ELSEVIER

Decision Support Systems 34 (2002) 1–17

Decision Support Systems

www.elsevier.com/locate/dsw

CI Spider: a tool for competitive intelligence on the Web

Hsinchun Chen *, Michael Chau, Daniel Zeng

Department of Management Information Systems, Eller College of Business and Public Administration, University of Arizona, Tucson, AZ 85721, USA

Accepted 1 December 2001

Abstract

Competitive Intelligence (CI) aims to monitor a firm's external environment for information relevant to its decision-making process. As an excellent information source, the Internet provides significant opportunities for CI professionals as well as the problem of information overload. Internet search engines have been widely used to facilitate information search on the Internet. However, many problems hinder their effective use in CI research. In this paper, we introduce the Competitive Intelligence Spider, or CI Spider, designed to address some of the problems associated with using Internet search engines in the context of competitive intelligence. CI Spider performs real-time collection of Web pages from sites specified by the user and applies indexing and categorization analysis on the documents collected, thus providing the user with an up-to-date, comprehensive view of the Web sites of user interest. In this paper, we report on the design of the CI Spider system and on a user study of CI Spider, which compares CI Spider with two other alternative focused information gathering methods: Lycos search constrained by Internet domain, and manual within-site browsing and searching. Our study indicates that CI Spider has better precision and recall rate than Lycos. CI Spider also outperforms both Lycos and within-site browsing and searching with respect to ease of use. We conclude that there exists strong evidence in support of the potentially significant value of applying the CI Spider approach in CI applications.

Keywords: Competitive intelligence; Internet searching and browsing; Internet spider; Noun phrasing; Document clustering; Experimental research

1. Introduction

The goal of Competitive Intelligence (CI), a subarea of Knowledge Management, is to monitor a firm's external environment to obtain information relevant to its decision-making process [7]. Many major companies, such as Ernst & Young and General Motors, have formal and well-organized CI units that enable managers to make informed decisions about critical business matters such as investment, marketing, and strategic planning. Traditionally, CI relied upon published company reports and other kinds of printed information. In recent years, Internet has rapidly become an extremely good source of information about the competitive environment of companies and has been reported by a Futures Group survey in 1997 to be one of the top five sources for CI professionals [6].

* Corresponding author. Department of Management Information Systems, University of Arizona, McClelland Hall 430Z, Tucson, AZ 85721, USA. Tel.: +1-520-621-4153.

E-mail addresses: hchen@bpa.arizona.edu (H. Chen), mchau@bpa.arizona.edu (M. Chau), zeng@bpa.arizona.edu (D. Zeng).

0167-9236/02/$ - see front matter
PII: S0167-9236(02)00002-7

Although the Internet represents significant CI opportunities, it has also brought about many technical, cognitive, and organizational challenges. Because the amount of information available on the Internet is overwhelming, CI professionals are constantly facing the problem of information overload. It is estimated that there are over 1 billion Web pages on the Internet as of February 2000 [8]. Much time and effort is required for CI professionals to search for the relevant information on the Internet and then analyze the information collected in the correct context.

Internet search engines have been useful in helping people search for information on the Internet. Nevertheless, the exponential growth of information sources on the Internet and the largely unregulated and dynamic nature of many Web sites are making it increasingly difficult to locate useful information using these search engines. It has been estimated that none of the search engines available indexes more than 16% of the total Web that could be indexed [12]. This has resulted in a low recall rate when the user is looking for obscure or unusual material. In addition, since Web page contents are extremely dynamic and may change daily or even every minute, conventional preindexed search engines suffer from the problem of providing many outdated and obsolete links.

In this paper, we present a novel approach implemented as Competitive Intelligence Spider (CI Spider) that can be used to alleviate some of the problems associated with the usual search engine approach. CI Spider accepts as input the URLs the user specifies, and follows the embedded Web links to search for user-specified keywords. After collecting on the fly a certain number (user-definable) of Web pages, CI Spider performs further text analysis to extract noun phrases from these pages. These noun phrases represent a list of key topics covered on the Web sites of interests. CI Spider also provides the functionality of visualizing the retrieved Web pages in a 2-D map where Web pages sharing similar topics are grouped together in regions. The main research hypothesis examined in this paper is that an integrated approach, such as CI Spider, can better facilitate CI professionals to analyze and summarize relevant Web sites than existing approaches using Internet search engines.

The rest of the paper is organized as follows. Section 2 briefly reviews the basic concept of Competitive Intelligence (CI) and discusses various technological supports available for CI professionals, including Internet search engine technology and related information management issues. In Section 3, we present the architectural design of the CI Spider system and give detailed technical information for the major components of CI Spider. Section 4 focuses on an evaluation methodology designed to evaluate our research hypothesis concerning the effectiveness and efficiency of an integrated approach for CI tasks. In Section 5, we report a user study performed to test our hypothesis and discuss the strength and weakness of the CI Spider system. Section 6 concludes the paper with a summary and a discussion about future research directions.

2. Literature review

2.1. Competitive intelligence

The Society of Competitive Intelligence Professionals defines competitive intelligence as "the process of ethically collecting, analyzing and disseminating accurate, relevant, specific, timely, foresighted and actionable intelligence regarding the implications of the business environment, competitors and the organization itself" [22]. CI is different from espionage, which implies illegal means of information gathering; CI is restrained to the gathering of *public* information. Indeed, another definition of CI is "the use of public sources to develop information about the competition, competitors, and market environment" [15].

One of the main differences between CI and general business information, such as business growth rate and transaction figures, is that CI is of strategic importance on the organization. It is not only the collection of information from a variety of sources, but also the analysis and synthesis of such information, which could help the company decide the course of action to improve its position [23].

A typical CI process consists of a series of business activities that involve identifying, gathering, developing, analyzing and disseminating information [7,10,25,26]. The following list shows a typical sequence in which these activities take place.

(1) Identify competitors, markets, customers, suppliers, or other variables in the environment to be

monitored. Identify what information is to be collected.
(2) Specifically identify possible sources of information and collect the information from these sources.
(3) Evaluate the validity, reliability, and usefulness of the information collected.
(4) Gather information collected from different sources and integrate them.
(5) Interpret and analyze the information for strategic or tactical significance. Draw conclusions and recommend actions.
(6) Disseminate and present analyzed findings to management.
(7) Respond to ad hoc inquiries for decision support.

Information gathering and information analysis are the key areas of the CI process. Our proposed CI Spider tool supports CI professionals to perform activities (1) to (5) of the above CI process on the Internet.

2.2. Competitive intelligence and the Internet

Commercial online databases, such as Dialog (http://www.dialog.com) and Lexis-Nexis (http://www.lexisnexis.com), contain a large amount of well-organized information on a variety of subjects, storing information ranging from company annual reports to US patent laws, and from history journals to chemistry periodicals. Most of the documents are provided in plain text format. Traditionally, these commercial databases are among the major sources used by CI professionals.

Recent years have seen the tremendous growth of the Internet. Many commercial online databases are now accessible through the Internet. The Internet also enables organizations to monitor and search the Web sites of their competitors, alliances, and possible collaborators. Internet-based information sources are becoming increasingly important in the CI process. Corporate Web sites usually contain a variety of useful information, including company history, corporate overviews, business visions, product overviews, financial data, sales figures, annual reports, press releases, biographies of top executives, locations of offices, hiring ads, etc. [1,5,16]. These data are valuable in providing direct or indirect contextual information to enable the CI professionals to analyze corporate strategies.

