated analyses to identify key words and phrases. These data are then stored in the search engine's database, along with an address; formed as a uniform resource locator (URL), that represents where the file stays. The index is used in a centralized way to answer queries submitted from different places in the Web. A user then using a browser to submit queries to the search engine's database [63].

Generally, a user enters a keyword (or keywords along with Boolean modifiers, such as 'and', 'or', 'not') into a search engine, which then searches indexed Web pages for the keywords. There are two important features of the user interface of search engines: the query interface and the answer interface. The basic query interface is a box where one or more words can be typed. Although a user would assume that a given sequence of words represent the same query in all search engines, it does not. For example, in AltaVista [90] a sequence of words is a reference to the union of all web pages having at least one of those words, while in HotBot [91] it is a reference to the Web pages having all the words [63].

As a result, search engine displays a list of Web pages according to query; the display usually contains of a list of the top ten ranked Web pages. Each entered page in this list includes some information about the page that it represents. Usually, the information includes the date, size, URL when the page was indexed, and a couple of lines with its content such as title, or headings from that page. In order to connect to the sites identified by the search engines a user has to click on to URLs. To determine in which order to list of documents the system uses an algorithm to rank sites that contain the keyword. Index based search engines have been the primary mechanism by which user search for information. With today's technology the largest such search engines have ability to store and index much of the Web. Such engines can therefore create enormous indices that allow user quickly retrieve the set of all Web pages containing a given word or string.

The number of queries submitted per day to AltaVista is over 13 million [63]. Users select a search engine primarily based on ease of use, speed, coverage, relevance of the answer, and habit. The main purposes are research, leisure, business, and education. When searching 25% of the users use a single keyword, and on average their queries have only two or three terms. In addition, about 15% of the users restrict the search to a predefined topic and most of them (nearly 80%) do not modify query. Most users (about 85%) only look at the first screen with results and 64% of the queries are unique [63, pg. 390].

Figure 3 [63, pg. 374] shows the software architecture of a search engine based on the AltaVista [90] architecture. It has two parts: one that deals with the users, consisting of the user interface and the query engine and another that consists of the crawler and indexer modules. In 1998, the overall AltaVista system was running on 20 multi-processor machines, all of them having more than 130 Gb of RAM and over 500 Gb of disk space. Only the query engine uses more than 75% of these resources [63].

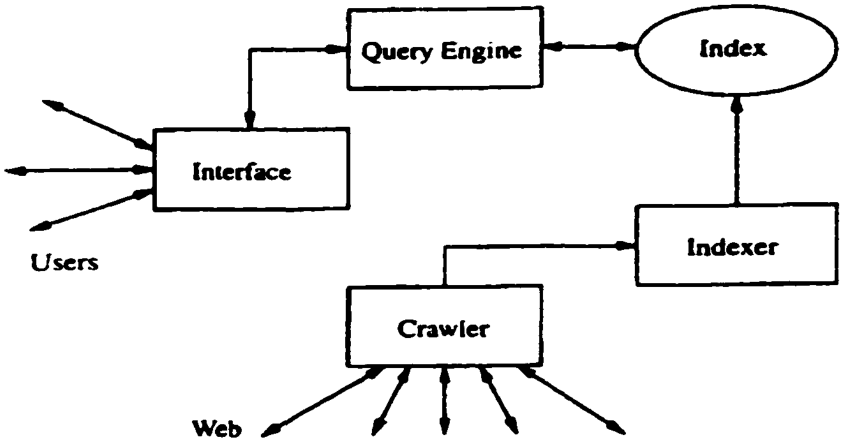


Figure 3

Typical crawler-indexer architecture

Due to the highly dynamic nature of the Web, the full filled communication links, and the high load at Web servers, the primary problem faced by this architecture is the collecting the data.

> Another problem faced by automated indexing is that most search engines recognize text only. The increasing interest in the Web, though, has come about because of the medium's ability to display images, whether graphics or video. Some research has moved forward toward finding colors or patterns within images. But no group can deduce the underlying meaning and cultural significance of an image. At the same time, the way information is structured on the Web is changing so that Web crawler often cannot examine it. Many Web pages are no longer static files that can be analyzed and indexed by such programs. In many cases, the information displayed in a document is computed by the Web site during a search in response to the user's request. The site might assemble a map, a table, and a text document from different areas of its database, a disparate collection of information that conforms to the user's query. A newspaper's Web site, for instance, might allow a reader to specify that only stories on the Information technology are displayed in a personalized version of the paper. The database of stories from which this document is put together could not be searched by a Web crawler that visits the site [65, pg. 4].

Another important problem is the amount of the data. In fact, the crawler-indexer architecture may not be able to encounter with Web growth in the near future. Especially important is to coordinate balancing between the different duties of a search engine, internally (ranking, displaying queries and indexing) and externally (crawling). Instead of working with indexes Directories [3] work with descriptions of Web pages submitted by either Webmasters or editors who have reviewed the pages, such as LookSmart [103] and

Yahoo! [92]. Since the Web still needs standards that would help automated indexing, as a result documents on the Web are not structured so that crawlers can reliably collect the usual information that a human indexer might find through a cursory inspection: author, date of publication, length of text and subject matter (This information known as a metadata). A Web crawler might turn up the desired article authored by Jane Doe. But it might also find thousands of other articles in which such a common name is mentioned in the text or in a bibliographic reference [65, pg. 3].

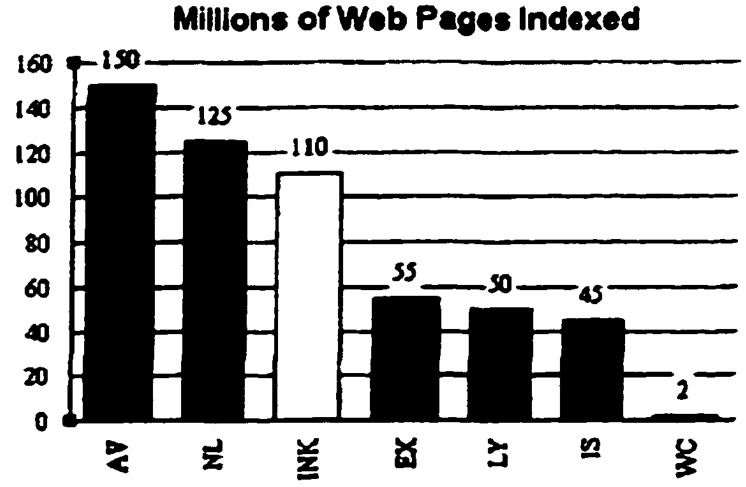
The largest search engines, considering Web coverage in June 1998, were AltaVista [90], HotBot [91], Northern Light [94], and Excite [93], in that order. According to recent studies, these engines cover 28-55% or 14-34% of all Web pages, whose number was estimated at over 300 million in 1998. Table 2 [63, pg. 375] lists the most important search engines and their estimated sizes along with their corresponding URLs. Beware that the same internal engine powers some search engines. For example, HotBot [91], GoTo [104], and Microsoft [105] are powered by Inktomi [99] and Magellan [106] by Excite's [93] internal engine [63].

Search Engine	URL	Web Pages Indexed
AltaVista	www.altavista.com	140
AOL Netfind	www.aol.com/netfind/	-
Excite	www.excite.com	55
Google	www.google.com	25
GoTo	goto.com	-
HotBot	www.hotbot.com	110
Infoseek	www.infoseek.com	30
Lycos	www.lycos.com	30
Magellan	www.mckinley.com	55
Microsoft	search.msn.com	-
NorthernLight	www.northernlight.com	67
WebCrawler	www.webcrawler.com	2

Table 2
URLs and estimated size (millions) of the largest search engines (May 1998)

Figure 4 [100] shows the size of each search engine's index. The larger the index, the more likely the search engine will be a comprehensive record of the Web. That's primarily helpful for those looking for rare material.

Sizes are as reported by each search engine and as of February 1, 1999.



KEY: AV=AltaVista, NL=Northern Light, INK=Inktomi, EX=Excite
LY=Lycos, IS=Infoseek, WC=WebCrawler

Figure 4

Size of the Search Engine's index

AltaVista [90] leads the search engines, followed by Northern Light [94], then Inktomi [99]. These search engines are also favorites among librarians and researchers, and index size has much to do with this.

In April 1998, a Science study estimated that there were 320 million indexable pages on the Web. Figure 5 [100] shows what percentage of the Web is covered by search engine, using this estimate versus each search engine's size in April 1998.

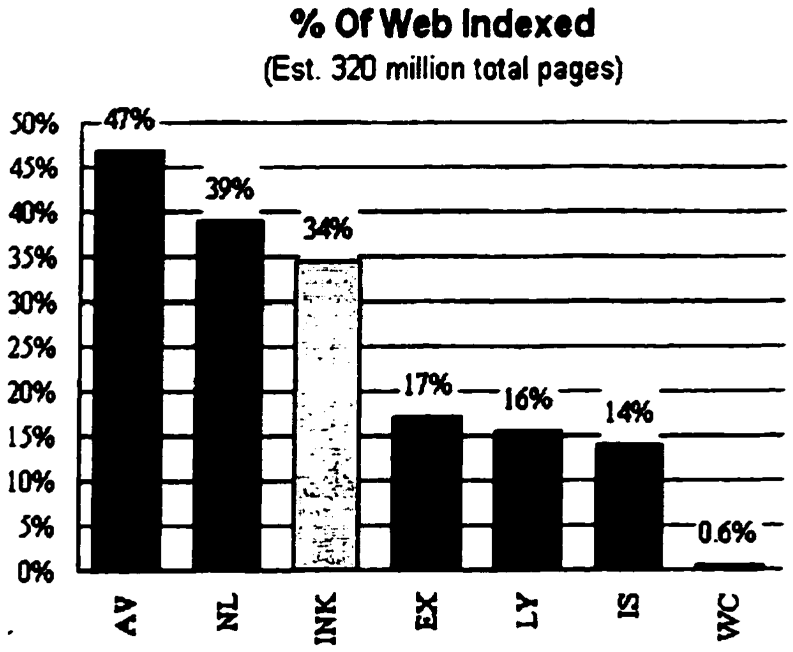


Figure 5

Percentage of the Web covered by Search engines

One of the alternatives to index search engines is to using Metasearchers. Metasearchers are Web servers that send a given query to several search engines, Web directories and other databases, collect the answers and combine them. Examples are Metacrawler [95] and SavySearch [96]. The main advantage of Metasearchers is the ability to combine the results of many search engine results using a single common interface. Metasearchers differ from each other in how ranking is performed in the combined result, and how well they translate the user query to specific query language of each search engine or Web directory. Table 3 [63, pg. 388] shows the URLs of the main metasearch engines as well as the number of search engines, Web directories and other databases that they search.

