(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 June 2001 (14.06.2001)

PCT

(10) International Publication Number
WO 01/42986 A1

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/33340

(22) International Filing Date: 8 December 2000 (08.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/169,889 9 December 1999 (09.12.1999) US

(71) Applicant *(for all designated States except US)*: **RENSSELAER POLYTECHNIC INSTITUTE** [US/US]; 3210 J Building, Troy, NY 12180 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **SZYMANSKI, Boleslaw, K.** [—/—]; - (**).

(74) Agent: **GROSSMAN, Jon, D.**; Dickstein Shapiro Morin & Oshinsky LLP, 2101 L Street NW, Washington, DC 20037-1526 (US).

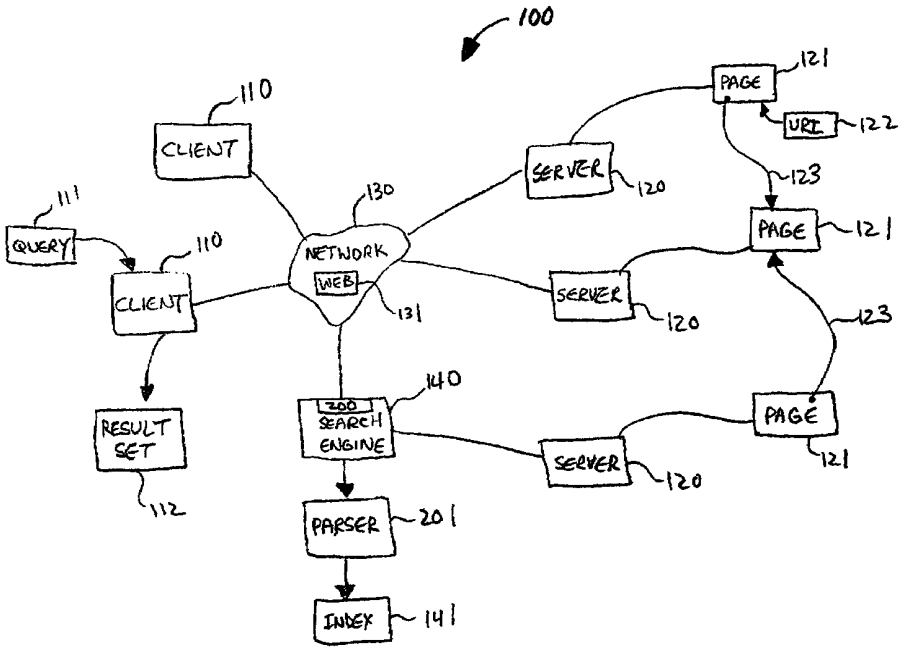
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR INDEXING DOCUMENTS USING CONNECTIVITY AND SELECTIVE CONTENT ANALYSIS



(57) Abstract: The present invention provides a method for indexing (141) documents. More particularly, a method in which the contents of the documents linked to the indexed (141) page (121) is used. In other words, the method uses a structure of words used by the linked pages (121). All the stored information is parsed (201) and references containing links to the document to be indexed are captured. The captured references are further parsed (201) to collect its content.

WO 01/42986 A1

METHOD AND SYSTEM FOR INDEXING DOCUMENTS USING CONNECTIVITY AND SELECTIVE CONTENT ANALYSIS

FIELD OF THE INVENTION

This invention relates generally to computerized information retrieval, and more particularly to ranking retrieved documents based on the content and the connectivity of the documents.

DISCUSSION OF THE RELATED ART

It has become common for end-users connected to the World Wide Web (the " Web") to employ Web browsers and search engines to locate Web pages having specific content of interest to end-users. A Web page is a document on the Web as compared to a Web site which is a location on the Web. Each Web site may contain several Web pages or documents. A search engine indexes hundreds of millions of Web pages maintained by computers all over the world. The end-users compose queries usually comprised of one or more key terms ("key words") and the search engine identifies pages that match the key words, e.g., pages that contain one or more of the key words. These matched pages are known as a result set.

In many cases, particularly when a query is short, (i.e. contains very few key words) or not well defined, the result set can be quite large, for example, thousands of pages. Alternatively, the pages in the result set regardless of its size may or may not satisfy the end-user's actual information needs. For this reason, most search engines rank order the result set, and only a small number, for example, twenty, of the highest ranking pages are actually returned. Therefore, the quality of search engines can be evaluated not only by the number of relevant pages that are indexed, but also on the usefulness of the ranking process that determines the order in which those relevant pages are returned. A good ranking process will obviously rank relevant pages higher than pages that are less relevant.

Sampling of search engine operations has shown that most queries tend to be quite short, on the average about 1.5 key words (it shall be noted that words like "and" or "the" are not treated as key words). Therefore, there is usually not enough information in the query itself to effectively rank the pages. Furthermore, there may be pages that are very relevant to the search that do not include any of the key words specified in the query. This makes effective retrieval and good ranking problematic.

In one prior art technique, Information Retrieval (IR), some ranking approaches have used feedback by the users. This requires the users to supply relevance information for some of the results that were returned by an initial search to iteratively improve ranking. However, studies have shown that users are generally reluctant to provide relevance feedback. In addition, the data environment of the Web is quite different from the setting of conventional static database oriented information retrieval systems. The main reasons are: users tend to use very short queries and the collection of pages changes continuously.

In yet another prior art technique, an algorithm for connectivity analysis of a neighborhood graph (n-graph) is described by Kleinberg in "Authoratative Sources in a Hyperlinked Environment," Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998. The algorithm analyzes the link structure, or connectivity of Web pages "in the vicinity" of the result set to suggest pages that are relevant to the search.

The "vicinity" of a Web page is defined by the hyperlinks that connect the retrieved page to others. There are two types of links to be analyzed: a Web page can point to other Web pages, and the retrieved Web page can be pointed to by other web pages. Close pages are directly linked

to the retrieved Web page, while farther pages are indirectly linked (e.g. directly linked to a Web page that is in turn directly linked to the retrieved Web page). This connectivity can be expressed as a graph where nodes represent the pages, and the directed edges represent the links. The vicinity of all the pages in the result set combined is called the neighborhood graph.