Another reason attracting CI professionals to use the Internet is that most of the contents available on the Internet are available free of charge. In fact, Internet is now one of the most important resources for CI information collection. According to a Future Groups survey taken in 1997, 82% of the respondents said that the Internet was a primary source of information, while only 70% agreed that commercial databases were their primary sources [6].

2.3. Internet-based CI tools

2.3.1. Challenges of using Web-based sources for CI

A survey of over 300 CI professionals shows that data collection is the most time-consuming task in typical CI projects, accounting for more than 30% of the total time spent [19]. Managers tend to think that more information is better. In today's business environment, however, it is not necessarily true. CI professionals could be spending too much time and effort on data collection rather than data analysis. This information overload problem is particularly pertinent to the Internet-based CI. Compounding the problem further is that many Web pages are updated weekly, daily or even hourly. For CI professionals to manually access the Internet, read the information on every single Web page at a company Web site to locate the useful information, and to synthesize information is mentally exhausting and overwhelming. To address this information and cognitive overload problem, research has been conducted in developing techniques and tools to analyze, categorize, and visualize large collections of Web pages, among other text documents. In turn, a variety of tools have been developed to assist searching, gathering, monitoring and analyzing information on the Internet. A prominent example is Web search engines.

2.3.2. Web search engines

Many different search engines are available on the Internet. Each has its own characteristics and employs its preferred algorithm in indexing, ranking and visualizing Web documents. For example, AltaVista (http://www.altavista.com) and Infoseek (http://www.infoseek.com) allow users to submit queries and present the Web pages in a ranked order, while

Yahoo! (http://www.yahoo.com) groups Web sites into categories, creating a hierarchical directory of a subset of the Internet. There are also special-purpose domain-specific search engines, such as BuildingOnline (http://www.buildlingonline.com), which specializes in searching in the building industry domain on the Web, and LawCrawler (http://www.lawcrawler.com), which searches for legal information on the Internet.

A Web search engine usually consists of the following components:

1. Spiders (a.k.a. crawlers): to collect Web pages from the Web using different graph search algorithms.
2. Indexer: to index Web pages and store the indices into database.
3. Retrieval and Ranking: to retrieve search results from the database and present ranked results to users.
4. User Interface: to allow users to query the database and customize their searches.

The problem with this approach is that given the size of the Web, it takes a long time to spider and index all the relevant Web pages, even for domain-specific search engines. Many Web pages may be spidered, but not indexed, resulting in outdated or incorrect information.

Another type of search engines is the *meta-search* engines, such as MetaCrawler (http://www.metacrawler.com) and Dogpile (http://www.dogpile.com). These search engines do not keep their own indexes. When a search request is received, a meta-search engine connects to multiple popular search engines and integrates the results returned by these search engines. As each search engine covers different portion of the Internet, meta-search engines are useful when the user needs to get as much of the Internet as possible.

Given the growing popularity of peer-to-peer (P2P) technology, distributed search systems also have been proposed. Similar to meta-search engines, InfraSearch (http://www.infrasearch.com), using Gnutella as the backbone, does not keep its own indexes. Each participating computer runs a piece of software to links itself to a few other computers. When a request is received from a user, the request is passed to neighbor computers to see if any computer can fulfill the request. Each computer can have its own strategy on how to respond to the request. As a result, timely and dynamic information can be returned to the user because the search is no longer dependent on indexes of typical search engines. However, one drawback of this approach is that each Web site has full control on how to respond to each search request. As a result, a company may, for example, be able to hide its information from particular competitors.

In addition to the above commercial Web search engines that can be accessed through an Internet browser, there is another type of search engines that reside on the user machine. Because the software is running on the client machine, more CPU time and memory can be allocated to the search process and more functionalities can be possible. In recent years, more powerful client-side spiders have been developed. For example, Blue Squirrel's WebSeeker (http://www.bluesquirrel.com) and Copernic 2000 (http://www.copernic.com) connect with different search engines, monitor Web pages for any changes, and schedule automatic search. Focused Crawler [2] locates Web pages relevant to a predefined set of topics based on example pages provided by the user. In addition, it also analyzes the link structures among the Web pages collected.

2.3.3. Monitoring and filtering

Because of the fast changing nature of the Internet, different tools have been developed to monitor Web sites for changes and filter out unwanted information. *Push Technology* is one of the emerging technologies in this area. The user first needs to specify some areas of interest. Then the tool will automatically *push* related information to the user. Ewatch (http://www.ewatch.com) is one such example. It monitors information not only from Web pages, but also from Internet Usenet groups, electronic mailing lists, discussion areas and bulletin boards to look for changes and alert the user.

Another popular technique used for monitoring and filtering employs a software agent, or intelligent agent [14]. Personalized agent can monitor Web sites and filter information according to particular user needs. Machine learning algorithms, such as an artificial neural network, are usually implemented as agents to learn the user' preferences.

2.3.4. Text analysis and visualization

There have been many studies on textual information analysis from the information retrieval and natural language processing literature. In order to retrieve documents based on given concepts, the source documents have to be indexed. Since the mental effort and time requirements for manual indexing are prohibitively high, automatic indexing algorithms have been used to extract key concepts from textual data. It has been shown that automatic indexing can be as effective as human indexing [21]. Many proven techniques have been developed. One of such techniques, the Arizona Noun Phraser, performs indexing for phrases rather than just keywords [24]. Such techniques are useful in extracting meaningful phrases from text documents not only for document retrieval, but also for further follow-up analysis. Natural language processing has also been applied in analyzing user search queries. For example, instead of performing keyword-based searches, AskJeeves (http://www.ask.com) accepts search query posted in the form of a question, such as "Where can I find profiles of companies in Arizona?" Such questions may be better answered by search engines with natural language processing techniques.

In Web document clustering, there are, in general, two ways to define the categories. In the first approach, the categories are predefined manually based on library science classification methods. NorthernLight (http://www.northernlight.com), a commercial search engine, is an example of this approach. Although the categorization algorithm is not disclosed, our experience with the system indicates that when a user submits a search query to NorthernLight, the results of the search are classified into optional predefined categories. In the second approach, documents are classified "on the fly" without predefined categories. Category labels will be defined based on the keywords that appear in the documents collected. This approach usually relies on some kind of machine learning algorithms. For example, the self-organizing map (SOM) approach classifies documents into different categories, which are automatically determined during the classification process, using neural network algorithms [11].

After the Web documents are analyzed, the results have to be displayed to the user in an organized and meaningful way. A graphical representation can facilitate the elicitation of competitive intelligence knowledge to CI professionals and management [9]. Various visualization tools based on different metaphors have been developed. There are two main types of document visualization. The first type is the visualization of document attributes and aims to provide the user with more information about the documents. Most techniques in this category present the user with a list of available documents with a short summary for each document. The second type of visualization is based on interdocument similarities. These techniques aim at reducing the multidimensional document space to a 2-D or 3-D space by aggregating similar documents under the same topic. They provide users with a quick overview of the whole collection such that the users do not have to manually click into each link and read the document content. Grouper [29] presents the categories as a list ranked by the coherence within each category. The SOM technique assigns documents into different regions of a 2-D map based on the similarity of the documents. Regions that are similar to each other are located close to each other [11,13, 20]. Applications using this technique are reported to make it more efficient and satisfying for users to browse the document collection.