Metasearcher	URL	Sources Used
Cyber 411	www.cyber411.com	14
Dogpile	www.dogpile.com	25
Highway61	www.highway61.com	5
Inference Find	www.infind.com	6
Mamma	www.mamma.com	7
MetaCrawler	www.metacrawler.com	7
MetaFind	www.metafind.com	7
MetaMiner	www.miner.uol.com.br	13
MetaSearch	www.metasearch.com	-
SavySearch	savy.cs.colostate.edu:2000	>13

Table 3
URLs of metasearchers and number of sources that they use (October 1998)

The advantages of metasearchers are that the results can be sorted by different characteristics such as search engine, keyword, etc. which can be more informative than the results of a single search engine. Therefore, browsing the results should be easier. On the other hand, the results are not necessarily all the Web pages that matching the query, because the number of results per search engine retrieved by the metasearcher is limited to some number. Although, pages returned by more than one search engine should be more relevant to the given query. But even with these improvements, results would be thousands of pages, including irrelevant pages [63].

In short, the variety of materials on the Web are far beyond the scope of the traditional library. Existing search engines service millions of queries a day. Yet it has become clear that they are less than ideal for retrieving an information on the Web. Because of the great number of distributed information, Web users want direction about where to spend the limited amount of time they have to research for a subject. They are increasingly faced with the problem of filtering and interpreting huge number of information. From this bulk of data they need to extract information which will allow them to make right decisions. They may need to know the three "best" documents for a given purpose. They need a system to organize and interpret the information that is necessary to support their tasks. The Visual Analysis System gives solutions to these problems using data mining and information visualization tools using the link structure of the Web.

5. PROJECT OVERVIEW

The growth in the volume, number and availability of information sources has increased interest in data mining. Data mining techniques allows users to interactively search and explore hidden knowledge and information within these information sources. Visualization augments this search process. Combining visualization techniques with data mining and analysis forms a unique and powerful new paradigm for search and discovery – visual analysis [12].

In order to be useful, visualization and data mining techniques must improve existing discovery and exploration methods. Since data mining techniques try to give users necessary tools to think and explore, effective visual interfaces for these systems remain an important component due to human perceptual issues. Therefore, we need to combine visual interface with information retrieval and analysis engines for better visual analysis.

Because of its underlying structure and quantity of information, the World Wide Web is one of the best examples of an information space where users need support. The structure of the Web has evolved rather than been designed, and quantity of information is both very large and it is changing rapidly. Locating and retrieving needed and relevant information on the Web has become important task. Web-based information retrieval systems, also known as Web search engines, have different measures of success. Because any topic of query will return hundreds or thousands of pages, it's very frustrating for users to view a long list of pages that are not related to their search intentions and manually search outwards from this point without any information to guide them. Therefore, this manual search is almost impossible. Here, we could combine visualization techniques and data mining techniques to augment query results to give user an additional feedback about this big information space visually, extract hidden predictive information and derive new information from the given databases. This retrieved information can be easily visualized using data mining techniques along with existing information. Visualization is used in data mining very often as an important tool for generate initial views, convey the results of an analysis and navigation.

Visual Analysis System is user-centric, allowing users to interact with the information stored in large sets of documents. Visualization is an integral part of the overall process. VAS focuses primarily on visualization and visual interfaces in support of information retrieval and data mining, shortly visual analysis.

VAS covers two related categories. The first category includes collecting and analysis of new information and relationship in collected pages by clustering and displaying visually. Once the user has retrieved the pages that are related to his/her query, he/she often needs to analyze the different pieces of information in these documents and discover new information. Usually, the number of retrieved information is immense, which makes analysis and access difficult. Initially, we need to condense a large Web search topic to an appropriate size by identifying the topic's most definitive or authoritative Web pages. It means locate not only a set of relevant pages, but also those relevant pages of the highest quality. The second category examines hyperlinks that connect one page to another.

> This hyperlink structure contains enormous amount of latent human annotation that can help automatically infer notations of authority. Specifically, the creation of hyperlink by the author of a Web page represents an implicit endorsement of the page being pointed to. By mining the collective judgement contained in the set of such endorsement, we can gain a richer understanding of the relevance and quality of the Web's contents [1, pg. 60].

VAS uses an instinctive technique for finding the most focal, authoritative, sites on broad search topics by making use of hyperlinks. Hyperlinks draw together the hundreds of millions of pages into a web of knowledge. A link may also created just for navigational purposes ("Click this button to return to the main menu") or as an advertisement ("Best place to buy a car is only click away"). But when a large enough number is taken into account, these links do confer authority. In addition to these authoritative sites that have many recommendations, the Web also contains another type of page: hubs that points to authoritative pages. Hubs appear from professionally created lists on commercial sites to index of "My Favorite Links" on personal home pages. A valuable authority is a page that is pointed to by many good hubs and a useful hub is a page that points to many valuable authorities [59].

> There is, clearly, no definite universal system that controls the creation of Web pages and hyperlinks; how then can one determine authoritative forms of structure from the underlying link topology? Defining the way by which authority is conferred on the WWW is itself a difficult problem.

> Broad representation of any topic contains a number of prominent, authoritative pages and structure comes from the way in which such authorities are implicitly supported through hyperlinks. Most of the cases, authoritative pages on a common topic do not link one another directly. For example; Microsoft and Netscape are both good authorities for the topic of web browsers, but they do not link to each other, and usually they can only be grouped together through an intermediate layer of hub pages, which link to multiple, related authorities. Therefore, hubs and authorities are different types of pages that present a natural form of symbiosis: a good hub points to many good authorities, while a good authority is pointed to by many good hubs. But a good hub may not even be pointed to by any page; this means some of the most valuable structural contributions to the Web are being made by relatively unrecognized individuals [111, pg. 1].

The model also provides a natural way to present underlying structure between both the set of hubs, who often do not know directly of one another's existence. Other researchers refer to a closely coupled set of hubs and authorities as a community [60]. According to these researchers, this use of the term community is not meant to indicate that these structures have been constructed in a centralized or planned fashion. Rather, their experiments suggest that communities of hubs and authorities are a recurring consequence of the way in which creators of World Wide Web pages link to one another in the context of topics of widespread interest [60][100, pg.1].

> One can find closely coupled linked combination of hubs and authorities in a exceptionally different form of settings on the World Wide Web. Because these set of hubs and authorities have an intrinsic meaning in terms of the underlying link structure, we can define them even in the absence of a specific topic description. Other researchers suggest that this is a promising approach to WWW categorization. Rather than assuming a priori collection of subjects, we can let the link-based communities themselves define the prevailing topics, niches, and user populations of interest on the WWW. They also point out to important to problem that the issue here is not simply to partition the WWW into focused groups of this sort; the full representation of any such group on the WWW is typically enormous, and this small set of related authorities must serve the critical function of providing a compact yet informative representation of a much larger underlying population [100, pg. 2].

Although the main goals here are to understand the primary idea behind of structure that apply in a global sense on the WWW, this thesis uses analysis techniques to examine only a few thousand Web pages, collected by the Visual Analysis System. First, the system use the data mining techniques to locate the best authorities, to see which pages point to them and call those locations good hubs. Second, it takes the hub information to clarify the authorities, and determines where the best hubs point most heavily and call these good authorities.

> Web's link structure can be use in any of different methods to show notions of authority, some much more effective than others can. Because the link structure implies an underlying social structure in the way that pages links are created, an understanding of this social organization can provide us with the most leverage. The goal of designing algorithms for mining link information here is to take advantage of what we can observe about the Web's intrinsic social organization [1, pg. 60].

The system might best be understood in visual terms. This is the second category in VAS project: Browsing through the World Wide Web hyperspace without becoming lost, based on a visual representation of the hyperspace structure. While browsing, users often would like to view the names of the documents and how they are linked to each other without actually opening and reading each document. With VAS, after visually understanding the hyperspace structure and contents, the user could decide the open and read none, some or all the documents represented on the visual space. And VAS uses one of the best ways to let users to navigate in hypermedia systems with an overview diagram of the information space.

> The vision of visual data mining system stems from the following principles: Simplicity, user autonomy, reliability, reusability, availability, and security. A visual data mining system must be syntactically simple to be useful. Simple to learn means use of intuitive and friendly input mechanisms as well as instinctive and easy to interpret output knowledge. Simple to apply means an effective discourse between humans and information. Simple to retrieve or recall means a customized data structure

to facilitate fast and reliable searches. Simple to execute means a minimum number of steps needed to achieve the results. It shouldn't impose knowledge on its users, but instead guide them through the mining process to draw conclusions. It must provide estimated error accuracy of the projected information for each step of mining process. It must be adaptable to a variety of systems and environments to reduce the customization efforts [18, pg. 20].

6. METHODOLOGY

6.1. SYSTEM DESCRIPTION

Visual Analysis System is implemented using Visual C++ (Microsoft C/C++ compiler) [108] with MFC (Microsoft Foundation Classes) [107] in Windows - NT [109] platform.

Windows is a graphical user interface (GUI), and Windows applications can make full use of graphics and formatted text on the video display. This graphical interface is not only more attractive in visually, but it can also give a high level of information to the user. Graphics gives better utilization of screen, a visually rich environment for displaying information, and the possibility of WYSIWYG (what you see is what you get) video display of graphics. This graphical user interface is also the main source for user input. The display shows different icons, buttons and scroll bars. Using the keyboard or mouse, the user can manipulate these objects on the screen. Icons can be dragged, buttons can be pushed, and scroll bars can be scrolled. The interaction between the user and a program thus becomes easier. Instead of using keyboard as a main input source, with graphical user interface, users directly interacts with the objects on the display [85].

MFC [107] is the C++ class library Microsoft [110] provides to place an object-oriented wrapper around the Windows Application Interface. MFC is an application structure, simply a collection of classes. MFC helps define the structure of an application and handles many routine calls on the application's behalf. MFC encapsulates virtually every part of a program's operation. For example, MFC's document/view architecture builds a powerful infrastructure on top of the API that separates a program's data from graphical representations, or views, of that data [86].

MFC provides an object oriented application interface to the Windows operating system that supports reusability, self-containment, and other principals of object oriented programming without overwhelming improper overhead on the system or unnecessarily adding to an application's memory requirements. It accomplishes this by using classes to encapsulate windows, dialog boxes, and other objects and by including key virtual functions that can be overridden to replace the behavior of derived classes [86].