Specifically, the algorithm attempts to identify "hub" and "authority" pages in the neighborhood graph for a user query. Hubs and authorities exhibit a mutually reinforcing relationship; a good hub page is one that points to many good authority pages, and a good authority page is pointed to by many good hubs. Kleinberg's algorithm constructs a graph for a specified base set of hyperlinked pages. Using an iterative algorithm, an authority weight x and a hub weight y is assigned to each page. When the algorithm converges, these weights are used to rank the pages as authorities and hubs. When a page points to many other pages with large x values, the page receives a large y value and is designated as a good hub. When a page is pointed to by many pages having large y values, the page receives a large x value and is designated as a good authority.

However, there are some problems with Kleinberg's algorithm due to the fact that the analysis is strictly based on connectivity. First, there is a problem of topic drift. For example, if a user composes a query that includes the key words "jaguar" and "car," then the graph will tend to have more pages that mention "car" than "jaguar." These self-reenforcing pages will tend to overwhelm pages mentioning "jaguar" to cause topic drift.

Second, it is possible to have multiple "parallel" edges from pages stored by a single host to the same authority or hub page. This occurs when a single Web site stores multiple copies or versions of pages having essentially the same content. In this case, the single site has undue influence, hence, the authority or hub scores may not be representative.

Third, many Web pages are generated by Web authoring or database conversion tools. Frequently, these tools will automatically insert hyperlinks even though the topics of the Website may not be related. For example, the Hypernews system, which turns Usenet News articles into Web pages, automatically insert links to the Hypernews Web site which may contain topics which are not relevant or even related to the end-user's search.

Therefore, what is needed is a method to reduce the effect of irrelevant pages or overrated pages in a result set. A small, carefully selected subset of pages need to be identified for topic distillation, such that meaningful ranking results can be presented to users in a more timely manner.

SUMMARY OF THE INVENTION

The present invention provides a method and system for indexing documents. More particularly, a method is provided in which the content of the documents linked to or from the indexed document is used. More precisely, the method uses frequencies of occurrence of words used in the linked documents, whereas the prior art methods use the structure of the connections between linked documents and the frequency of words in the indexed document only. All the stored information is parsed and references containing links to the document to be indexed is captured. The captured references are further parsed to collect its content.

Also, a system is provided wherein a processor is configured to parse stored information and capture references containing links to the

document to be indexed. The processor is further configured to parse captured references to collect its content.

The above advantages and features of the invention will be more clearly understood from the following detailed description which is provided in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating the hyperlinked environment of the invention;

Figure 2 is a flow diagram illustrating a process for capturing hyperlinks according to a preferred embodiment of the invention;

Figure 3 is a flow diagram illustrating a process for selective content analysis according to a preferred embodiment of the invention of Figure 2;

Figure 4 is a diagram illustrating a tokenizing process of Figure 3 according to a preferred embodiment of the invention; and

Figure 5 is a diagram illustrating a system according to another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention will be described in connection with an exemplary embodiment of a method and system for indexing documents as illustrated in Figures 1-5. Other embodiments may be utilized and structural or logical changes may be made without departing from the spirit or scope of the present invention. Although the invention is described with respect to documents that are Web pages, it should be understood that the invention can also be worked with any linked data object of a database whose content and connectivity can be characterized. For example, it is anticipated that the present invention can be employed with relational, object oriented databases, or with the collection of the scientific articles indexed by such services as Science Citation Index that use citation as linkages between documents, to name a few. Like items are referred to by like reference numerals throughout the drawings.

Figure 1 shows a distributed network of computers 100 of the present invention. Client computers 110 and server computers 120 (hosts) are connected to each other by a network 130, for example, the World Wide Web. The network 130 includes an application level interface called the Web.

The Web allows the client computers 110 to access documents, for example, multi-media Web pages 121 maintained by the server 120. The location or address of each page 121 is indicated by an associated Universal Resource Identifier (URI) 122. Many of the pages include "hyperlinks" 123 to other pages. The hyperlinks are also in the form of URIs.

In order to help users locate Web pages of interest, a search engine 140 maintains an index 141 of Web pages in a memory, for example, disk storage. In response to a query 111 composed by a user, the search engine 140 returns a result set 112 which satisfies the terms (key words) of the query 111. Since the search engine 140 stores many millions of pages, the result set 112, particularly when the query 111 is loosely specified, can include a large number of qualifying pages. These pages may, or may not satisfy the user's actual information need. Therefore, the order

in which the result set 112 is presented to the client 110 is indicative of the usefulness of the search engine 140. A good ranking process will return "useful" pages before pages that are less so.

We provide an improved ranking method and system 200, 300 utilizing a parser 201 that can be implemented as part of the search engine 140. Alternatively, the method 200 can be implemented by one of the clients 110, or some other computer system on the path between the search engine and the clients. The method of the present invention uses content analysis, as well as connectivity analysis, to improve the ranking of pages in the result set 112.

Referring now to Figures 2 and 3, flowcharts illustrating the steps of the exemplary embodiment of the invention of Figure 1 are shown. In Figure 2, a user inputs a query at 202 and a full URI is sent to a web server 120 and the received data stream is transferred to a parser 201 for capturing tokens (words) and hyperlinks at 204 by breaking down text into recognized strings of characters for further analysis. The parser 201 can be any known parser configured to practice the invention. Parser 201 reads the data stream and parses line by line. The parser 201 looks for the character pattern "href" in each line 206. If there is no such pattern in a

line, the line is skipped 208. If there are such patterns, hyperlinks are individually captured from the line 210. After getting a hyperlink, the parser checks if the hyperlink is a full URI 212. If not, the full URI has to be determined 214. Parser 201 makes every relative URI become a full URI and only a full URI is output by the parser 201. At the same time, the protocol of the hyperlink is checked 216. If it is other than an HTTP protocol, the link is dumped 218. Also, the hyperlink is checked to see if it is in the exclusion list 221. If so, then the link is dumped 222.