2.3.5. CI systems

In order to address the needs of CI professionals in strategic decision making, a lot of commercial CI systems have been developed. For example, Excalibur RetrievalWare and Internet Spider (http://www.excalib.com) collect, monitor and index information from text documents on the Web as well as graphic files. They, however, do not automatically categorize documents into different groups. Autonomy's products (http://www.autonomy.com) support a wide range of information collection and analysis tasks, which includes automatic searching and monitoring information sources in the Internet and corporate Intranets, and categorizing documents into categories predefined by users or domain experts. Verity's knowledge management products (http://www.verity.com), such as Agent Server, Information Server and Intelligent Classifier, also perform similar tasks in an integrated manner. CI Spider, the system developed in this research, searches for relevant Web pages based on keywords and other criteria specified by the user. The documents are indexed and clustered into different

groups, where categories do not need to be predefined.

2.4. System evaluation methodologies

Studies in the area of information retrieval and analysis have been numerous. Different evaluation methodologies have been used by different researches. The traditional evaluation method for searching tools relies on precision rate, recall rate, and the time required to search for particular information. While these methods are adequate for measuring retrieval effectiveness, they do not capture how well an experiment participant understands the content of all the documents retrieved.

Document categorization systems are evaluated differently. Besides precision and recall, a categorization system should also be evaluated on the basis of its usefulness as a browsing tool [4]. Other measures, such as *size of clusters*, *term relevance* and *term association*, have also been used [17,18]. To the best of our knowledge, no previous studies have attempted to evaluate a combination of Web document retrieval with document categorization in an effort to evaluate the impact of both technologies.

Another set of evaluations is possible through qualitative feedback. Qualitative data are mostly collected through recording users' "think aloud" protocols and thorough questionnaires. During typical experiments, subjects are encouraged to express their likes and dislikes about the system explicitly as well as to give reasons behind their navigation choices. They are also asked to complete questionnaires regarding the experiment. These comments are usually recorded by the experimenters and are later analyzed using protocol analysis. Qualitative data constitute an important aspect of our evaluation reported in Section 5.

3. CI Spider system architecture

3.1. Architectural design

In this section, we present a detailed technical description of a novel CI approach implemented as the CI Spider system. The architecture of CI Spider system is shown in Fig. 1. CI Spider has four main components, i.e., User Interface, Internet Spiders, Noun Phraser, and Self-Organizing Map (SOM).

Because CI professionals need timely and updated information, CI Spider uses a real-time search strategy, which is different from typical commercial search engines. Instead of keeping a database of index that may be a few weeks old, CI Spider collects and indexes Web pages only when requested by the user. Despite the fact that the current implementation of the proposed system does not handle dynamic content, this search strategy ensures that all static Web information returned to the user is up to the minute. In the

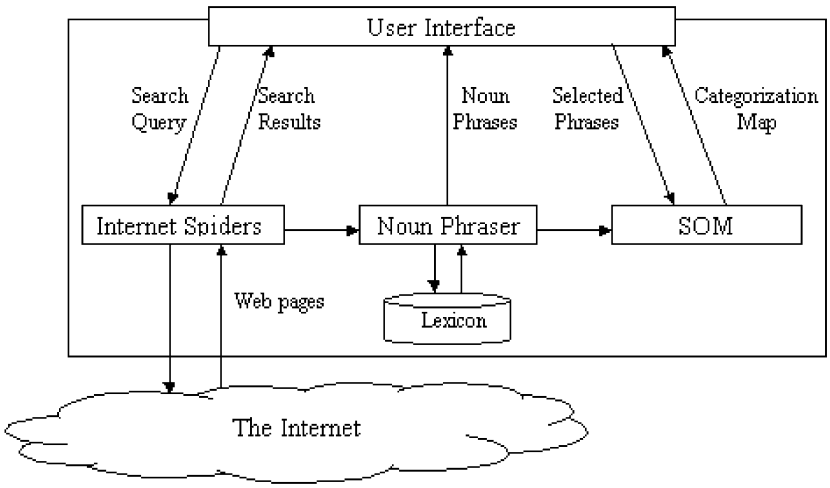

Fig. 1. CI Spider architecture.

following sections, we describe each of these components in details.

3.2. User Interface

The User Interface component of CI Spider allows the user to enter the starting URLs where he or she wants to begin a search, and the keywords being sought (Fig. 2). The user can also specify other search options, such as the number of pages to be searched, the number of Internet Spiders to be deployed, Boolean operators for search terms, and the search constraints (e.g., limiting the search to certain domains such as .edu). The search request is then sent to the Internet Spiders.

3.3. Internet Spiders

The Internet Spiders are simple Java spiders employing a Breadth-First Search algorithm that starts from the URLs specified by the user and follows all the possible links to search for the given keywords, until the number of Web pages collected reaches the user-specified target. The spiders run in multithread such that the fetching process will not be affected 2by slow server response time. Robots Exclusion Protocol is also implemented such that the spiders will not access sites where the Web master has placed a text file named robots.txt in the host or a special meta-tag in a Web page, indicating the unwillingness to serve Internet spiders for a particular subset of Web pages.

Whenever a page is collected during the search, the link to that page is displayed dynamically on one of the result screens (Fig. 3). The left frame in Fig. 3 shows a hierarchical structure of the Web pages visited, which shows how the Web pages collected are linked to each other on the Web site. When the user clicks on a link on the left, the link and all the links contained in that page will be displayed in the right frame. The user can then click on any page displayed and read its full content without having to wait for the whole search to be completed. The user can also switch to the Good URL List to browse the pages that contain the search keyword (Fig. 4). When the number of Web pages collected meets the target, the spiders stop and the results are sent to the Noun Phraser for further analysis.

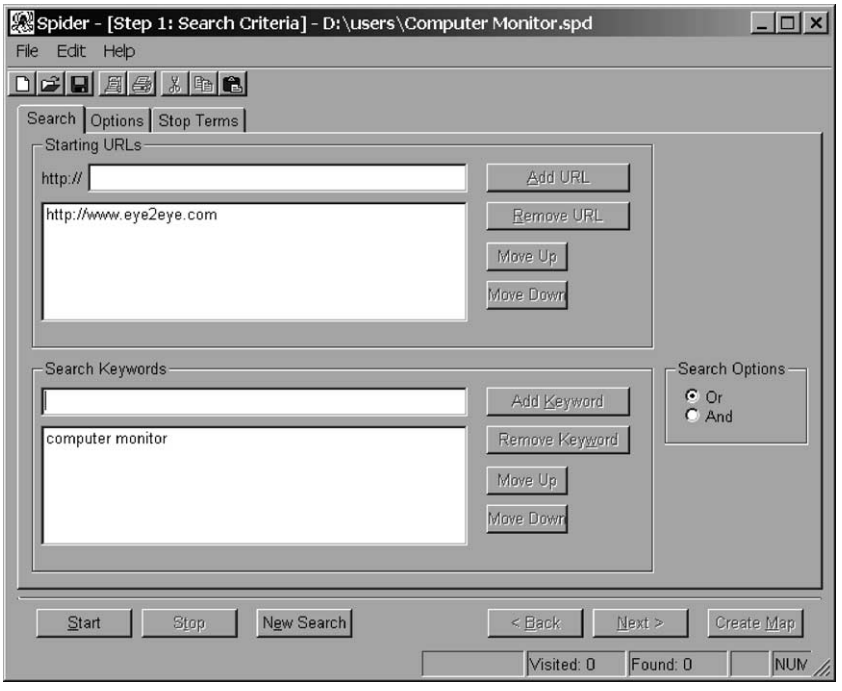

Fig. 2. First screen of CI Spider where users can enter search query.