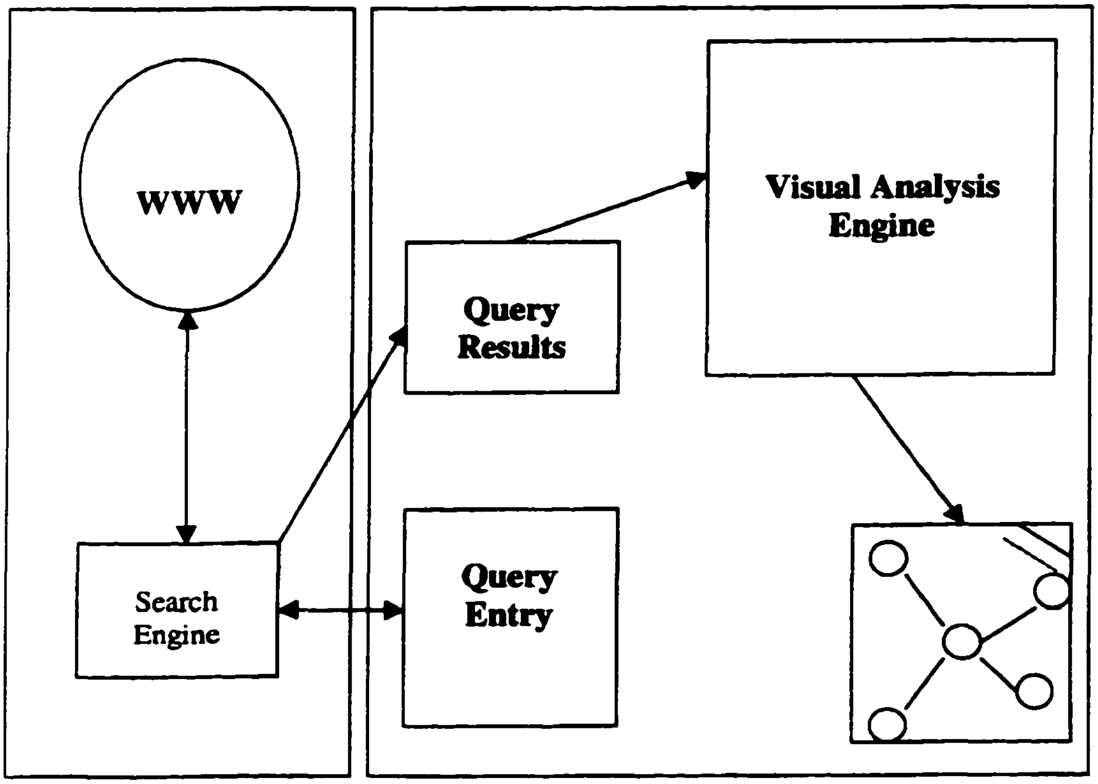
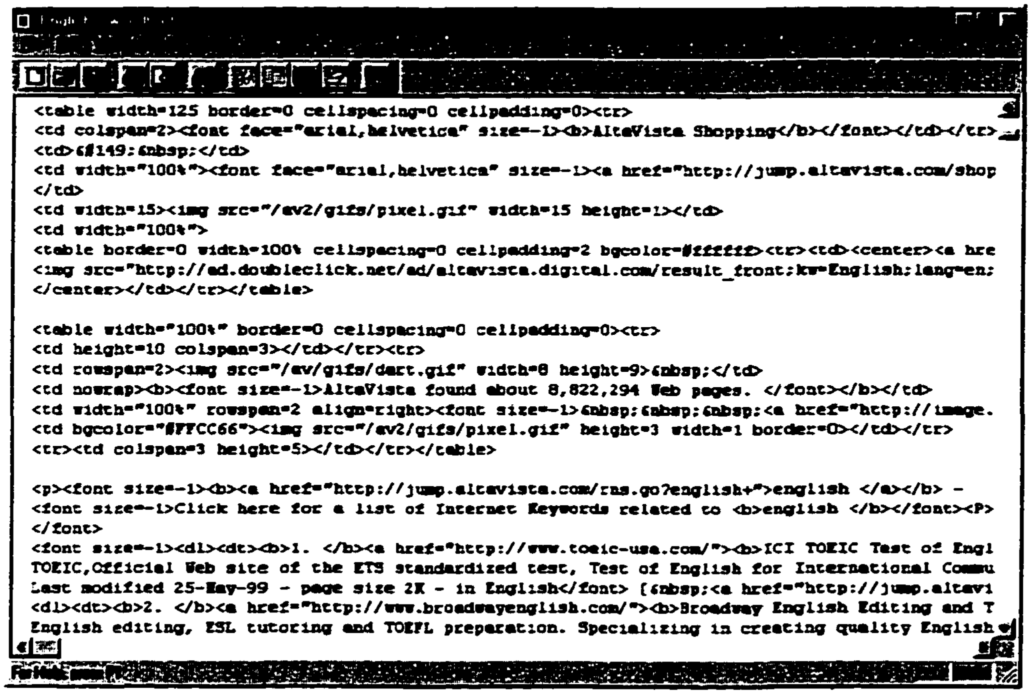


Figure 6
Visual Analysis System (VAS) Architecture

The VAS system is made up of several components (Figure 6). Under this visual data mining architecture, the system receives a user's query as a keyword and sends the query to one of the search engines. The algorithm then starts constructing a list of pages in which VAS will search for hubs and authorities. To construct the list, the system first uses the query terms to collect a *root set* [59] of pages. For any query of a topic, VAS first collects a list of pages, which is about 200, from a standard index based search engine such as Alta Vista [90]. This set of pages may not necessarily contain authoritative pages. However, since many of these pages are relevant to the search topic according to search engine, some of them to have links to most of the authorities. After retrieving the best matches, which serve as a starting point for VAS, the system starts its own search by following links in these pages to retrieve a *second level set* [59]. The resulting collection will typically contain between 1000 and 3000 links. Now, VAS can expand the root set into base set by including all the pages that the root-set pages link to, and all pages that link to a page in the root set, up to certain number. This method of defining authorities follows the intuition that the influence of authoritative pages derives typically from the support of many relevant pages that are not in themselves prominent [59]. VAS updates the authority and hub weights as follows: If many good hubs point to a page, it increases its authority weight.

VAS uses MFC's document/view architecture to separates program's data from graphical displays. Program's data contains a big collection of web pages as HTML (HyperText Markup Language) format (Figure 7). Using parsing techniques, VAS collects URLs from this file to create a new list of URLs and stores its results in array of pointers. After calculating authorities and hubs, VAS links this data to its view classes and stores these results in two-dimensional array, for matrix implementation of graph and spring algorithm.



```
<table width=125 border=0 cellspacing=0 cellpadding=0><tr>
<td colspan=2><font face="arial,helvetica" size=-1><b>AltaVista Shopping</b></font></td></tr>
<td>&#149;&nbsp;</td>
<td width="100%"><font face="arial,helvetica" size=-1><a href="http://jump.altavista.com/shop
</td>
<td width=15><img src="/av2/gifs/pixel.gif" width=15 height=1></td>
<td width="100%">
<table border=0 width=100% cellspacing=0 cellpadding=2 bgcolor=#ffffff><tr><td><center><a hre
<img src="http://ad.doubleclick.net/ad/altavista.digital.com/result_front;kw=English;lang=en;
</center></td></tr></table>

<table width="100%" border=0 cellspacing=0 cellpadding=0><tr>
<td height=10 colspan=3></td></tr><tr>
<td rowspan=2><img src="/av/gifs/dart.gif" width=8 height=9>&nbsp;</td>
<td nowrap><b><font size=-1>AltaVista found about 8,822,294 Web pages. </font></b></td>
<td width="100%" rowspan=2 align=right><font size=-1>&nbsp;&nbsp;&nbsp;<a href="http://image.
<td bgcolor="#FFCC66"><img src="/av2/gifs/pixel.gif" height=3 width=1 border=0></td></tr>
<tr><td colspan=3 height=5></td></tr></table>

<p><font size=-1><b><a href="http://jump.altavista.com/rns.go?english+">english </a></b> -
<font size=-1>Click here for a list of Internet Keywords related to <b>english </b></font><P>
</font>
<font size=-1><dl><dt><b>1. </b><a href="http://www.toeic-usa.com/"><b>ICI TOEIC Test of Engl
TOEIC,Official Web site of the ETS standardized test, Test of English for International Commu
Last modified 25-May-99 - page size 2K - in English</font> [&nbsp;<a href="http://jump.altavi
<dl><dt><b>2. </b><a href="http://www.broadwayenglish.com/"><b>Broadway English Editing and T
English editing, ESL tutoring and TOEFL preparation. Specializing in creating quality English
```