Since HTML syntax is not strict, the parser has to precisely get the hyperlinks, and account for any errors in the HTML 220. Also, since web browsers can parse HTML syntax errors to a certain extent, this makes parsing hyperlinks more complicated. Hence, the following rules are observed for the different browser readable forms of HTML links: element names are always case-insensitive; leading and trailing white spaces are ignored in the value of href; quotations for the value of href are ignored as well as leading and trailing white spaces; any white spaces in the value are invalid; if the value of href is quoted and the colon or double slash after protocol name begins with a new line without leading white spaces, the hyperlink is valid; and the end tag is ignored for resource anchors.

Further, since a relative URI is an easy and convenient way for an HTML author to link different pages residing in different directories in the local server, they are very common. This is problematic because, some relative URIs appear different in text but have the same logical links, and they actually point to the same page. Hence, the invention determines the full URI to avoid repeated access to the same page. That is, parser 201 prevents looping during crawling. Thus, the present method calculates the base URI according to the following order (highest priority to lowest). First, the base URI is set by the BASE element, then the base URI is given by meta data discovered during a protocol interaction, and lastly, by default, the base URI is that of the current document.

Similarly, fragment identifiers have a similar function as a relative URI but the scope for the bookmark is within a single page. A fragment identifier is an anchor within a page and can be referred or linked in the same page. An anchor is any object which is fixed so that its position relative to some other object remains the same during repagination. A URI with a fragment identifier is appended a number followed by an anchor name. The present method strips the anchor and gets the links 224 by parsing the URI.

Since some characters are not valid in the URI, like white spaces, double quotations and single quotations. The white space is invalid within URI because that would fragment the URI, and a quotation mark would be mistaken as the end of the URI. The present method utilizes character encoding to resolve this problem 226. That is, some special characters are encoded into another form and decoded by web servers or browsers. For instance, since the space is quite often within URIs with parameters, the space is encoded as a plus sign. This kind of URI (URI with parameters) usually appears in HTTP when a "POST" method is used. Any ASCII characters, except space, may be encoded as a percent sign and two digital hexadecimal code. Furthermore, different kinds of languages may have different encoding schemes.

Next, the method detects any networking and log problems at 228. Most networking problems encountered are, for instance, "pages not found on servers." This error is mostly due to broken links. Web servers usually respond with a web page indicating that pages can not be found. The present method parses this error by reading the HTML title to see whether two key words, "Not" and "Found," are within the title tag 230. Another potential problem is "unknown host", where the server we

specified in the URI can not be located. This error is detected by Java Exception. Lastly, another problem is "no route to host". This occurs when the attempted connection to web servers does not respond for a long time. This is usually because networks are temporarily unavailable or the server is down.

Referring now to Figure 3, the method proceeds with the selective analysis of the content of the documents at 232. It should be noted that certain links could point to a "gif", "ps" or a "pdf" file which are not desired. Therefore, the method first checks the trailing part of a URI to see if it points to a parseable resource 234. If not, the resource is dumped 236. If the resource is readable, it starts to obtain it. When a page is obtained, all the HTML tags are first removed 238.

Next, the remains of the page are tokenized at 240. The tokens corresponding to the words that are not specific to the content of the document (such as "a", "the", "in", etc.) are removed from further analysis 242. This could be achieved by using a stop list, defined by the user that enumerates the words to be removed. Also, the tokenized word often contains some other signs, like a comma(,), colon(:), semicolon(;) and so on. Those which are neither characters nor digits are stripped at 244.

Then, the content of the documents linked to or from the indexed page as described above are stored in a database 246. In one embodiment, a simple in-memory database which stores all the hyperlinks and tokens may be utilized. Another is a stand alone, such as a Hypersonic-SQL database, which stores all the words. The in-memory database is implemented by Java containers, such as, HashMap and HashSet. HashMap can be used for maintaining the linking structure and HashSet can be used for storing the hyperlinks. Every node in the linking structure is associated with a key and a link and can be easily obtained via an associated key. Hypersonic-SQL is a small, high-performance database which easily supports basic SQL needs. Also, Hypersonic-SQL has JDBC drivers which can easily be manipulated.

Referring now to Figure 4, yet another embodiment of the present invention is described. For each document, using a database described above, the system creates a list of most common words by processing each token as follows: (i) if there is no keyword in a database corresponding to the token, a new keyword is entered, otherwise the count for this keyword is increased by one, (ii) all keywords are sorted in decreasing order of their count, and (iii) the keywords with counts

exceeding a threshold value form the frequent word set for this document. Threshold value should be a function of the total number of words in the documents, such as a square root of the number of words or the logarithm of the number of words. A set of words S1 most common in the indexed document D1 is created at 235. Then, a second set of words S2 which is a combination of the frequent word sets in documents D2 directly pointing to document D1 is created at 237. The combination of the frequent word sets may involve conjunction of the sets or use frequency of appearance of a word in the frequent word sets with the same rules of selecting the most frequent words from the sets as are used in selecting the most frequent words from the document as described above. Next, a third set of words S3 which is a combination of the frequent word sets in documents D3 pointing to documents D2 but not to document D1 (as indicated by the crossed-out, dashed arrows) is created at 239. In this way, words that are relevant to document D1 are identified by words which are common both in D1 and in D2 but which are not contained in D3. In other words, words that are relevant to document D1 may be obtained by intersecting set of words S1 with the set of words S2 and eliminating from the result set of words S3. Note, although a three set method is provided, the present

invention is equally applicable to a method utilizing any number of sets greater than three.