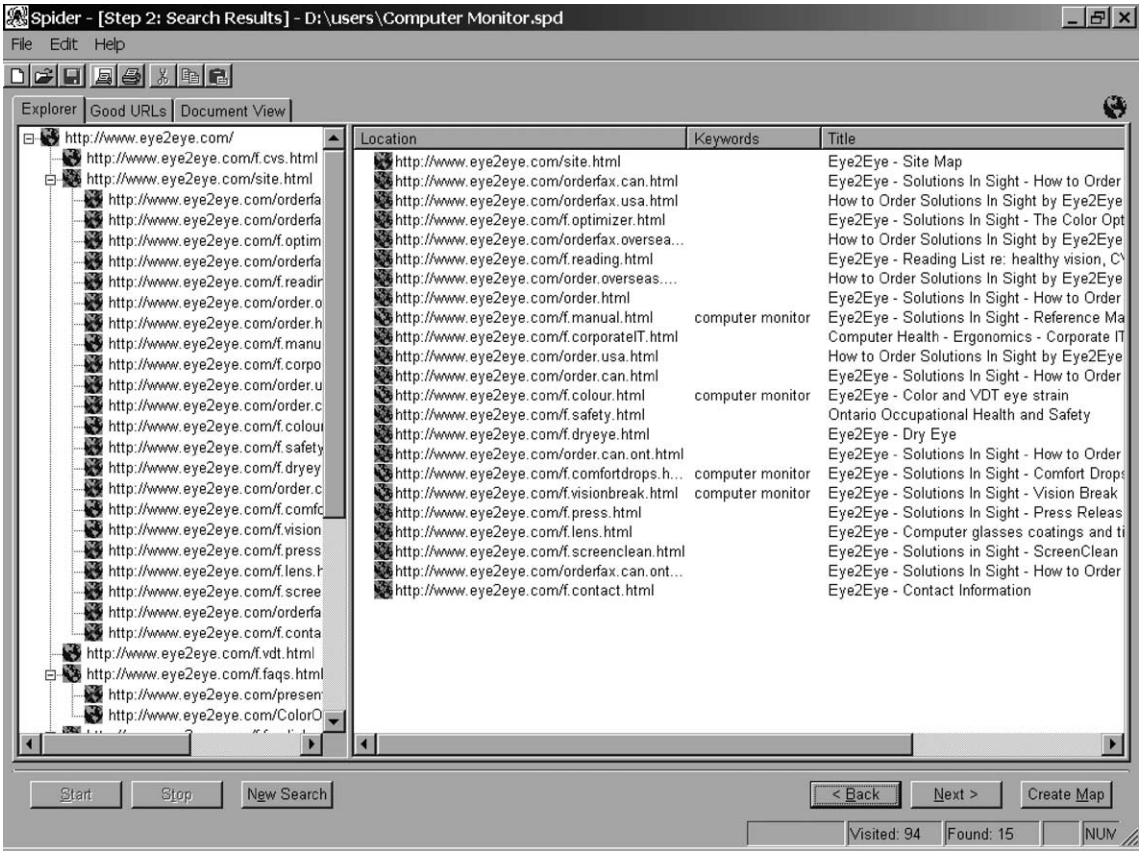

Fig. 3. Hierarchical display of Web pages collected.

3.4. Analysis and visualization

The indexing tool integrated in CI Spider is called the Arizona Noun Phraser (AZNP). Developed at the University of Arizona, AZNP extracts and indexes all the noun phrases from each document collected by the Internet Spiders based on part-of-speech tagging and linguistic rules [24]. AZNP has three components. The *tokenizer* takes Web pages as text input and creates output that conforms to the UPenn Treebank word tokenization rules by separating all punctuation and symbols from text without interfering with textual content. The *tagger* module assigns part-of-speech to every word in the document. The last module, called the *phrase generation* module, converts the words and associated part-of-speech tags into noun phrases by matching tag patterns to noun phrase pattern given by linguistic rules. For example, the phrase *new business transformation* will be considered a valid noun phrase because it matches the rule that an adjective–noun–noun pattern forms a noun phrase. The frequency of every phrase is recorded and sent to the User Interface. The user can view the document frequency of each phrase and link to the documents containing that phrase (Fig. 5). After all documents are indexed, the data are aggregated and sent to the Self-Organizing Map (SOM) for automatic categorization.

CI Spider uses an approach based on the Kohonen SOM to give users an overview of the set of documents collected [11]. The Kohonen SOM employs an artificial neural network algorithm to automatically cluster the Web pages collected into different regions on a 2-D map (Fig. 6). In SOM, each document is represented as an input vector of keywords and a 2-D grid of output nodes are created. The distance between the input and each output node is then computed and the node with

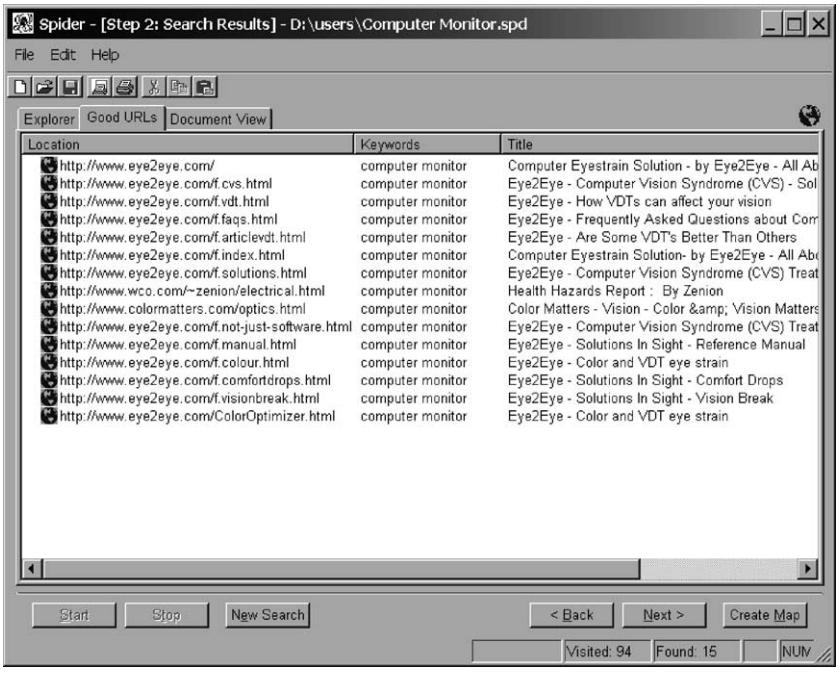

Fig. 4. List of URLs that contain the search keywords.

the minimum distance is selected. After the network is trained through repeated presentation of all inputs, the documents are submitted to the trained network and each region is labeled by the phrase which is the key concept that most represents the cluster of documents in that region. More important concepts occupy larger regions (e.g., *symptoms*), and similar concepts are grouped in a neighborhood [13]. The map is displayed through the User Interface and the user can view the documents in each region by clicking on it. The Dynamic SOM (DSOM) technique is used in CI Spider such that the user can select and deselect phrases for inclusion in the analysis and produce a new map on the fly within seconds.