Figure 7

Collected Web Pages as HTML format

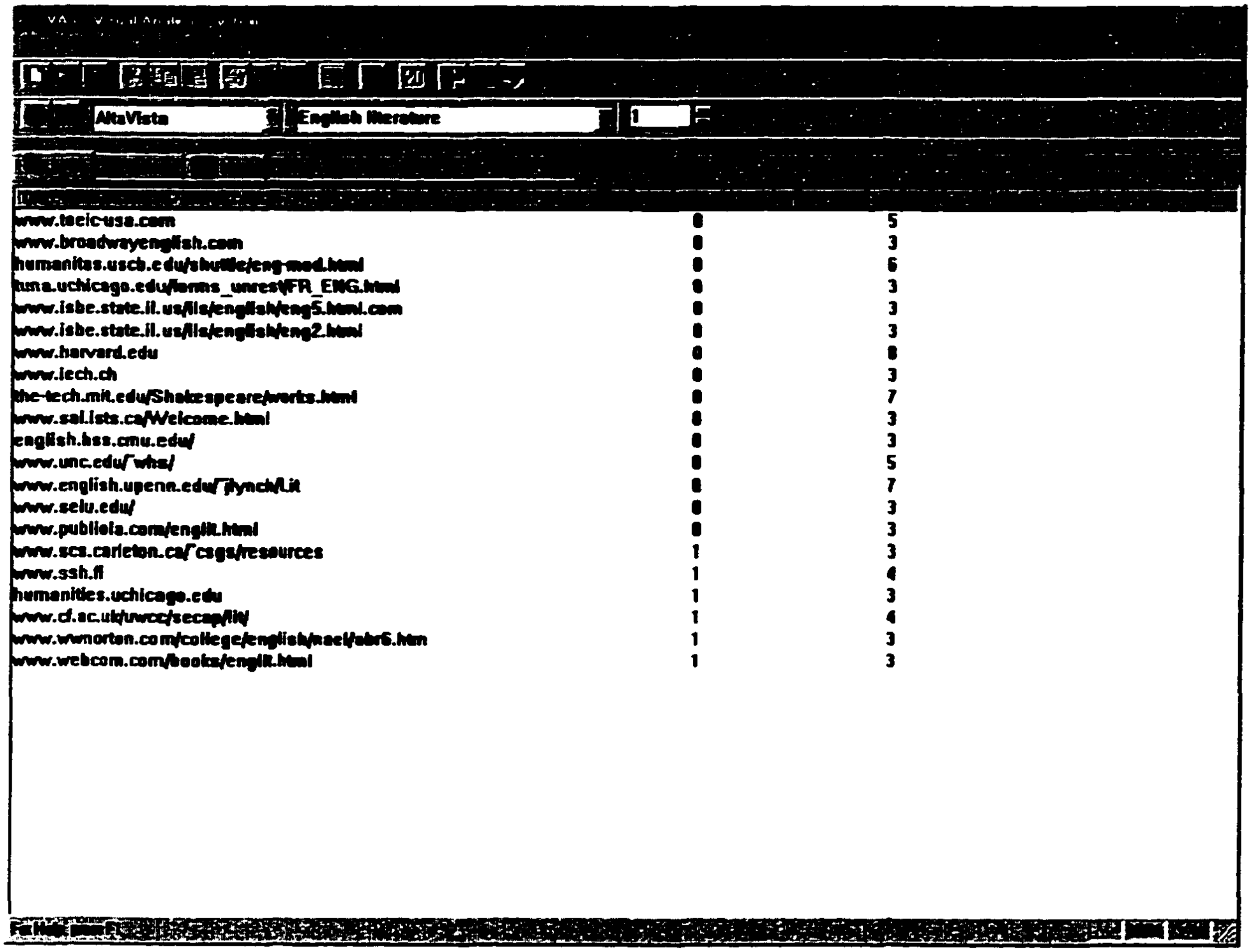
The algorithm might best be understood in visual term. Visualization space contains:

- Visual presentation of results
- Selecting by pointing
- Immediate and continues feedback.

For visual presentations graph layout criteria center on how quickly and clearly the meaning of the diagram are conveyed to the viewer, i.e. the readability of the graph. Graph drawing algorithms have as their goal the layout and presentation of an inherently mathematical entity in a manner, which meets various criteria for human observation. The creative aspects of a layout determine its readability and can be formulated as optimization goals for the drawing algorithm [50, pg. 10].

Network nodes can positioned in two dimensions using a graph layout algorithm [53] based on the spring metaphor which is similar to Kamada and Kawai's [54] two-dimensional network layout algorithm. As with other spring algorithms, nodes are treated as connectors and spring length and strength among connectors is derived from network link distances. Varying spring length and strength allows layouts, which are useful for user interaction and visually reveal clustering and connectivity among nodes [50, pg. 10]. The size and density of visual space requires viewing and navigation tools that allow users to perceive the overall structure of documents relations, explore smaller regions in detail, and select and view individual documents. The overview nodes are landmarks in that they supply information about both the content and structure of the database. This helps attenuate disorientation by providing the context for the individual network nodes, which are in view [50, pg. 11].

VAS combines three display views, Links View (Figure 8), Web Browser View, and Model View (2D – Graph) (Figure 9) on its main window. Links View contains a short list consisting of the pages with the largest hub weights and the pages with the largest authority weights for the given search topic (Figure 8). Links View uses three columns for displaying results as a list: URL, level, and weight. URL, lists the best authoritative pages, level indicates, where the authority was found, and weight displays weights of each page to give a user more information about authoritative pages.



AltaVista | English literature | 1

www.toeic-usa.com	0	5
www.broadwayenglish.com	0	3
humanitas.ucsb.edu/shuttle/eng-mod.html	0	6
tuna.uchicago.edu/forms_unrest/FR_ENG.html	0	3
www.isbe.state.il.us/ils/english/eng5.html.com	0	3
www.isbe.state.il.us/ils/english/eng2.html	0	3
www.harvard.edu	0	8
www.iech.ch	0	3
the-tech.mit.edu/Shakespeare/works.html	0	7
www.sal.lsts.ca/Welcome.html	0	3
english.hss.cmu.edu/	0	3
www.unc.edu/~whs/	0	5
www.english.upenn.edu/~jlynch/Lit	0	7
www.selu.edu/	0	3
www.publiola.com/englit.html	0	3
www.scs.carleton.ca/~csgs/resources	1	3
www.ssh.fi	1	4
humanities.uchicago.edu	1	3
www.cf.ac.uk/uwcc/secap/lit/	1	4
www.wwnorton.com/college/english/nael/abr6.htm	1	3
www.webcom.com/books/englit.html	1	3

Figure 8

VAS – Results as a list

Model view displays results as nodes using graph and spring algorithms. If there are links between these authoritative pages, Model view displays these connections as edges on graph layout. Bigger nodes indicate the best authoritative pages, and their color indicates levels of the pages, which they found. Blue nodes indicate root level, and red nodes indicate second level. Viewing this Web abstraction as a directed graph, consisting of a set of nodes with directed edges between certain nodes pair's, helps visualize the information sources on the Web (Figure 9).

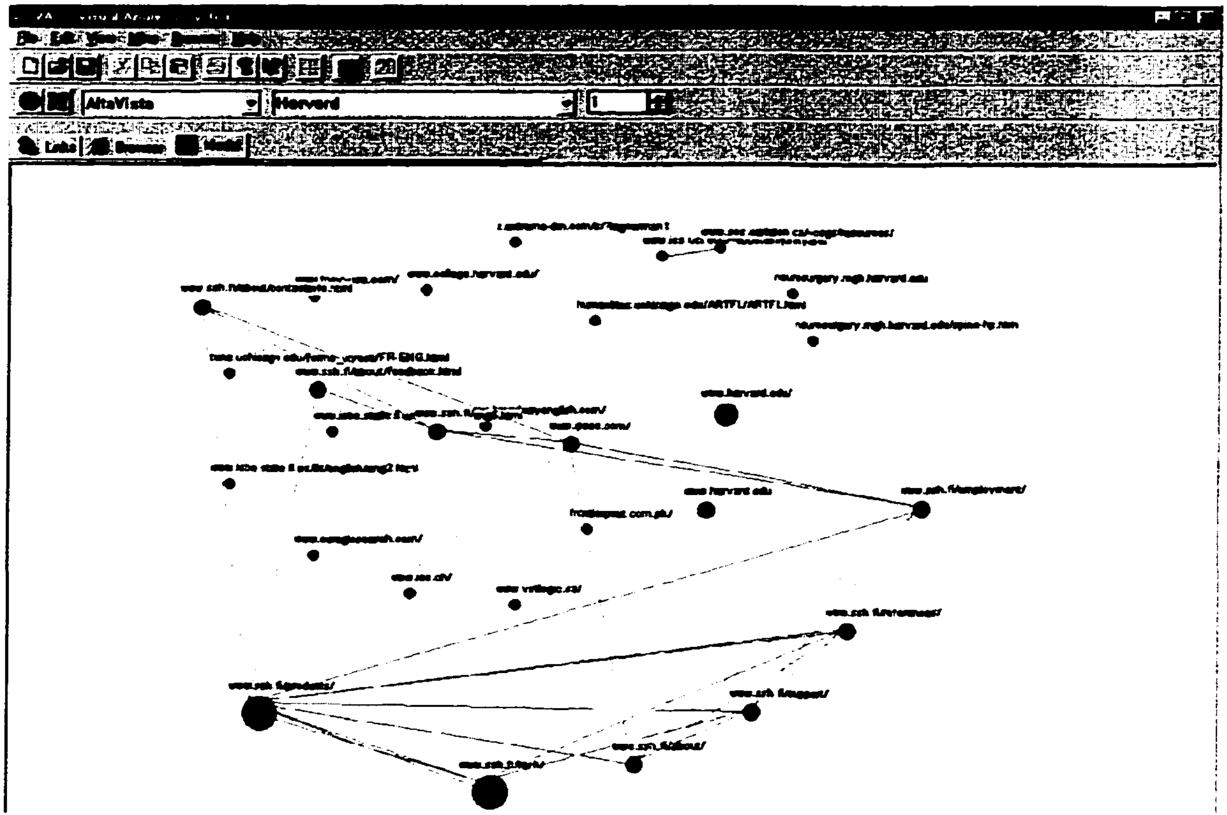


Figure 9

VAS – 2D View

Algorithm behind the Visual Analysis System applies three main steps:

Algorithm behind the Visual Analysis System applies three main steps:

(1) A collecting part, which constructs a collection of several thousand Web pages likely to be rich in relevant authorities (Figure 7).

(2) A weight-calculation part, which determines numerical weights of hub and authority.

(3) A visualization component, which display results as a list (Figure 8) and a graph (Figure 9) showing connections between pages.

6.2. COMPUTING HUBS AND AUTHORITIES

J. M. Kleinberg [59] developed a set of algorithmic tools for extracting information from the link structures of such environments, and demonstrated their effectiveness in a variety of contexts on the World Wide Web. He proposed and tested an algorithmic formulation of the notion of authority, based on the relationship between a set of relevant authoritative pages and the set of hub pages that join them in the link structure. His formulation has connections to eigenvectors of certain matrices associated with the link graph. This motivates additional heuristics for clustering and for computing a type of link-based similarity among hyperlinked documents. This section describes Kleinberg's algorithm to calculate hubs and authorities [59, pg. 4-10]. Visual analysis system applies these algorithms to calculate hubs and authorities.

According to Kleinberg's algorithm we can view any collection V of hyperlinked pages as a directed graph $G = (V, E)$: the nodes correspond to the pages, and a directed edge $(p, q) \in E$ indicates the presence of a link from p to q. The out-degree of a node p is the number of nodes it has links to, and the in-degree of p is the number of nodes that have links to it. From a graph G, algorithm isolates small regions, or subgraphs, in the following way. If $W \subseteq V$ is a subset of the pages, G[W] denotes the graph induced on W: its nodes are the pages in W, and its edges correspond to all the links between pages inW.