Referring now to Figure 5, a processor based system which may be programmed to parse retrieved documents as described above in Figures 1-4 is illustrated. As shown in Figure 5, the processor system, such as a computer system, for example, comprises a central processing unit (CPU) 302, for example, a microprocessor, that communicates with one or more input/output (I/O) devices 308, 310 over a bus 316. The computer system 300 also includes random access memory (RAM) 312, a read only memory (ROM) 314 and may include peripheral devices such as a floppy disk drive 304 and a compact disk drive 306 which also communicates with CPU 302 over the bus 316. Memory 312 can be configured to store the parsed information as described above in Figures 1-4. It may also be desirable to integrate the processor 302 and memory 312 on a single integrated chip.

Hence, the present invention provides a method and system for indexing documents. More particularly, a method in which the contents of the documents linked to or from the indexed page is used. In other words, the method uses a structure of words used by the linked pages, whereas the

prior art methods use only the content of the indexed page and/or the structure of the connections between linked pages.

Although the invention has been described above in connection with exemplary embodiments, it is apparent that many modifications and substitutions can be made without departing from the spirit or scope of the invention. In particular, although the invention is described with web pages, it should be appreciated that the invention is equally applicable to an article, legal case record or any other type of textual document. Accordingly, the invention is not to be considered as limited by the foregoing description, but is only limited by the scope of the appended claims.

What is claimed as new and desired to be protected by Letters Patent of the United States is:

1. A method for indexing a document, comprising the steps of:

parsing stored information;

capturing a first set of references containing links to said document;

capturing a second set of references containing links to said first set of references but not to said document; and

selectively parsing said first and second set of references for tokenizing content.

2. The method of claim 1, wherein said step of tokenizing further comprises the steps of:

forming a first set of words from said document;

forming a second set of words from said first set of references;

forming a third set of words from said second set of references; and

combining common words from said first and second set of words and eliminating from the result words from said third set of words.

3. The method of claim 1, further comprising the step of storing said links and said tokenized contents of said first and second set of references.

4. The method of claim 1, wherein said step of capturing references further comprises the step of determining whether said link is a full URI.

5. The method of claim 4, wherein said step of capturing references further comprises the step of determining whether said link is HTTP protocol.

6. The method of claim 5, wherein said step of capturing references further comprises the step of determining whether said link is in an exclusion list.

7. The method of claim 6, wherein said step of capturing references further comprises the step of accounting for any errors in the HTML.

8. The method of claim 7, wherein said step of capturing references further comprises the step of stripping fragment identifiers.

9. The method of claim 8, wherein said step of capturing references further comprises the step of character encoding said URI.

10. The method of claim 9, wherein said step of capturing references further comprises the step of determining any network problems and parsing said problems.

11. The method of claim 10, wherein said step of selectively parsing said captured references further comprises the step of determining whether said content is a parseable resource.

12. The method of claim 11, wherein said step of selectively parsing said captured references further comprises the step of removing HTML tags from said content.

13. The method of claim 12, wherein said step of selectively parsing said captured references further comprises the step of removing non-specific content.

14. The method of claim 13, wherein said step of selectively parsing said captured references further comprises the step of stripping non-characters and non-digits.

15. The method of claim 1, wherein said stored information is stored as a plurality of pages of information.

16. The method of claim 15, wherein said plurality of pages are Web pages.

17. The method of claim 15, wherein each of the plurality of pages has a said unique URI.

18. The method of claim 3, wherein the stored links and contents are stored in at least one storage device.

19. A method for indexing a document, comprising the steps of:

capturing a first set of references containing links to said document;

capturing a second set of references containing links to said first set of references but not to said document;

forming a first set of words from said document;

forming a second set of words from said first set of references;

forming a third set of words from said second set of references; and

combining common words from said first and second set of words and eliminating from the result, words from said third set of words.

20. The method of claim 19, further comprising the step of storing said links and said tokenized contents of said first and second set of references.

21. The method of claim 19, wherein said step of capturing references further comprises the step of determining whether said link is a full URI.

22. The method of claim 21, wherein said step of capturing references further comprises the step of determining whether said link is HTTP protocol.

23. The method of claim 22, wherein said step of capturing references further comprises the step of determining whether said link is in an exclusion list.

24. The method of claim 23, wherein said step of capturing references further comprises the step of accounting for any errors in the HTML.

25. The method of claim 24, wherein said step of capturing references further comprises the step of stripping fragment identifiers.

26. The method of claim 25, wherein said step of capturing references further comprises the step of character encoding said URI.

27. The method of claim 26, wherein said step of capturing references further comprises the step of determining any network problems and parsing said problems.

28. The method of claim 27, wherein said step of selectively parsing said captured references further comprises the step of determining whether said content is a parseable resource.

29. The method of claim 28, wherein said step of selectively parsing said captured references further comprises the step of removing HTML tags from said content.

30. The method of claim 29, wherein said step of selectively parsing said captured references further comprises the step of removing non-specific content.

31. The method of claim 30, wherein said step of selectively parsing said captured references further comprises the step of stripping non-characters and non-digits.

32. The method of claim 19, wherein said stored information is stored as a plurality of pages of information.

33. The method of claim 32, wherein said plurality of pages are Web pages.

34. The method of claim 32, wherein each of the plurality of pages has a said unique URI.

35. The method of claim 20, wherein the stored links and contents are stored in at least one storage device.

36. A system for indexing a document, comprising:

a parsing means configured to parse stored information;

a capturing means configured to capture a first set of references containing links to said document and a second set of references containing links to said first set of references but not to said document; and

a selectively parsing means configured to parse said document and first and second set of references for tokenizing content.

37. The system of claim 36, wherein said tokenizing is performed by a tokenizing means configured to:

form a first set of words from said document;

form a second set of words from said first set of references;

form a third set of words from said second set of references; and

combine common words from said first and second sets and eliminate common words from the second and third set of words.

38. The system of claim 36, further comprising a storing means configured to store said links and said tokenized contents of said first and second set of references.

39. The system of claim 36, wherein said parsing means is further configured to determine whether said link is a full URI.