4. Evaluation methodology

4.1. Experimental tasks

To evaluate the effectiveness and efficiency of different methods in performing both document retrieval and document categorization tasks in the CI process, we performed a comparative user study to contrast CI Spider with two other traditional CI approaches. For this user study, traditional evaluation methodologies do not apply since they treat document retrieval and document categorization separately. In our study, we designed the experimental tasks in such a way that we could measure and evaluate the performance of the combination of the systems' retrieval and categorization functionalities. Our evaluation involved asking the test subjects to identify the major themes related to a certain topic at a particular Web site. More specifically, each subject was first instructed to locate the pages containing the given topic accessible through the given Web site using the different search methods described in the next section. The subject was then required to comprehend the contents of all the Web pages relevant to that keyword, and to summarize the findings as a number of themes. The tasks were designed according to those open-ended "soft" queries evaluated in the Text Retrieval Conferences (TREC). The TREC series is sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA) to encourage research in information retrieval from large text collections. TREC strives to

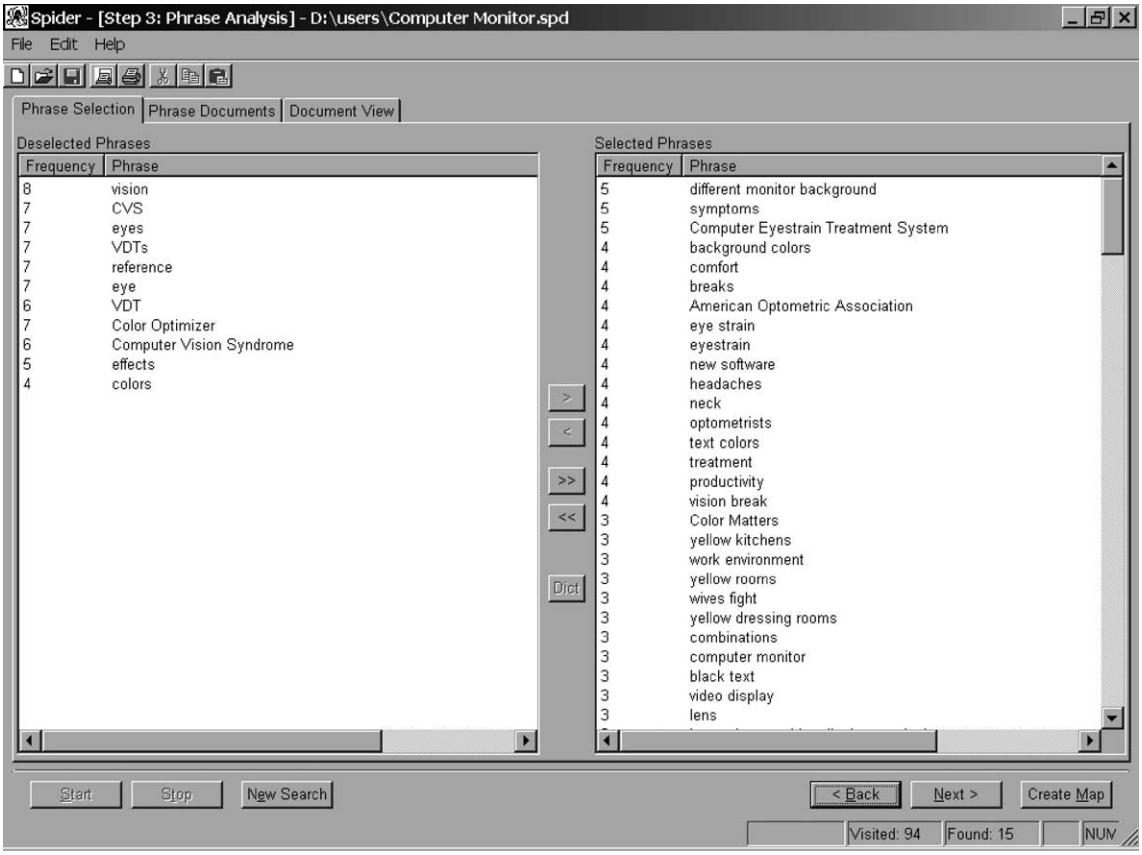

Fig. 5. List of noun phrases extracted from the Web pages collected.

provide a common task evaluation that allows cross-system comparisons [27], which is consistent with our user study.

In our experiment, a theme was defined as "a short phrase which describes a certain topic". Phrases like *success of the 9840 tape drive in the market* and *business transformation services* are examples of themes in our experiment. By examining the themes that the subjects came up with using different search methods, we were able to evaluate how effectively and efficiently each method helped a user locate a collection of documents and gain a general understanding of the response to a given search query on a certain Web site.

During the experiment, each subject was first given the URL of a Web site and a topic to investigate. A sample session is shown in Fig. 7. The subject launched the search in CI Spider by typing the URL and topic in the query box. In the example, the subject searched for the phrase "computer monitor" in the Web site "http://www.eye2eye.com". After clicking the "Start" button, CI Spider started fetching pages from the Web and performed verification and noun–phrase indexing. In the Good URL List, the subject could browse the Web pages collected while CI Spider was still searching for more pages. After CI Spider had collected the specified number of Web pages, the noun phrase indexes were aggregated and presented to the subject, ranked by the frequency of the noun phrase. The subject could click on any of these phrases and see the relevant Web pages. The subject could also select and phrases he or she liked and deselect the others to produce a 2-D map (SOM), which provided an overview of the Web pages collected. If the subject did not like that map, he or she could generate a new map by choosing a new set of phrases. Finally, the subject could summarize the findings based on all these information.