Given a broad-topic of query, specified by a query string σ, to be able to determine authoritative pages by an analysis of the link structure; first system must determine the subgraph of the WWW on which algorithm will operate. The goal here is to focus the computational effort on relevant pages. Thus, for example, we could restrict the analysis to the set Q_σ of all pages containing the query string; but this has two significant drawbacks. First, this set may contain well over a million pages, and hence entail a considerable computational cost; and second, some or most of the best authorities may not belong to this set.

Ideally, system needs to focus on a collection S_σ of pages with the following properties:

(1) S_σ is relatively small.

(2) S_σ is rich in relevant pages.

(3) S_σ contains most (or many) of the strongest authorities.

By keeping S_σ small, system is able to afford the computational cost of applying non-trivial algorithms - by ensuring it is rich in relevant pages. This makes it easier to find good authorities, as these are likely to be heavily referenced within S_σ.

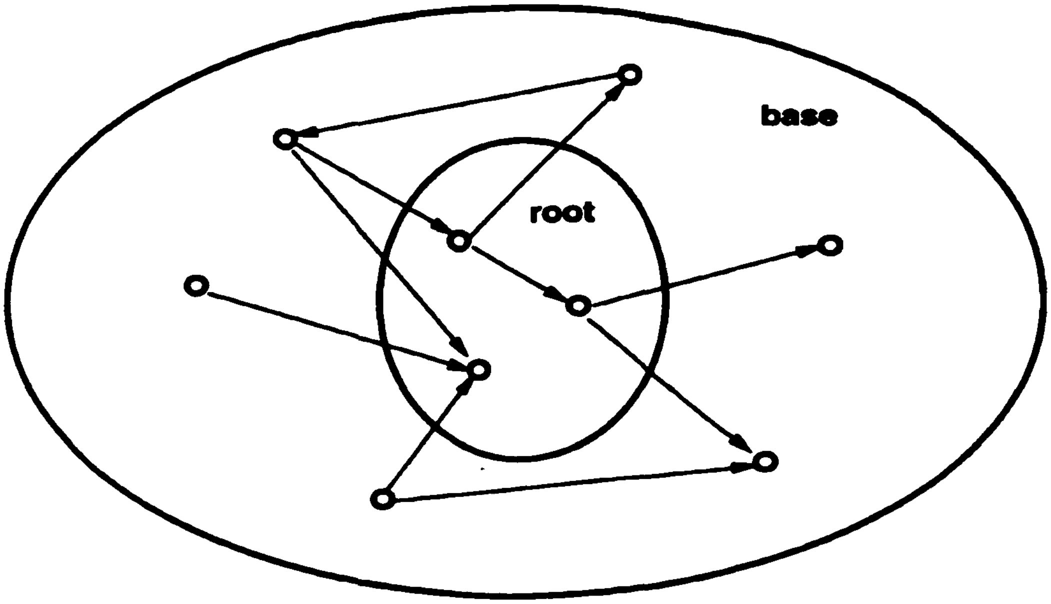


Figure 10

Expanding the root set into a base set

How can we find such a collection of pages? For a parameter t (typically set to about 200), algorithm first collects the t highest-ranked pages for the query σ from a text-based search engine such as AltaVista [90]. Kleinberg refers to these t pages as the root set R_σ (Figure 10 [59, pg. 5]). This root set satisfies (1) and (2) of the desiderata listed above, but it generally is far from satisfying (3). To see this, note that the top t pages returned by the text-based search engines will all contain the query string σ, and hence R_σ is clearly a subset of the collection Q_σ of all pages containing It is interesting to observe that there are often extremely few links between pages in R_σ, rendering it essentially "structureless". For example, VAS experimented root set for the query “English literature” contained 23 links between pages in different domains; the root set for the query "geometry" contained 36 links between pages in different domains.

These numbers are typical for a variety of the queries tried; they should be compared with the 200 x 199 = 39800 potential links that could exist between pages in the root set. Algorithm uses the root set R_σ, however, to produce a set of pages S_σ that will satisfy the conditions system is seeking. Consider a strong authority for the query topic - although it may well not be in the set R_σ, it is quite likely to be pointed to by at least one page in R_σ. Hence, system can increase the number of strong authorities in this subgraph by expanding R_σ along the links that enter and leave it. In concrete terms, Kleinberg defines the following procedure:

```
Subgraph($\sigma$,.$\varepsilon$, t, d)
$\sigma$: a query string.        $\varepsilon$: a text-based search engine.
t, d: natural numbers.
Let $R_\sigma$ denote the top t results of $\varepsilon$ on $\sigma$.
Set $S_\sigma := R_\sigma$    For each page $p \in R_\sigma$
Let $\Gamma^+$ (p) denote the set of all pages p points to.
Let $\Gamma^-$ (p) denote the set of all pages pointing to p.
Add all pages in $\Gamma^+$ (p) to $S_\sigma$
If $|\Gamma^-(p)| \leq d$ then
    Add all pages in $\Gamma^-$ (p)  to $S_\sigma$
Else
    Add an arbitrary set of d pages from $\Gamma^-$ (p)  to $S_\sigma$.
End
Return $S_\sigma$
```

Thus, system obtains S_σ by growing R_σ to include any page pointed to by a page in R_σ and any page that points to a page in R_σ - with the restriction that system allows a single page in R_σ to bring at most d pages pointing to it into S_σ. This latter point is crucial since a number of WWW pages are pointed to by several hundred thousand pages, and system can't include all of them in S_σ if we wish to keep it reasonably small.

Kleinberg refers to S_σ as the base set for σ; in his experiments system construct it by invoking the subgraph procedure with the search engine AltaVista [90], t = 200, and d = 50, he found that S_σ typically satisfies points (1), (2), and (3) above - its size is generally in the range 1000-5000; and a strong authority need only be referenced by any one of the 200 pages in the root set R_σ in order to be added to S_σ.

Kleinberg also define a heuristic that is very useful for offsetting the effect of links that serve purely a navigational function. First, let $G[S_\sigma]$ denote, as above, the subgraph induced on the pages in S_σ. He distinguishes between two types of links in $G[S_\sigma]$ that a link is "transverse" if it is between pages with different domain names, and "intrinsic" if it is between pages with the same domain name. By "domain name" means here that the first level in the URL string associated with a page. Since intrinsic links very often exist purely to allow for navigation of the infrastructure of a site, they convey much less information than transverse links about the authority of the pages they point to. Thus, system deletes all intrinsic links from the graph $G[S_\sigma]$, keeping only the edges corresponding to transverse links; this results in a graph $G\sigma$.

This is a very simple heuristic, but effective for avoiding many of the pathologies caused by treating navigational links in the same way as other links. There are other simple heuristics that can be valuable for eliminating links that do not seem intuitively to confer authority. One that is worth mentioning is based on the following observation. Suppose a large number of pages from a single domain all point to a single page p. Quite often this corresponds to a mass endorsement, advertisement, or some other type of "collusion" among the referring pages e.g. the phrase "This site designed by ..." and a corresponding link at the bottom of each page in a given domain. To eliminate this phenomenon, system can fix a parameter m (typically m ≈ 4-8) and only allow up to m pages from a single domain to point to any given page p. Again, this can be an effective heuristic in some cases, although Kleinberg did not employ it when running the experiments that follow neither the VAS.

The algorithm provides a small subgraph G_σ that is relatively focused on the query topic - it has many relevant pages, and strong authorities. System now turns to the problem of extracting these authorities from the overall collection of pages, purely through an analysis of the link structure of G_σ.

Approach of ranking purely by in-degree does typically work much better in the context of $G_{\sigma.}$ In some cases, it can produce uniformly high-quality results. However, the approach still retains some significant problems. For example, on the query “java”, the pages with the largest in-degree consisted of www.gamelan.com and java.sun.com, together with pages advertising for Caribbean vacations, and the home page of Amazon Books. This mixture is representative of the type of problem that arises with this simple ranking scheme: While the first two of these pages should certainly be viewed as "good" answers, the others are not relevant to the query topic; they have large in-degree but lack any thematic unity. The basic difficulty this exposes is the inherent tension that exists within the subgraph $G_{\sigma\,\text{between}}$ strong authorities and pages that are simply "universally popular"; system expects the latter type of pages to have large in-degree regardless of the underlying query topic.

One could wonder whether circumventing these problems requires making further use of the textual content of pages in the base set, rather than just the link structure of G_{σ}. But this is not the case - it is in fact possible to extract information more effectively from the links - and starts from the following observation. Authoritative pages relevant to the initial query should not only have large in-degree; since they are all authorities on a common topic, there should also be considerable overlap in the sets of pages that point to them. Thus, in addition to highly authoritative pages, we expect to find what could be called hub pages: these are pages that have links to multiple relevant authoritative pages. It is these hub pages that "pull together" authorities on a common topic, and allow system to throw out unrelated pages of large in-degree. (A skeletal example is depicted in Figure 11 [59, pg. 8]; in reality, of course, the picture is not nearly this clean.).

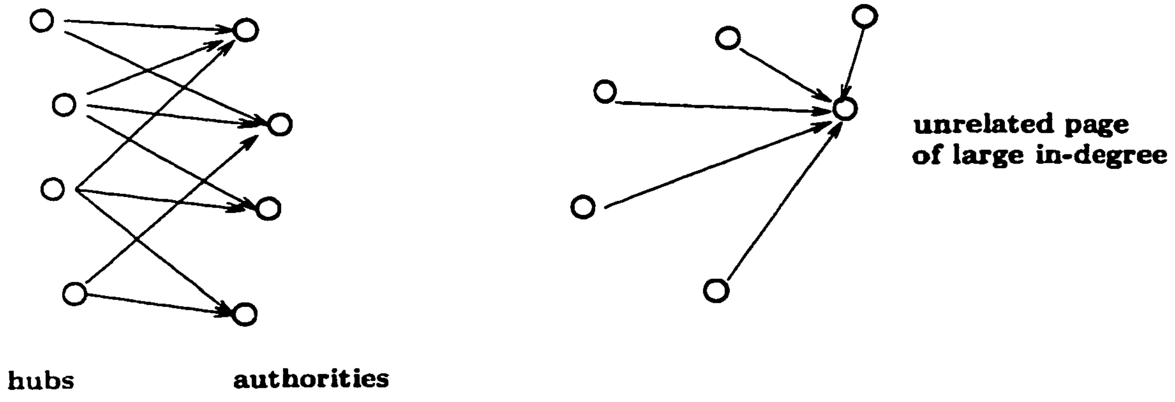


Figure 11

A densely linked set of hubs and authorities

Hubs and authorities exhibit what could be called a mutually reinforcing relationship: a good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs. Clearly, if we wish to identify hubs and authorities within the subgraph G_σ we need a method for breaking this circularity.