40. The system of claim 39, wherein said parsing means is further configured to determine whether said link is HTTP protocol.

41. The system of claim 40, wherein said parsing means is further configured to determine whether said link is in an exclusion list.

42. The system of claim 41, wherein said parsing means is further configured to account for any errors in the HTML.

43. The system of claim 42, wherein said parsing means is further configured to strip fragment identifiers.

44. The system of claim 43, wherein said parsing means is further configured to character encode said URI.

45. The system of claim 44, wherein said said parsing means is further configured to determine any network problems and parse said problems.

46. The system of claim 45, wherein said selective parsing means is further configured to determine whether said content is a parseable resource.

47. The system of claim 46, wherein said selective parsing means is further configured to remove HTML tags from said content.

48. The system of claim 47, wherein said selective parsing means is further configured to remove non-specific content.

49. The system of claim 48, wherein said selective parsing means is further configured to strip non-characters and non-digits.

50. The system of claim 36, wherein said stored information is stored as a plurality of pages of information.

51. The system of claim 50, wherein said plurality of pages are Web pages.

52. The system of claim 50, wherein each of the plurality of pages has a said unique URI.

53. The system of claim 38, wherein the stored links and contents are stored in at least one storage device.

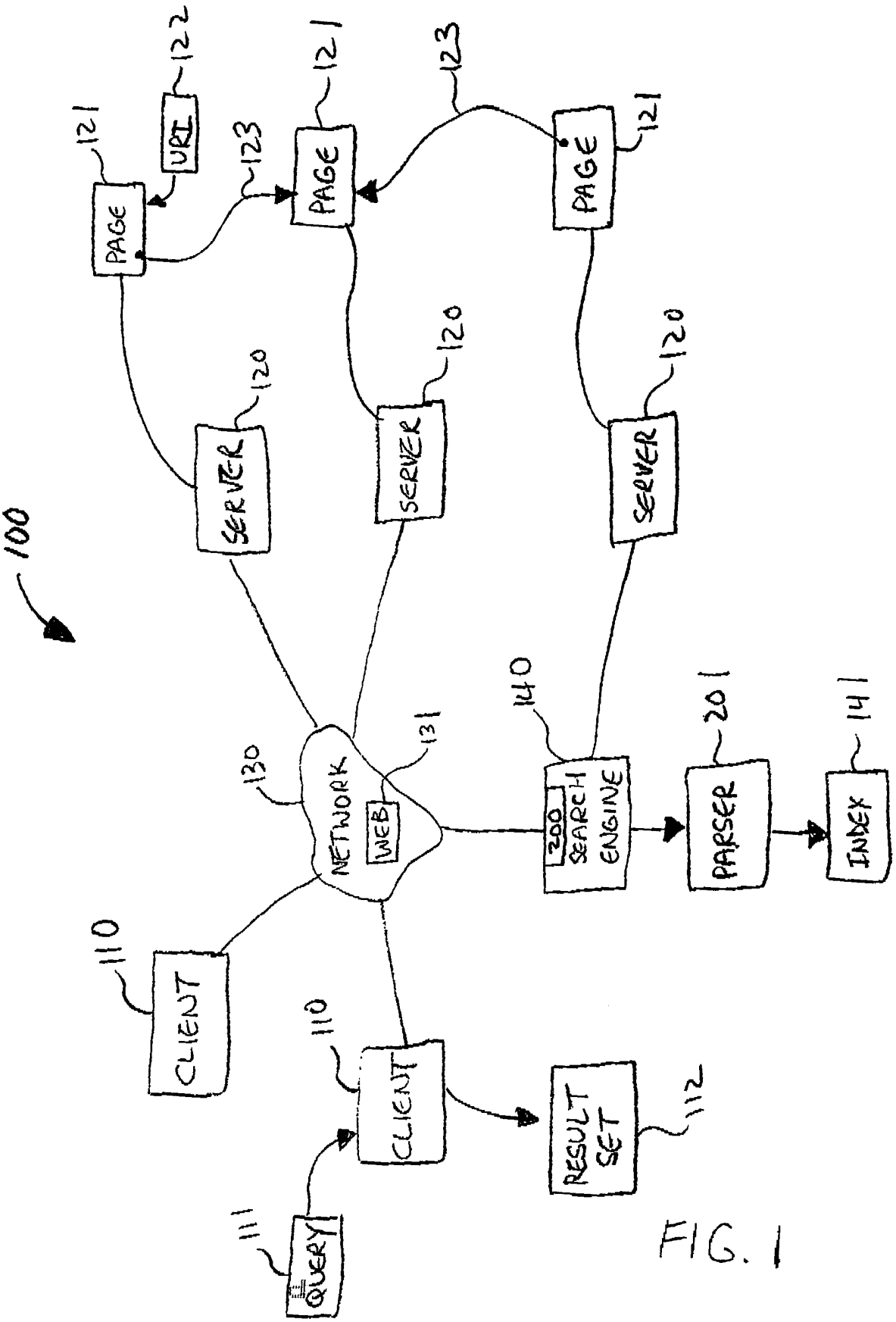

100
PAGE
121
URI
122
123
PAGE
121
PAGE
121
SERVER
120
SERVER
120
SERVER
120
NETWORK
130
WEB
131
200 SEARCH ENGINE
140
PARSER
201
INDEX
141
CLIENT
110
CLIENT
110
QUERY
111
RESULT SET
112