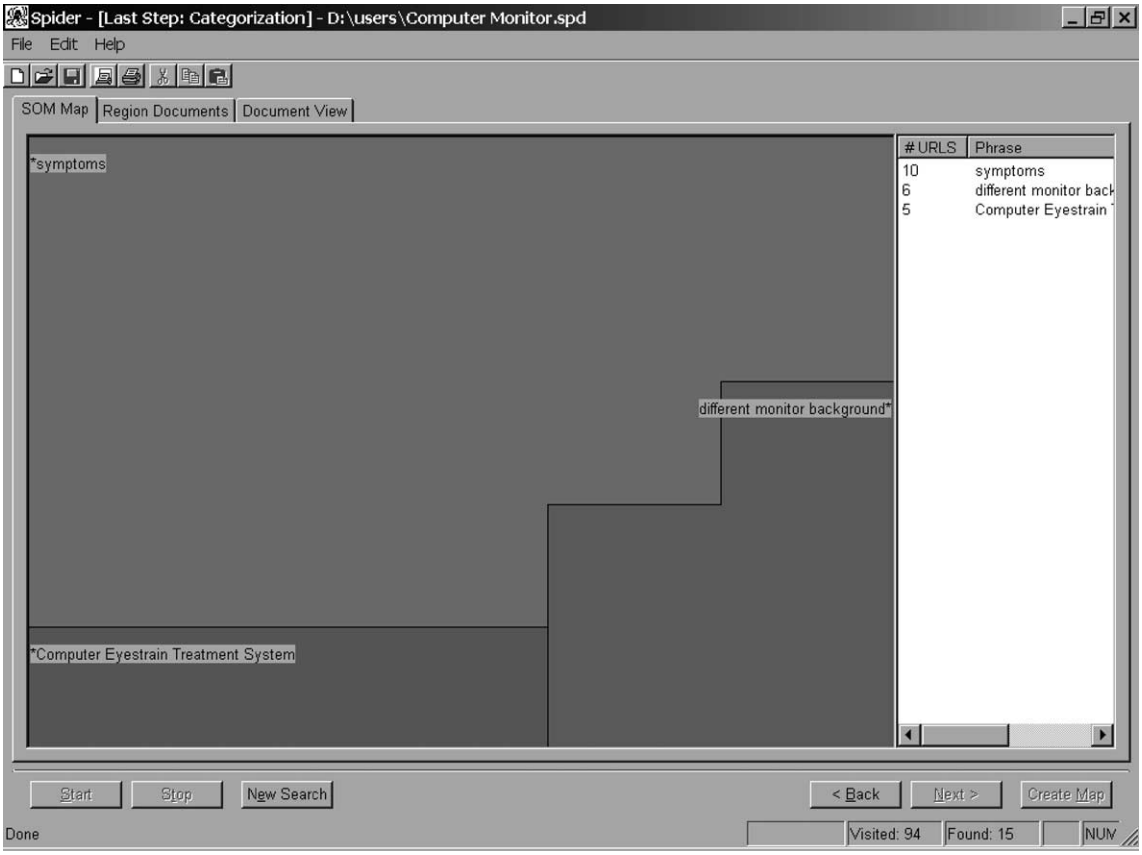

Fig. 6. Documents clustered into different regions in SOM.

4.2. Experiment design and hypotheses

In our experiment, CI Spider was compared against two popular approaches that CI professionals use to search for competitive information on the Internet. CI professionals often use popular commercial search engines to collect data on the Internet, or they simply explore the Internet manually. We compared CI Spider with both approaches. The first approach evaluated was using Lycos to collect competitive information on the Web. We chose Lycos because of its popularity and also because that it allows the user to search for a certain keyword in a given Web domain. Our subjects were instructed to take benefits of this feature during the experiment. The second method was manual "within-site" browsing and searching, corresponding to the situation where the subject freely explores the contents in the given Web site using an Internet browser. The following hypotheses were tested in our experiment:

Hypothesis 1. CI Spider achieves a higher precision and recall than Lycos for searching within a domain.

Hypothesis 2. CI Spider achieves a higher precision and recall than within-site browsing/searching for searching within a domain.

Hypothesis 3. It is easier to search within a domain using CI Spider than using Lycos.

Hypothesis 4. It is easier to search within a domain using CI Spider than using within-site browsing/searching.

Hypothesis 5. CI Spider requires less time than within-site browsing/searching for searching within a domain.

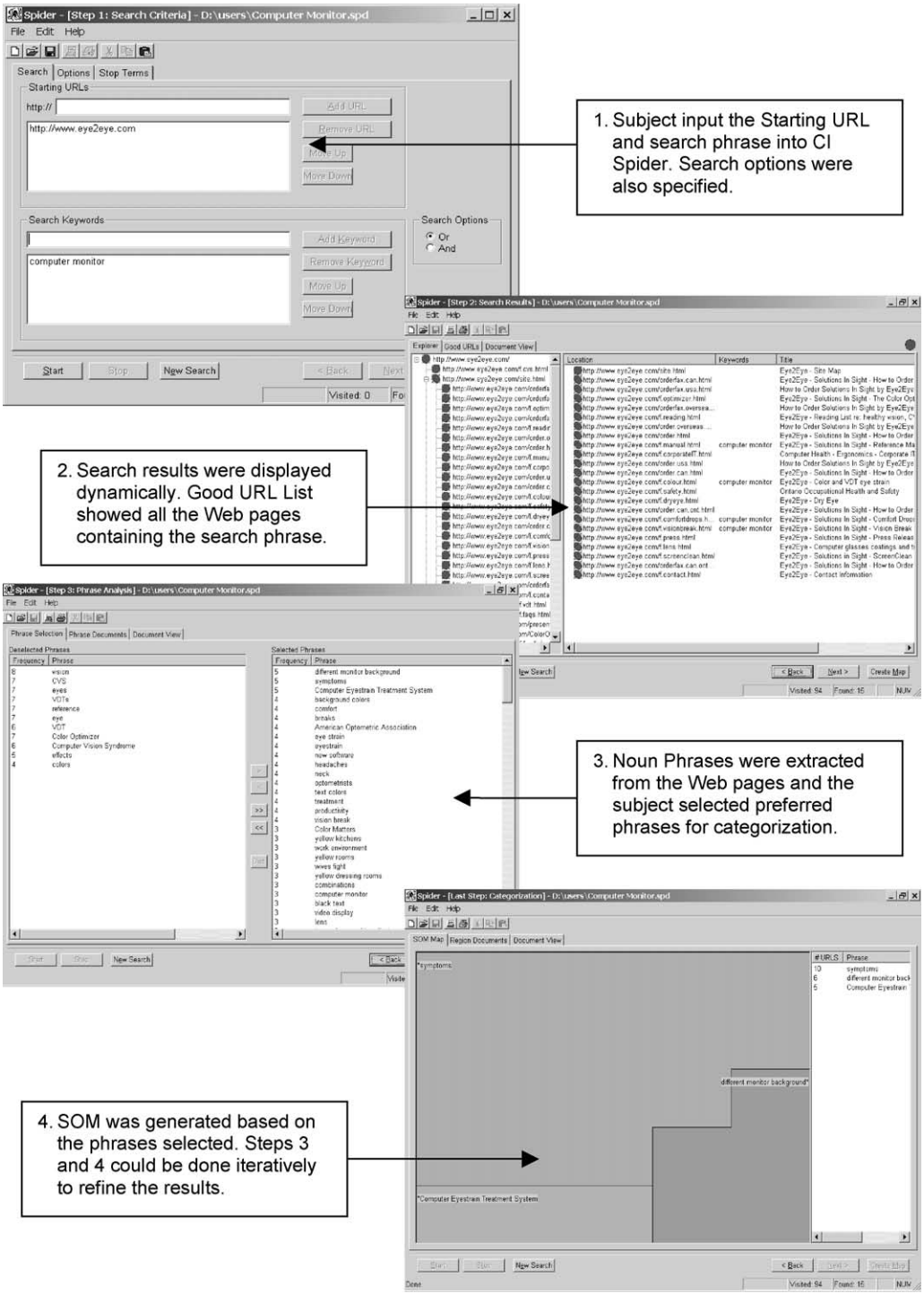

Fig. 7. Example of a user session.

Six search queries were designed for the experiment based on suggestions given by CI professionals we consulted. For example, one of our search tasks was to locate and summarize the information related to "merger" on the Web site of a company called Phoenix Technologies (http://www.phoenix.com). Two pilot studies were conducted in order for us to refine the search tasks and experiment design. During the real experiment, thirty subjects, mostly juniors from the Department of Management Information Systems from our home institution, were recruited and each subject was required to perform three out of the six different searches using the three different search methods. Rotation was applied such that the order of search methods and search tasks tested would not bias our results. Web sites with different sizes, ranging from small sites, such as http://www.eye2eye.com to large sites such as http://www.ibm.com, were chosen for the experiments.

4.3. Performance measurement

Two graduate students majoring in library science were recruited as expert judges for this experiment. They manually went through all relevant pages in the given Web sites and individually summarized their findings into themes. Their results were then aggregated to form the basis for evaluation. Precision and recall rates for the number of themes were used to measure the effectiveness of each search method.