System makes use of the relationship between hubs and authorities via an iterative algorithm that maintains and updates numerical weights for each page. Thus, with each page p, system associates a non-negative authority weight $x^{<p>}$ and a non-negative hub weight $y^{<p>}$. We can maintain the invariant that the weights of each type are normalized so their squares sum to 1: $\sum_{p \in S\sigma} (x^{<p>})^2 = 1$, and : $\sum_{p \in S\sigma} (y^{<p>})^2 = 1$. System views the pages with larger x- and y- values as being "better" authorities and hubs respectively.

Numerically, it is natural to express the mutually reinforcing relationship between hubs and authorities as follows: If p points to many pages with large x-values, then it should receive a large y-value; and if p is pointed to by many pages with large y-values, then it should receive a large x-value. This motivates the definition of two operations on the weights, which we denote by I and O. Given weights { $x^{<p>}$}, { $y^{<p>}$}, the I operation updates the x-weights as follows.

$$x^{<p>} \leftarrow \sum_{q:\,(q,p) \in E} y^{<p>}$$

The O operation updates the y-weights as follows.

$$y^{<p>} \leftarrow \sum_{q:\,(q,p) \in E} x^{<p>}$$

Thus I and O are the basic means by which hubs and authorities reinforce one another. (Figure 12 [59, pg. 10]).

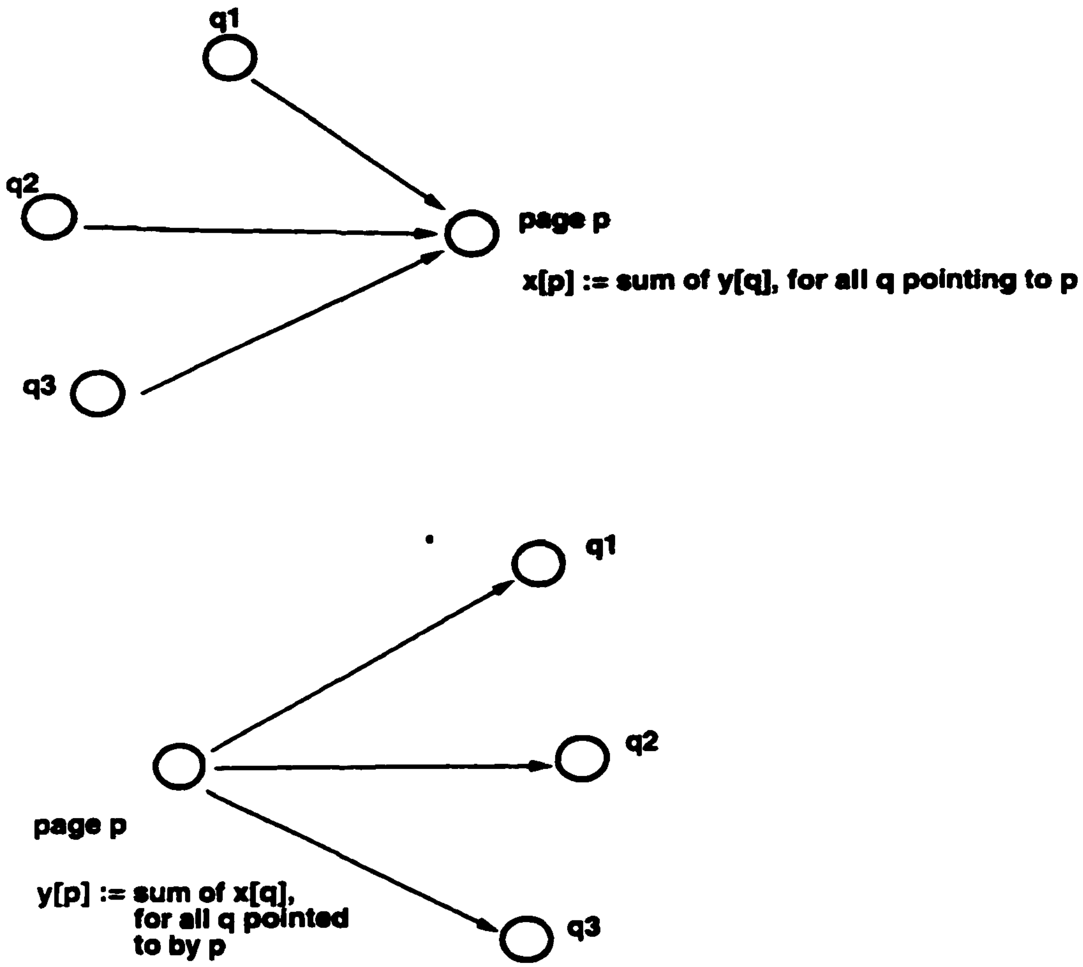


Figure 12

The basic operations

Now, to find the desired "equilibrium" values for the weights, Kleinberg [59] applies the I and O operations in an alternating fashion, and see whether a fixed point is reached. The set of weights represents as { $x^{<p>}$} as a vector x with a coordinate for each page in G_σ; analogously, the set of weights represents as{ $y^{<p>}$}as a vector y.

```
Iterate(G, k)
G: a collection of m linked pages
k: a natural number
Let z denote the vector $(1,1,1,..., 1) \in R^n$.
Set $x_0 := z$.
Set $y_0 := z$.
For i = 1,2,..., k
    Apply the I operation to $(x_{I-1}, y_{I-1})$, obtaining new x-weights $x^{'i}$.
    Apply the O operation to $(x_{I,} y_{I-1})$ obtaining new y-weights $y'^{i}$
    Normalize $x_i'$ obtaining $x_i$
    Normalize $y_i'$ obtaining $y_i$
End
Return $(x_k, y_k)$
```

This procedure can be applied to filter out the top c authorities and top C hubs in the following simple way.

```
Filter (G, k, c)
G: a collection of m linked pages
k, c:  natural numbers
    $(x_k, y_k)$ := Iterate(G, k).
Report the pages with the c largest coordinates in $x_k$ as authorities.
Report the pages with the c largest coordinates in $y_k$ as hubs.
```

Algorithm applies the Filter procedure with G set equal to G_σ, and typically with c≈5-10. To address the issue of how best to choose k, the number of iterations, system first show that as one applies Iterate with arbitrarily large values of k, the sequences of vectors {Xk} and {Yk} converge to fixed points X* and Y*.

Let M be a symmetric n x n matrix. An eigenvalve of M is a number λ with the property that, for some vector w, we have Mw = λw. The set of all such w is a subspace of R^n, which Kleinberg [59] refers to as the eigenspace associated with λ; the dimension of this space will be referred to as the multiplicity of λ. It is a standard fact that M has at most n distinct eigenvalues, each of them a real number, and the sum of their multiplicities is exactly n. These eigenvalues can be denoted by λ1(M), λ2(M),... , $\lambda_n(M)$, indexed in order of decreasing absolute value, and with each eigenvalue listed a number of times equal to its multiplicity. For each distinct eigenvalue, system chooses an orthonormal basis of its eigenspace; considering the vectors in all these bases, system can obtain a set of eigenvectors w1(M),w2(M),... , $w_n(M)$ that can index in such a way that $w_i(M)$ belongs to the eigenspace of $\lambda_i(M)$.

When this assumption holds, Kleinberg [59] refers to w1(M) as the "principal eigenvector", and all other $w_i(M)$ as "non-principal eigenvectors". When the assumption does not hold, the analysis becomes less clean, but it is not effected in any substantial way.

7. CONCLUSION

This master thesis introduces a visual analysis system for finding authoritative pages relevant to broad search topics on the World Wide Web based on a structural analysis of the link topology.

Basic components of this system are:

- The amount of related information is growing rapidly, making it difficult for users to filter the available resources, especially for broad search topics on the WWW. To be able to deal with this problem, one needs better data mining techniques to distill a broad search topic, possibly with millions of relevant web pages, down to a size that make sense to the users. For this reason, VAS defines a notion of "authoritative" [59] sources, based on the underlying link structure of the WWW.
- VAS locates authoritative pages that are of high quality as possible in the context of what is available on the WWW. Therefore, VAS underlying structure is not limited to a certain set of pages.
- VAS's main goal is to discover authoritative pages, but it also identifies a more complex pattern of social organization [60] on the WWW, in which hub pages link to set of related authorities.

- Finally, VAS is user-centric, allowing users to interact with the information stored in large sets of documents. Visualization is an integral part of the overall process. VAS focuses primarily on visualization and visual interfaces in support of information retrieval and data mining, shortly visual analysis.

Inferring Web Communities from Link Topology [60] describes a notion of hyperlinked communities on the World Wide Web through an analysis of the link topology. The paper describes, how the communities can be viewed as containing a central core, authoritative pages linked together by hub pages; and how they exhibit a natural type of hierarchical topic generalization that can be inferred directly from the pattern of linkage using 15 keywords or topics.

VAS used 5 of these keywords, these are:

- Harvard
- Geometry
- Cryptography
- English Literature
- Michael Jordan

Note that these selected topics have different levels of connection to the broad area of computer science, ranging from topics that are heavily computer oriented such as cryptography, to entirely different academic disciplines such as English Literature.

Research shows that communities on sufficiently broad topics tend to have a fairly robust structure. The groupings of pages discovered tend to be relatively independent of the exact choice of root set. The greatest degree of orderly structure is found in communities for which the number of relevant pages, and the density of hyperlinking, is the largest, especially with cryptography and English literature topic.

Finally, even with growing size of the World Wide Web, exploring the link structure of the Web is one of the promising solutions to the solve the issues presented in this thesis. This type of analysis that VAS has performed has to be repeated because of the dynamic growth of the WWW and results should be compared. Such an analysis provides a way of studying the communities [60] on the Web and of understanding how the techniques used in this project will modify as the Web continues to grow in both size and complexity.

8. LIMITATIONS

The evaluation of the methods presented with this thesis is a challenging task. Because VAS is attempting to define and compute, "authority", that is naturally based on human determination [59]. Also, the underlying structure of the WWW adds complexity to the problem of evaluation; it is a new domain, with a shortage of standard benchmarks.