FIG. 1

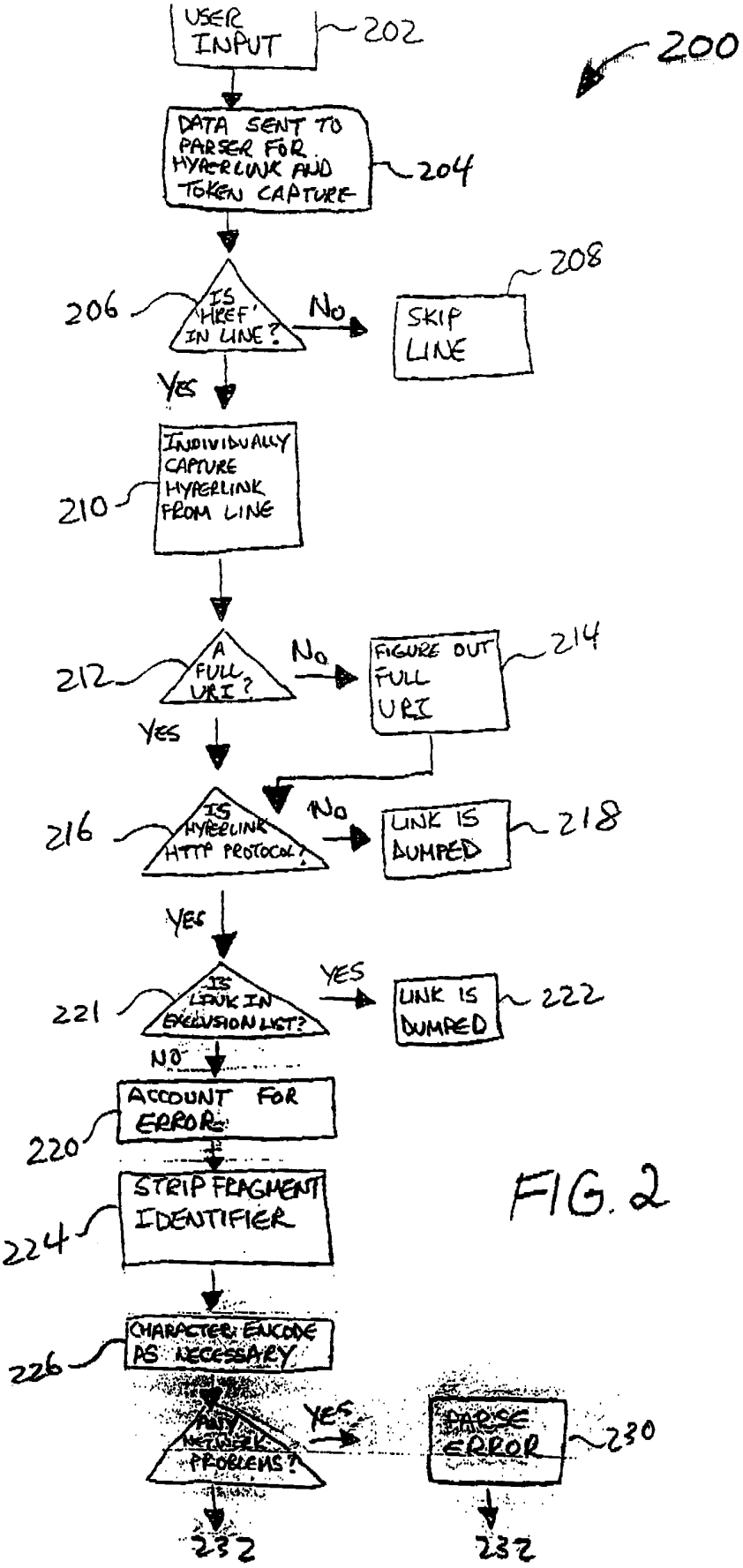

USER INPUT
202
200
DATA SENT TO PARSER FOR HYPERLINK AND TOKEN CAPTURE
204
206
IS 'HREF' IN LINE?
No
208
SKIP LINE
Yes
INDIVIDUALLY CAPTURE HYPERLINK FROM LINE
210
212
A FULL URI?
No
FIGURE OUT FULL URI
214
Yes
216
IS HYPERLINK HTTP PROTOCOL?
No
LINK IS DUMPED
218
Yes
221
IS LINK IN EXCLUSION LIST?
YES
LINK IS DUMPED
222
NO
ACCOUNT FOR ERROR
220
STRIP FRAGMENT IDENTIFIER
224
CHARACTER ENCODE AS NECESSARY
226
ANY NETWORK PROBLEMS?
YES
PARSE ERROR
230
232
232