The time spent for each experiment, including the system response time and the user browsing time, was recorded to evaluate the efficiency of the three search methods. During the experiment, we encouraged our subjects to tell us about the search method used and their comments were recorded. Finally, each subject filled out a questionnaire to give further comments about the three different methods.

5. Experiment results and discussion

5.1. Experiment results

The quantitative results of our user study are summarized in Table 1. Four main variables for each subject have been computed for comparison: precision, recall, time, and ease of use. Precision and recall rates are calculated as follows:

Table 1
Experiment results

	CI Spider	Lycos	Within-site browsing/searching
Precision			
Mean	0.708	0.477	0.576
Variance	0.120	0.197	0.150
Recall			
Mean	0.273	0.163	0.239
Variance	0.027	0.026	0.033
Time (min)			
Mean	10.02	9.23	8.60
Variance	11.86	44.82	36.94
Ease of Use			
Mean	3.97	3.33	3.23
Variance	1.34	1.13	1.29

$$\text{precision} = \frac{\text{number of correct themes identified by the subject}}{\text{number of all themes identified by the subject}}, \quad (1)$$

$$\text{recall} = \frac{\text{number of correct themes identified by the subject}}{\text{number of correct themes identified by expert judges}}. \quad (2)$$

The time recorded was the total duration of the search task, including both response time of the system and the browsing time of the subject. Ease of use was calculated based on subjects' responses to the question "How easy/difficult is it to locate useful information using [that search method]?" Subjects were required to choose a level from a scale of 1 to 5, with 1 being the most difficult and 5 being the easiest.

We performed various *t*-tests to examine whether the differences between these approaches are statistically significant. The results are summarized in Table 2. We conclude that both the precision and recall rates for CI Spider are significantly higher than those of Lycos at a 5% significant level. CI Spider also earns a statistically higher value in ease of use than Lycos and within-site browsing and searching.

5.2. Strengths and weaknesses of CI Spider

5.2.1. Precision and recall

The *t*-test results show that CI Spider performed statistically better in both precision and recall than Lycos-based search, confirming Hypothesis 1. CI Spider also performed better than within-site browsing

Table 2
t-Test comparisons (*p*-values)

	CI Spider vs. Lycos	CI Spider vs. within-site browsing/ searching	Lycos vs. within-site browsing/ searching
Precision	0.029	0.169	0.365
Recall	0.012	0.459	0.087
Time	0.563	0.255	0.688
Ease of Use	0.031	0.016	0.726

and searching in precision and recall, but the differences are not statistically significant (Hypothesis 2 unconfirmed). In terms of precision, we suggest that the main reason for the superior performance of CI Spider is its ability to fetch and verify the content of each Web page in real time. CI Spider ensures that every page shown to the user contains the keyword being searched. On the other hand, we found that some indexes in Lycos were outdated. As a result, a number of URLs returned by Lycos were irrelevant or dead links, resulting in low precision. Subjects also reported that in some cases two or more URLs returned by Lycos pointed to the same page, which created confusion and annoyance. For within-site browsing and searching, we found that 85% of subjects utilized internal search engines when available. These internal search engines are usually comprehensive and up to date. In addition, many sites provide a site map or a site index feature, facilitating subjects to locate information. However, using these internal search engines or site index did not lead to better precision performance due to the lack of real-time indexing and verification.

The high recall rate of CI Spider is mainly attributable to the exhaustive searching nature of the spiders. Lycos has the lowest recall rate because, like most other commercial search engines, it samples only a number of Web pages in each Web site, thereby missing other pages that contain the keyword. For within-site browsing and searching, a user can easily miss some important pages due to human errors and cognitive overload. The strength of within-site browsing and searching over Lycos is that internal search engines index most pages in their Web sites, resulting in a more comprehensive index. In many cases, most relevant information already has been clustered under the same subtopic at a Web site, making it easy to locate.

5.2.2. Display and analysis of Web documents

In our study, subjects believed it was easier to find useful information using CI Spider (with a score of 3.97/5.00) than using Lycos domain search (3.33) or manual within-site browsing and searching (3.23), confirming Hypothesis 3 and Hypothesis 4. Three main reasons may account for this. First, CI Spider's superior precision and recall performance saved users considerable time and mental effort. Second, CI Spider's intuitive and useful interface design helped subjects locate useful information easily. Third, the analysis tools helped subjects form an overview and summarization of all the relevant Web pages collected. The Arizona Noun Phraser allowed subjects to narrow and refine their searches as well as provided a list of key phrases that represented the collection. The Self-Organizing Map generated an easy-to-read 2-D map display on which subjects could click to view the documents related to a particular theme of the collection. In order to assess the value of each individual component of CI Spider, subjects were required to choose the component(s) that they thought to be most helpful. Among them, 77% of the subjects thought the tree display of URLs and Good URL List useful; 37% voted for the list of noun phrases; 10% chose SOM, the map-display.

In general, many subjects liked the display of CI Spider. Subjects indicated that they liked its neat and intuitive interface design. They also highly valued the availability of the Good URL List, which showed all the URLs containing the search keyword. Most subjects used this list as a search result list similar to that of commercial search engines, except for its being filtered and verified. They also commented that the Good URL List allowed them to find themes quickly without the need to browse through many Web pages.

The Arizona Noun Phraser in CI Spider helps users refine a search. When the user clicks on a noun phrase, a list of Web pages that contain both the search keyword and the chosen noun phrase will be displayed. A number of subjects used this as their primary way to understand and locate the topics for their search tasks. A significant number of subjects, however, did not use the list of the noun phrases since they had got all their findings from the previous step, i.e., the Good URL List.

The SOM component of CI Spider was not widely used by the subjects. One obvious reason is that most subjects collected what they needed from the Good

URL List and the Arizona Noun Phraser. They did not find it necessary to go to this final step of this application. Another reason could be attributed to the relatively small size of the documents collected. In general, the performance of SOM for document categorization is not very satisfactory when the number of documents is fewer than a hundred. For our tasks, the user seldom retrieved more than 30 relevant documents. Therefore, the SOM categories created during our experiments were not of high quality.

5.2.3. Speed

The *t*-test results demonstrated that the three search methods did not differ significantly in time requirements, leaving Hypothesis 5 unconfirmed. As discussed in the previous section, the time used for comparison is the total searching time and browsing time. Real-time indexing and fetching time, which usually took more than 3 minutes in our experiment, also was included in the total time for CI Spider. In other words, CI Spider can potentially save users' time and effort in the CI process because they only need to browse the verified and summarized results instead of manually going through the whole process. When fast response is needed, search engines like Lycos are more desirable, because they generate results within seconds. However, the user needs to spend more time browsing and verifying the documents.

6. Conclusion and future directions

In this paper, we describe a novel approach for CI applications. Our initial user study demonstrates the potential impact of using an integrated document retrieval and automatic categorization approach for CI tasks. We conclude that our approach has significantly higher precision and recall than Lycos-based searches in assisting the user to summarize Web pages related to a certain topic in a given Web domain. We also found that CI Spider has higher precision and recall than the usual within-site browsing and searching practice, but the result is not statistically significant.

Because CI Spider is capable of handling multiple starting URLs, we anticipate that CI Spider can perform better than within-site browsing and searching, since it is difficult for the user to manually integrate the results obtained from different sites. CI Spider also has received high scores for its user friendliness.