Visual Analysis System calculates a closely linked collection of pages without looking at their semantic contents. Therefore, these pages are related to each other based on their calculations about hubs and authorities that VAS performs for given broad search topic. When the initial query string specifies a topic that is not necessarily broad, however, there will usually not be enough relevant pages in sets to form authorities. As a result, system gives a basic way of abstracting a specific query topic to a broader, related one. Although this restricts the ability of system to locate most authoritative pages for specific query topic, this diffusion can be important analysis in its own right. Consider for example "geometry". AltaVista [90] indexed about 200 pages containing the keyword; however, the resulting base set contained more general mathematics related topics, and main authorities were in fact very general mathematics resources.

9. FUTURE WORK

Linking and browsing defined the way in which people navigate on the Web. They are also very important and closely coupled with underlying structure of the Web that the way it has evolved. Browsing and searching can be further improved by recognition of Web communities [60]. Some search engines are beginning to use link structure of the Web, but effective analysis tools can be built into browsers too. For example, simply displaying pages pointing to the page being browsed can guide a user to good hub pages very quickly and it can be effective aids for filtering pages. In general, visual analysis tools that combine high-level information about the WWW link topology can guide users to select more link-based browsing techniques, and can help in developing techniques to navigation that make more effective use of global structure information.

In the future, four important issues need to be addressed. First, even with the high interactivity, people still find it difficult to find information relevant to their information needs. Therefore, in the dynamic structure of the Web, which techniques will allow retrieval of higher quality? Second, quick response is becoming more and more a important factor due to quickly increasing needs for access. Therefore, which techniques or algorithms will result of faster indexes and responses? Third, the quality of the retrieval part of the system is primarily affected by the user interaction with the system.

Therefore, how will a better understanding of the user behavior effect the design and development of new information retrieval systems? Four, the user of the Web usually does not know how to formulate his/her requests nor has great difficulty to do it. Therefore, developing an advanced user interfaces are very important and how will the ranking engine effected by the user interface?

The main application in the system is to collect the relevant pages on the Web into a single, large document collection. Parallel and distributed techniques can be used on these large documents to speed up this analysis. This is the approach currently used by the most search engines. But, how will parallel and distributed information retrieval can apply to VAS? Alternatively, we can exploit the distributed system of computers that make up the Web and spread the work of collecting, organizing, and searching all of the documents. This is the approach taken by the Harvest [66] system. Harvest comprises a number of components for gathering, summarizing, replicating, distributing, and searching documents. User queries are processed by brokers, which collect and refine information from gatherers and other brokers. The information at a particular broker is typically related to a restricted set of topics, allowing users to direct their queries to the most appropriate brokers for their queries

Many challenges remain in the area of parallel and distributed information retrieval. One of the challenges is measuring retrieval effectiveness on large set of collections. Although we can easily measure the speedup achieved by a given parallel system, the quality of the results produced by the system needs to be measured also [63].

One of the most important changes to VAS would be using 3D visualization techniques to display results to the user. This way users would have better understanding of the information space they are in, and using zoom in and out techniques, they can navigate this 3D space as a browsing option. The use of 3-dimensional representation through which users can navigate provides a much richer visualization to the user. This is because of the increased information density [21].

A supporting direction to VAS would be combining the semantic analysis of text with hyperlinks structure of the WWW. Based on the content analysis of the text, we can weight certain links as more authoritative than others, in the referring of Web location. Specifically, we can apply this semantic analysis technique to the pages in the root set for analysis of contents of the pages. For example, if the query string appears frequently and close to the link, system increases the corresponding weight for that page.

Today's World Wide Web is very different from that of just five years ago and predicting that what it will be in five years appears to be unimaginable. Maybe, indexing the Web will be infeasible in five years. And if so, will this analysis of searching the Web go through primary changes? Now, we can say that the increasing growth of the information on the Web will continue to generate computational challenges for researchers.

REFERENCES

[1] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, J. Kleinberg, Mining the Web's Link Structure, in *IEEE Computer, Volume 32, Issue 8, Pages: 60-68, August 1999.*

[2] J. Carriere, R. Kazman, WebQuery: Searching and Visualizing the Web through Connectivity, in *The Sixth International World Wide Web Conference Proceedings, Pages: 701-711, April 1997.*

[3] I. Greenberg, L. Garber, Searching for New Search Technologies, in *IEEE Computer, Volume 32, Issue 8, Pages: 4-8, August 1999.*

[4] N. Gershon, J. Le Vasseur, J. Winstead, J. Croall, A. Pernick, and W. Ruh, Case Study: Visualizing Internet Resources, in *IEEE Proceedings of Information Visualization, Pages: 122-128, October 1995.*

[5] G. Furnas, T. Landauer, L. Gomez, S. Dumais, The Vocabulary Problem in Human-Systems Communication, in *Communications of the ACM, Volume 30, Issue 11, Pages: 964-971, November 1987.*

[6] R. Kazman, J. Carriere, Rapid Prototyping of Information Visualizations using VANISH, in *Proceedings, IEEE Symposium on Information Visualization, Pages: 21-28, 1996.*

[7] R. Fielding, Maintaining Distributed Hypertext Infostructures: Welcome to MOMspider's Web, in *Proceedings of the First World Wide Web Conference, May 1994, www8.org/past-conf.html.*

[8] S. Mukherjea, J. Foley, Visualizing the World Wide Web with the Navigational View Builder, in *Proceedings of the Third World Wide Web Conference, April 1995, www8.org/past-conf.html.*

[9] T. Bray, Measuring the Web, in *Proceedings of the Fifth World Wide Web Conference, May 1996, www8.org/past-conf.html.*

[10] S.Chakrabarti, B. Dom, P. Indyk, Enhanced Hypertext Classification Using Hyper-Links, *in ACM SIGMOND International Conference, Management of Data, Pages: 307-318, 1998.*

[11] S. Brin, L. Page, The Anatomy of Large Scale Hypertextual Web Search Engine, in *Proceedings of the Seventh World Wide Web Conference, 1998, www8.org/past-conf.html.*

[12] R. M. Rohrer, J. L. Sibert, D. S. Ebert, A Shape-Based Visual Interface for Text Retrieval, in *IEEE Computer Graphics and Applications, Volume 19, Issue 5, Pages: 40-49 September/October 1999.*

[13] D. Ebert, *Advanced Geometric Modeling*, The Computer Science and Engineering Handbook, 1997.

[14] D. Ebert, Procedural Shape Generation for Multidimensional Data Visualization, in *Proceedings of Data Visualization, May 1999.*

[15] R. Rohrer, D. Ebert, J.Sibert, The Shape of Shakespeare: Visualizing Text using Implicit Surfaces, in *IEEE Proceedings on Information Visualization, Pages: 121-129, 1998.*

[16] Smart Information Retrieval System, Cornell University, N.Y. *ftp://ftp.cs.cornell.edu/pub/smart/.*

[17] S. Mukherjea, K. Hirata, Y. Hara, Visualizing the results of Multimedia Web Search Engines, in *IEEE Proceedings of Information Visualization, Pages: 64-65, October 1996.*

[18] P. C. Wong, Visual Data Mining, in *IEEE Computer Graphics and Applications, Volume 19, Issue 5, Pages: 20-22, September/October 1999.*

[19] G. Robertson, S. Card, J. Mackinlay, Information visualization using 3D interactive animation, in *Communications of ACM Volume 36, Issue 4, Pages: 56-71, April 1993.*

[20] T. Finin, R. Fritzon, D. McKay, An Overview of KQML: A Knowledge Query and Manipulation language, *Technical Report, Department of Computer Science, University of Maryland, 1992.*

[21] R. J. Hendley, N. S. Drew, A. M. Wood, R. Beale, Narcissus: Visualizing Information, in *IEEE Proceedings of Information Visualization, Pages: 90-96, October 1995.*

[22] J. Vion-Dury, M. Suntan, Virtual images: Interactive visualization of distributed object-oriented systems, in *Proceedings of Object Oriented Programming Systems Languages and Applications, Pages: 65-84, 1994.*

[23] K. Andrews, Visualizing Cyberspace: Information Visualization in the Harmony Internet Browser, in *IEEE Proceedings of Information Visualization, Pages: 97-104, October 1995.*

[24] G. G. Robertson, S. K. Card, J. D. Mackinlay, The Information Visualiser, An Information Workplace, in *ACM Proceedings: Human factors in Computing Systems, Pages: 181-188, CHI 1991.*

[25] G. G. Robertson, S. K. Card, J. D. Mackinlay, The Perspective Wall: Detail and Context Smoothly Integrated, in *ACM Proceedings: Human factors in Computing Systems, Pages: 173-179, CHI 1991.*

[26] G. G. Robertson, S. K. Card, J. D. Mackinlay, Cone Trees: Animated 3D Visualizations of Hierarchical Information, in *ACM Proceedings: Human factors in Computing Systems, Pages: 189-202, CHI 1991.*

[27] K. M. Fairchild, L. Serra, N. Hern, L. B. Hai, A. T. Leong, Dynamic Fish Eye Information Visualizations, in *Virtual Reality Systems, 1993.*

[28] K. M. Fairchild, S. E. Poltrock, G.W. Furnas, SemNet: Three Dimensional representations of Large Knowledge Bases, in *Applications for Human Computer Interaction, 1988.*

[29] L. Serra, T.S. Chua, W S. Teh, A Model for Integrating Multimedia Information Around 3D Graphics Hierarchies, in *The Visual Computer, 1991.*

[30] P. A. Smith, J. R. Wilson, Navigating in Hypertext through Virtual Environments, in *Applied Ergonomics, 1993.*

[31] J. Tesler, S. Strasnick, FSN: The 3D File System Navigator, *Silicon Graphics, 1992.*

[32] K. Andrews, F. Kappe, H. Maurer, Hyper-G: Towards the Next Generation of Network Information Technology, in *Journal of Universal Computer Science, April 1995.*

[33] K. Andrews, F. Kappe, H. Maurer, Serving Information to the Web with Hyper-G, in *Computer Networks and ISDN systems, April 1995.*

[34] N. Gershon, Moving Happily Through the World Wide Web, in *IEEE Computer Graphics and Applications, Pages: 72-77, March 1996.*

[35] R. M. Rohrer, E. Swing, Web based Information Visualization, in *IEEE Computer Graphics and Applications, Volume 17, Issue 4, Pages: 52-60, July/August 1997.*

[36] K.A. Olsen, R.R. Korfhage, Information Display: Control of Visual Representations, in *Proceedings, Visual Languages, Pages: 56-61, 1991.*

[37] S. Mukherjea, K. Hirata, Y. Hara, Towards a Multimedia World Wide Web Information Retrieval Engine, in *The Sixth International World Wide Web Conference, Pages: 177-188, April 1997.*