FIG. 2

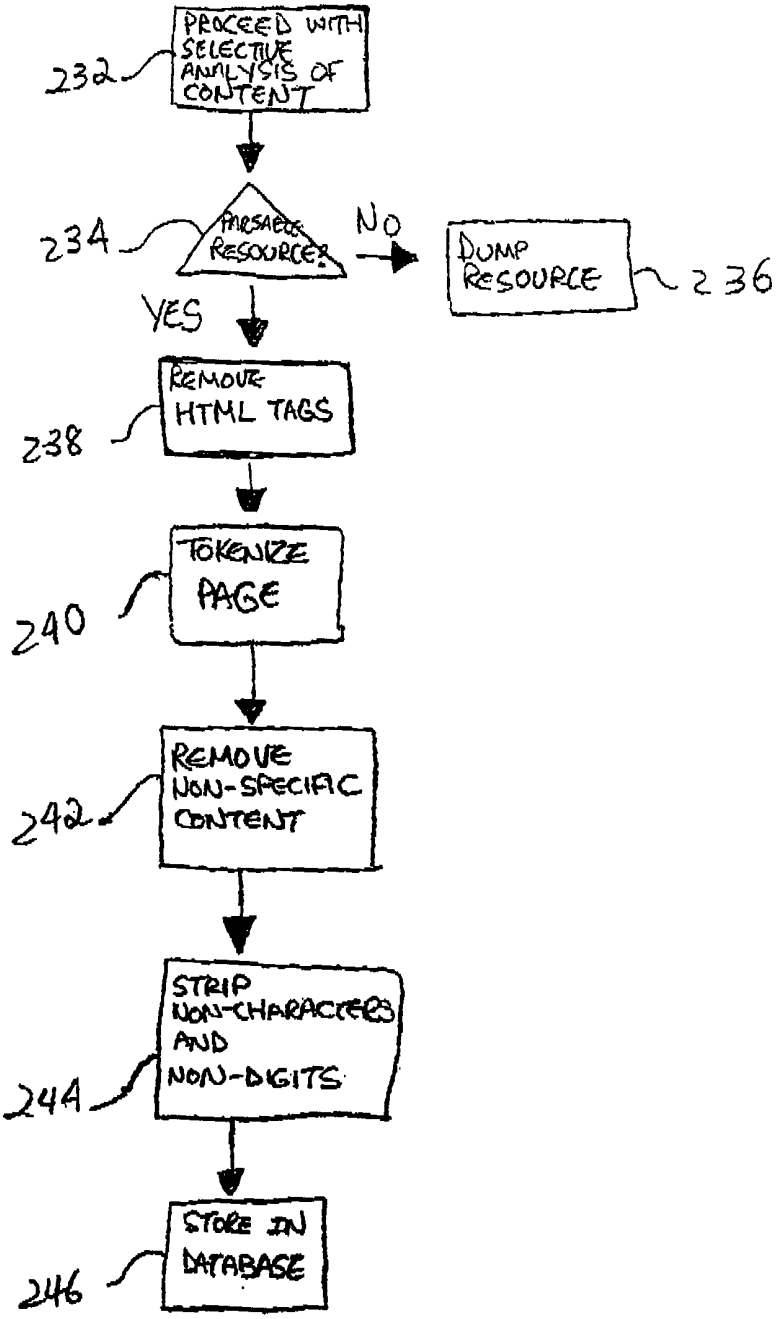

PROCEED WITH SELECTIVE ANALYSIS OF CONTENT
232
PARSABLE RESOURCE?
234
NO
DUMP RESOURCE
236
YES
REMOVE HTML TAGS
238
TOKENIZE PAGE
240
REMOVE NON-SPECIFIC CONTENT
242
STRIP NON-CHARACTERS AND NON-DIGITS
244
STORE IN DATABASE
246

FIG. 3

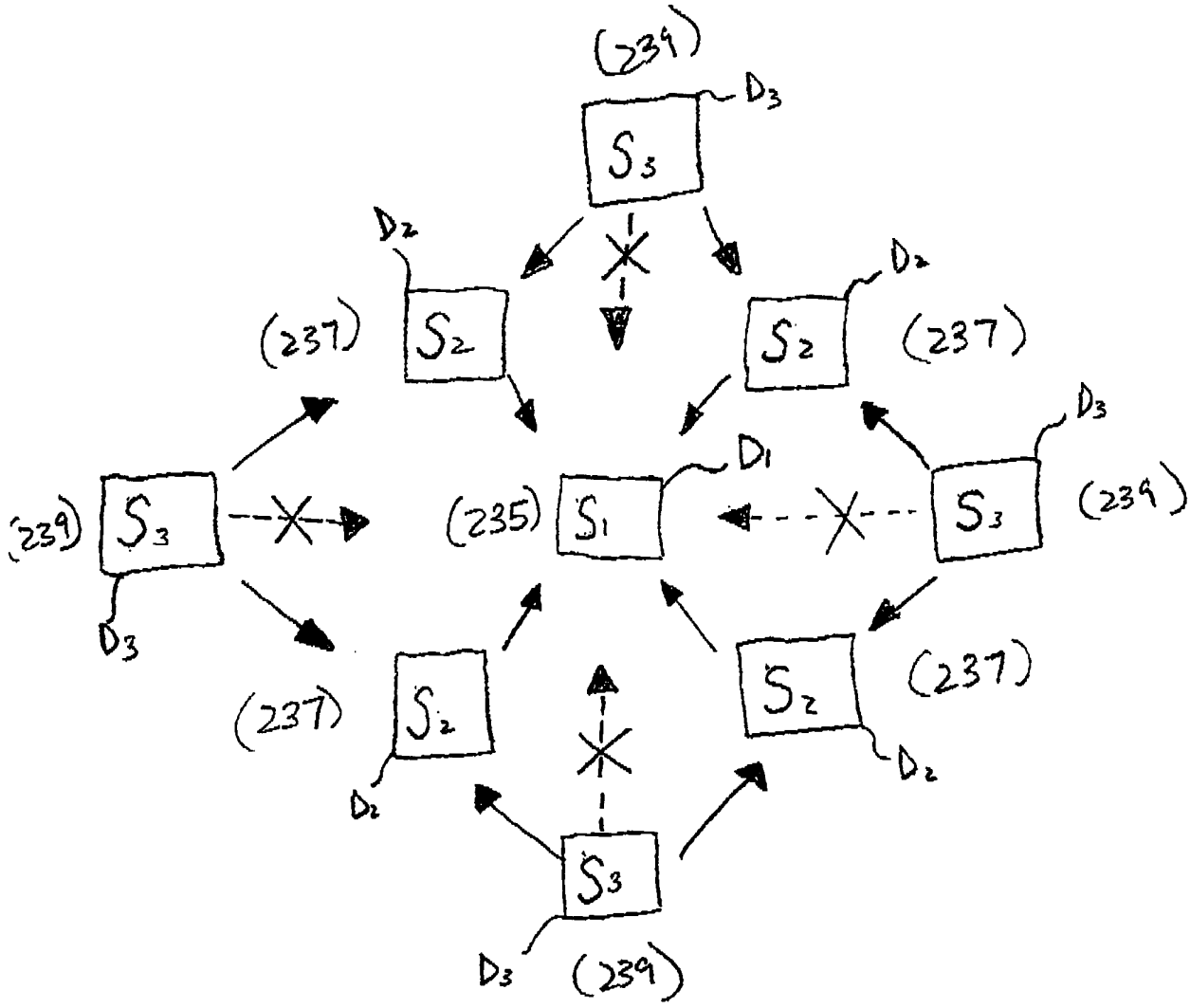

(239)
D3
S3
D2
(237)
S2
S2
(237)
D2
D3
(239)
S3
(235)
S1
D1
S3
(239)
D3
(237)
S2
S2
(237)
D2
D2
S3
D3
(239)