Another strength of the proposed approach is its ability to handle Web sites that have a large number of pages. We observed that our subjects usually spent a considerable amount of time in downloading Web pages when they used CI Spider for their search tasks. Because this step can be performed without user intervention, it may save the user much time in the searching process. CI Spider can collect, index and summarize thousands of pages within an hour. This is much desirable when precise result, rather than fast response, is needed.

We are currently engaged in research to improve CI Spider. To further enhance the system, we are working on providing more search functionalities and options to provide the user with finer control over the search. We also plan on improving the search algorithm. For example, it has been reported that the spiders can be made more effective by employing genetic algorithm, best-first search or hybrid simulated annealing instead of breadth-first search [3,28]. In addition, to let the user monitor the rapidly changing Web, we are working on developing time-tagging and intelligent caching mechanism such that changes can be detected and reported to the user.

Based on the positive experience from this study, we are currently applying CI Spider in specific domains, such as medicine and law enforcement, thereby allowing the use of a customized lexicon to suit the specific terminology of a domain. We are also looking into developing new CI Spider components to support multilingual text retrieval and analysis.

Acknowledgements

We would like to express our gratitude to the following agencies for supporting this project:

- NSF Digital Library Initiative-2, "High-performance Digital Library Systems: From Information Retrieval to Knowledge Management", IIS-9817473, April 1999–March 2002.
- NSF/CISE/CSS, "An Intelligent CSCW Workbench: Personalized Analysis and Visualization", IIS-9800696, June 1998–June 2001.

We would also like to thank all the members of the Artificial Intelligence Lab at the University of Arizona who have contributed to the implementation of the system, in particular, Wojciech Wyzga, Harry Li, Andy Clements and David Hendriawan.

References

[1] R.D. Aaron, Giving away the store? Tell people about your company on your Web site–but don't overdo it, Competitive Intelligence Review 8 (2) (1997) 80–82.
[2] S. Chakrabarti, M. van der Berg, B. Dom, Focused crawling: a new approach to topic-specific Web resource discovery, Proceedings of the 8th International World Wide Web Conference (Toronto, Canada, May 1999).
[3] H. Chen, Y. Chung, M. Ramsey, C.C. Yang, An intelligent Personal Spider (agent) for dynamic Internet/Intranet searching, Decision Support Systems 23 (1) (1998) 41–58.
[4] H. Chen, A. Houston, R. Sewell, B. Schatz, Internet browsing and searching: user evaluations of category map and concept space techniques, Journal of the American Society for Information Science, Special Issue on "AI Techniques for Emerging Information Systems Applications" 49 (7) (1998) 582–603.
[5] A. Dutka, Competitive Intelligence for the Competitive Edge, NTC Business Books, Chicago, IL, 1998.
[6] Futures Group, Ostriches & Eagles 1997, The Futures Group Articles, 1998.
[7] B. Gilad, T. Gilad, The Business Intelligence System, AMACOM, New York, 1988.
[8] Inktomi WebMap, available at http://www.inktomi.com/webmap/.
[9] R.J. Johnson, A Cognitive Approach to the Representation of Managerial Competitive Intelligence Knowledge, Doctoral dissertation, The University of Arizona, 1994.
[10] B.E. Keiser, Practical competitor intelligence, Planning Review 8 (1987) 14–18.
[11] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin, 1995.
[12] S. Lawrence, C.L. Giles, Accessibility of information on the Web, Nature 400 (1999) 107–109.
[13] C. Lin, H. Chen, J. Nunamaker, Verifying the proximity and size hypothesis for self-organizing maps, Journal of Management Information Systems 16 (3) (1999–2000) 61–73.
[14] P. Maes, Agents that reduce work and information overload, Communications of the ACM 37 (7) (July 1994) 31–40.
[15] J.J. McGonagle, C.M. Vella, Outsmarting the Competition, Sourcebooks, Naperville, IL, 1990.
[16] J.J. McGonagle, C.M. Vella, The Internet Age of Competitive Intelligence, Quorum Books, London, 1999.
[17] M. McQuaid, T. Ong, H. Chen, J. Nunamker, Multidimensional scaling for group memory visualization, Decision Support Systems 27 (1999) 163–176.
[18] R. Orwig, H. Chen, J. Nunamaker, A graphical, self-organizing approach to classifying electronic meeting output, Journal of the American Society for Information Science 48 (2) (1997) 157–170.
[19] J.E. Prescott, D.C. Smith, SCIP: who we are, what we do, Competitive Intelligence Review 2 (1) (1991) 3–5.
[20] D. Roussinov, H. Chen, Document clustering for electronic meetings: an experimental comparison of two techniques, Decision Support Systems 27 (1999) 67–69.
[21] G. Salton, Another look at automatic text-retrieval systems, Communications of the ACM 29 (7) (1986) 648–656.
[22] Society of Competitive Intelligence Professionals, http://www.scip.org/.
[23] H. Sutton, Competitive Intelligence, The Conference Board, New York, Report 913, 1988.
[24] K.M. Tolle, H. Chen, Comparing noun phrasing techniques for use with medical digital library tools, Journal of the American Society for Information Science 51 (4) (Apr. 2000) 352–370.
[25] K.W. Tyson, Business Intelligence: Putting It All Together, Leading Edge Publications, Lombard, IL, 1986.
[26] R.G. Vedder, M.T. Vanecek, C.S. Guynes, J.J. Cappel, CEO and CIO perspectives on competitive intelligence, Communications of the ACM 42 (8) (Aug. 1999) 109–116.
[27] E. Voorhees, D. Harman, Overview of the sixth text retrieval conference (TREC-6), in: E. Voorhees, D. Harman (Eds.), NIST Special Publication 500-240: The Sixth Text Retrieval Conference (TREC-6), National Institute of Standards and Technology, Gaithersburg, MD, USA, 1997.
[28] C.C. Yang, J. Yen, H. Chen, Intelligent Internet searching agent based on hybrid simulated annealing, Decision Support Systems 28 (2000) 269–277.
[29] O. Zamir, O. Etzioni, Grouper: a dynamic clustering interface to Web search Results, Proceedings of the 8th International World Wide Web Conference (Toronto, Canada, May 1999).

Hsinchun Chen is McClelland Professor of MIS and Andersen Professor of MIS at the University of Arizona, where he is the director of the Artificial Intelligence Lab and the director of the Hoffman E-Commerce Lab. His articles have appeared in Communications of the ACM, IEEE Computer, Journal of the American Society for Information Science and Technology, IEEE Expert, and many other publications. Professor Chen has received grant awards from NSF, DARPA, NASA, NIH, NIJ, NLM, NCSA, HP, SAP, 3COM, and AT&T. He serves on the editorial board of Decision Support Systems and the Journal of the American Society for Information Science and Technology and has served as the conference general chair in the International Conferences on Asian Digital Library in the past 4 years.

Michael C. Chau is a doctoral student in the Department of Management Information Systems at the University of Arizona, where he is also a research associate of the Artificial Intelligence Lab. His current research interests include information retrieval, natural language processing, Web mining, and multiagent systems. He received a BS in Computer Science (Information Systems) from the University of Hong Kong.

Daniel Dajun Zeng is an assistant professor in the Department of Management Information Systems at the University of Arizona. His research interests include software agents and their applications, distributed artificial intelligence, distributed decision support systems, negotiation, multiagent learning, supply chain management, and intelligent information gathering. He received MS and PhD degrees in industrial administration from Carnegie Mellon University, and a BS in economics and operations research from the University of Science and Technology of China, Hefei, China. He is a member of INFORMS and AAAI.