[38] M. Marchiori, The Quest for Correct Information on the Web: Hyper Search Engines, in *The Sixth International World Wide Web Conference, Pages: 265-276, April 1997.*

[39] C. Chang, C. Hsu, Customizable Multi – Engine Search Tool with Clustering, in *The Sixth International World Wide Web Conference, Pages: 257-264, April 1997.*

[40] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Oltson, V. Raman, T. Roth, P. J. Haas, Interactive Data Analysis: The Control Project, in *IEEE Computer, Volume 32, Issue 8, Pages: 51-59, August 1999.*

[41] B. Shneiderman, Dynamic Queries for Visual Information Seeking, *Research Paper, Department of Computer Science, University of Maryland, HCI Laboratory, January 1994.*

[42] B. A. LaMacchia, The Internet Fish Construction Kit, in *The Sixth International World Wide Web Conference, Pages: 277-288, April 1997.*

[43] E. Frecon, G. Smith, WEBPATH – A Three Dimensional Web History, in *IEEE Proceedings of Information Visualization, Pages: 3-10, October 1998.*

[44] D. Snowdon, S. Benford, C. Greenhalgh, R. Ingram, C. Brown, L. Fahlen, M. Stenius, A 3D Collaborative Virtual Environment for Web Browsing, in *Virtual Reality WorldWide'97, April 1997.*

[45] S. Card, G. Robertson, W. York, The WebBook and the Web Forager: An Information Workspace for the World Wide Web, in *Proceedings of Human Factors in Computing Systems, Pages: 111-117, CHI 1996.*

[46] P. Doemel, WebMap – A Graphical Hypertext Navigation Tool, in *Second International Conference on the WWW, 1994, www8.org/past-conf.html.*

[47] B. Bederson, J. Hollan, J. Stewart, D. Vick, L. Ring, E. Grose, C. Forsythe, A Zooming Web Browser, in *Human Factors in Web Development, 1998.*

[48] U. Wiss, D. Carr, H. Johnson, Evaluating Three - Dimensional Information Visualization Designs: A case study of Three Designs, in *IEEE Proceedings of Information Visualization, 1998.*

[49] C. S. Jeong, A. Pang, Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space, in *IEEE Proceedings of Information Visualization, Pages: 19-25, October 1998.*

[50] X. Meng, Z. Chen, R. H. Fowler, R. K. Fox, W. A. Lawrence-Fowler, Data Visualization, Indexing and Mining Engine – A Parallel Computing Architecture for Information Processing Over the Internet, *Research Paper, Department of Computer Science, The University of Texas – Pan American, February 1998.*

[51] R.H. Fowler, W.A.L. Fowler, J. L. Williams, 3D Visualization of the WWW Semantic Content for Browsing and Query Formulation, in *WebNet'96, World Conference of the Web Society Proceedings, Pages: 147-152, October 1996.*

[52] R. H. Fowler, B. A. Wilson, W. A. L. Fowler, Information Navigator: An information system using associative networks for display and retrieval, *Research Paper, Department of Computer Science, The University of Texas – Pan American, 1992.*

[53] R. H. Fowler, A. Kumar, A spring modeling algorithm to position nodes of an undirected graph in three dimensions, *Technical Report, Department of Computer Science, The University of Texas – Pan American, 1994.*

[54] T. Kamada, S. Kawai, An Algorithm for drawing general undirected graphs, in *Information Processing Letters, Volume 31, Pages: 7-15, 1988.*

[55] R. Botafogo, E.Rivlin, B. Schneiderman, Structural analysis of hypertext: Identifying hierarchies and useful metrics, in *ACM Trans. Inf. Sys. Volume 10, Issue 2, Pages: 115-141, April 1992.*

[56] M. E. Frisse, Searching for information in a hypertext medical handbook, in *Communications of the ACM, Volume 31, Issue 7, Pages: 880-886, July 1988.*

[57] H. Small, B. C. Griffith, *The structure of the scientific literatures I. Identifying and graphing specialties*, Science Studies.

[58] J. Pitkow, P. Pirolli, Life, death, and lawfulness on the electronic frontier, in *Proceedings of ACM CHI, Conference on Human Factors in Computing Systems: Common Ground, Pages: 383-390, 1997.*

[59] Jon. M. Kleinberg, Authoritative Sources in a Hyperlinked Environment, in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, Pages: 668-677, 1998.*

[60] D. Gibson, J. Kleinberg, P. Raghavan, Inferring Web Communities from Link Topology, in *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, Pages: 225-234, June 1998.*

[61] I. Ben-Shaul, M. Herscovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, V. Soroka, S. Ur, Adding support for Dynamic and Focused Search with Fetuccino, *www8.org/w8-papers/5a-search-query/adding/adding.html.*

[62] K. Cios, W. Pedrycz, R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers, 1998.

[63] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, 1999.

[64] E. Selberg, O. Etzioni, Multi-Service Search and Comparison Using the MetaCrawler, in *Proceedings of the 1995 World Wide Web Conference, www8.org/past-conf.html.*

[65] C. Lynch, *Searching the Internet*, Scientific American, March 1997.

[66] C. M. Bowman, P. B. Danzig, D. R. Hardy, Harvest: A Scalable, Customizable Discovery and Access System, *Technical Report, Department of Computer Science, University of Colorado, 1995.*

[67] J. E. McEneaney, Visualizing and Assessing Navigation in Hypertext, *Technical Report, Division of Education, Indiana University South Bend, 1999.*

[68] S. Chakrabarti, M. van den Berg, B. Dom, Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery, *www8.org/w8-papers/5a-search-query/crawling/index.html.*

[69] D. Gibson, J. Kleinberg, P. Raghavan, Clustering Categorical Data: An Approach Based on Dynamical Systems, in *Proceedings of the 24th International Conference on Very Large Databases, 1998.*

[70] O. Zamir, O. Etzioni, Web Document Clustering: A Feasibility Demonstration, *Technical Report, Department of Computer Science and Engineering, University of Washington.*

[71] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, The Web as a graph: measurements, models, and methods, *Technical Report, Department of Computer Science, Cornell University and IBM Almaden Research Center.*

[72] D. Mladenic, Text-Learning and Related Intelligent Agents: A Survey, in *IEEE Intelligent Systems, Pages: 44-55, July-August 1999.*

[73] W.W. Cohen and W. Fan, Learning Page-Independent Heuristics for Extracting Data from Web Pages, *www8.org/w8-papers/5a-search-query/learning.html.*

[74] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork, Measuring Index Quality Using Random Walks on the Web, *www8.org/w8-papers/2c-search-discover/measuring/measuring.html.*

[75] R. C. Miller, K. Bharat, SPHINX: a framework for creating personal, site-specific Web crawlers, *Technical Report, School for Computer Science, Carnegie Mellon University and Digital Systems Research Center.*

[76] J. Cho, H. Garcia-Molina, and L. Page, Efficient crawling through URL ordering, *Technical Report, Department of Computer Science, Stanford University.*

[77] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg, Automatic resource compilation by analyzing hyperlink structure and associated text, *Technical Report, IBM Almaden Research Center.*

[78] S. Mukherjea, J. D. Foley, Showing the Context of Nodes in the World Wide Web, Graphics, Visualization & Usability Center, *Technical Report, College of Computing, Georgia Institute of Technology.*

[79] S. Mukherjea, J. D. Foley, and S. Hudson, Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views, *Technical Report, Graphics, Visualization & Usability Center, College of Computing, Georgia Institute of Technology.*

[80] R. Weiss, B. Velez, M. A. Sheldon, C. Namprempre, P. Szilagyi, A. Duda, D. K. Grifford, HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering, *Technical Report, Programming Systems Research Group, MIT Laboratory for Computer Science.*

[81] S. F. Roth, M. C. Chuah, S. Karpedjiev, and J. A. Kolojejchick, Toward an Information Visualization Workspace: Combining Multiple Means of Expression, *in Human-Computer Interaction, 1997.*

[82] S. Lawrence and C. L. Giles, Accessibility of information on the web, *NEC Research Institute, http://www.neci.nj.nec.com.*

[83] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization, Using Vision to Think*, Morgan Kaufmann Publishers, Inc, 1999.

[84] B. Shneiderman, *Designing the User Interface, Strategies for Effective Human Computer Interaction*, Third Edition, 1998 by Addison Wesley Longman, Inc.

[85] C.Petzold, *Programming Windows 95, The Definitive Developer's Guide to the Windows 95 API*, 4th Edition, 1996 by Microsoft Press.

[86] J. Prosise, *Programming Windows with MFC*, Second Edition, 1999 by Microsoft Press.

URL REFERENCES

[87] Fetuccino light service: http://www.inb.com/java/fetuccino

[88] Google: http://www.google.com

[89] Mapuccino: http://www.ibm.com/java/mapuccino

[90] AltaVista: http://www.altavista.com

[91] HotBot: http://www.hotbot.com

[92] Yahoo!: http://www.yahoo.com

[93] Excite: http://www.excite.com

[94] Northern Light: http://www.northernlight.com

[95] MetaCrawler: http://www.metacrawler.com

[96] SavySearch: http://www.savysearch.com

[97] Netsizer: http://www.netsizer.com

[98] Netcraft: http://www.netcraft.com

[99] NEC Research Institute: http://www.neci.nj.nec.com

[100] Search Engine Watch: http://searchenginewatch.com/reports/

[101] Graphic Visualization and Usability Center: http://gvu.gatech.edu

[102] Web Growth Summary: http://www.mit.edu/people/mkgray/net/web-growth-summary.html

[103] LookSmart: http://www.looksmart.com

[104] Go To: http://goto.com

[105] MSN: http://www.msn.com

[106] Magellan: http://www.mckinley.com

[107] MFC: http://msdn.microsoft.com/library/devprods/vs6/visualc/vcmfc/_mfc_Class_Library_Referencc_Introduction.htm

[108] Visual C++: http://msdn.microsoft.com/isapi/msdnlib.idc?theURL=/library/devprods/vs6/visualc/vcedit/vcstartpage.htm

[109] Windows NT: http://www.microsoft.com/ntserver

[110] Microsoft: http://www.microsoft.com

[111] W3C – Web Characterization Workshop: http://www.w3.org/1998/11/05/wc-workshop/papers/kleinber1.html

VITA

Tarkan Karadayi

University of Texas – Pan American, Master of Science, Computer Science, May 2000

University of Ege, Bachelor of Science, Agriculture Engineering, May 1992

Permanent Address: 1806 S. Petunia Ave., Weslaco, TX 78596