FIG. 4

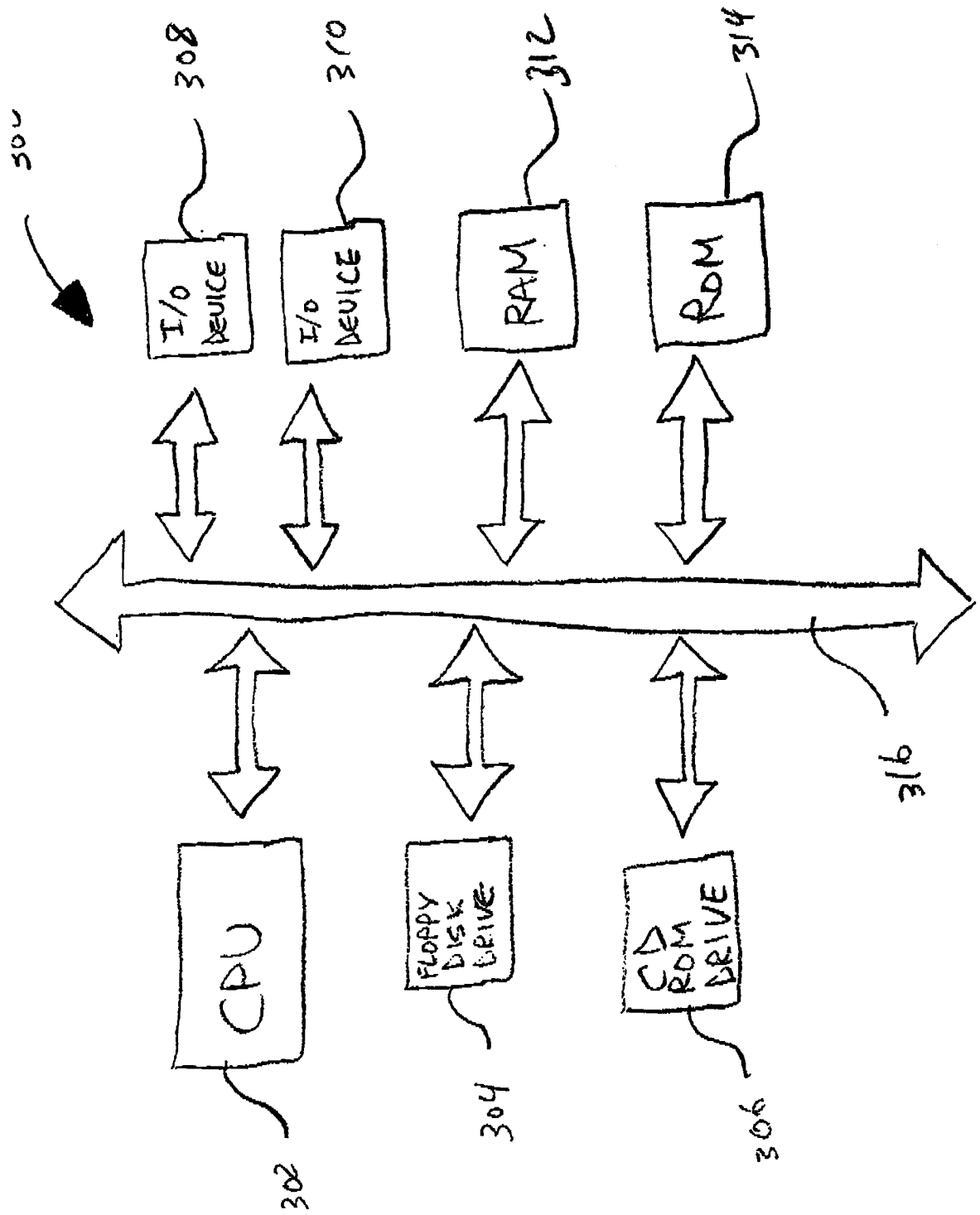

300
308
310
312
314
I/O DEVICE
I/O DEVICE
RAM
ROM
316
CPU
FLOPPY DISK DRIVE
CD ROM DRIVE
302
304
306

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/33340

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :GO6F 17/30
US CL :707/1, 3, 5, 102.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/1, 3, 5, 102.

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,940,624 A (KADASHEVICH et al) 17 August 1999, col. 4, 23-67, col. 5, lines 1-21, col. 6, lines 16-59, col. 7, lines 3-67, col. 8, lines 1-22, col. 9, lines 56-67, col. 10, lines 1-9, col. 13, lines 51-67, and col. 14, lines 1-19.	1-53
Y	US 5,835,905 A (PIROLLI et al) 10 November 1998, col. 1, 23-67, col. 2, lines 1-38, col. 3, lines 19-67, col. 4, lines 1-24, col. 5, lines 2-67, col. 6, lines 1-40, col. 7, lines 35-67, col. 8, lines 1-67, col. 9, lines 1-67, col. 10, lines 55-67, col. 11, lines 7-25, col. 13, lines 51-67, and col. 14, lines 1-10.	1-53

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 22 APRIL 2001	Date of mailing of the international search report 09 MAY 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer ELLA COLBERT Telephone No. (703) 308-7064

Form PCT/ISA/210 (second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/33340

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,838,906 A (DOYLE et al) 17 November 1998, col. 1, lines 46-67, col. 2, lines 1-65, col. 3, lines 52-67, col. 4, lines 1-9 and lines 66-67, col. 5, lines 1-14, col. 6, lines 50-67, col. 7, lines 1-6, col. 8, lines 26-36, col. 9, lines 25-58, col. 12, lines 1-56, col. 13, lines 37-44, and col. 14, lines 10-67.	1-53

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/33340

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

WEST (All Databases)
Search terms: document, indexing, parsing, Internet, World Wide Web, links, tokenizing, link, URI, HTTP protocol, list, HTML, fragment, tags, Web pages.