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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 March 2006 (02.03.2006)

PCT

(10) International Publication Number
WO 2006/023941 A1

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

(21) International Application Number:
PCT/US2005/030024

(22) International Filing Date: 23 August 2005 (23.08.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/603,762	23 August 2004 (23.08.2004)	US
60/623,975	1 November 2004 (01.11.2004)	US
11/122,577	5 May 2005 (05.05.2005)	US

(71) Applicant *(for all designated States except US)*: **WEST SERVICES, INC.** [US/US]; 610 Opperman Drive, Eagan, Minnesota 55123 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CONRAD, Jack G.** [US/US]; 782 Sunset Drive, Eagan, Minnesota 55123 (US). **CLAUSSEN, Joanne R.S.** [US/US]; 8109 129th Street West, Apple Valley, Minnesota 55124 (US). **LIN, Jie** [CN/US]; 925 Scenic Drive, Shoreview, Minnesota 55126 (US).

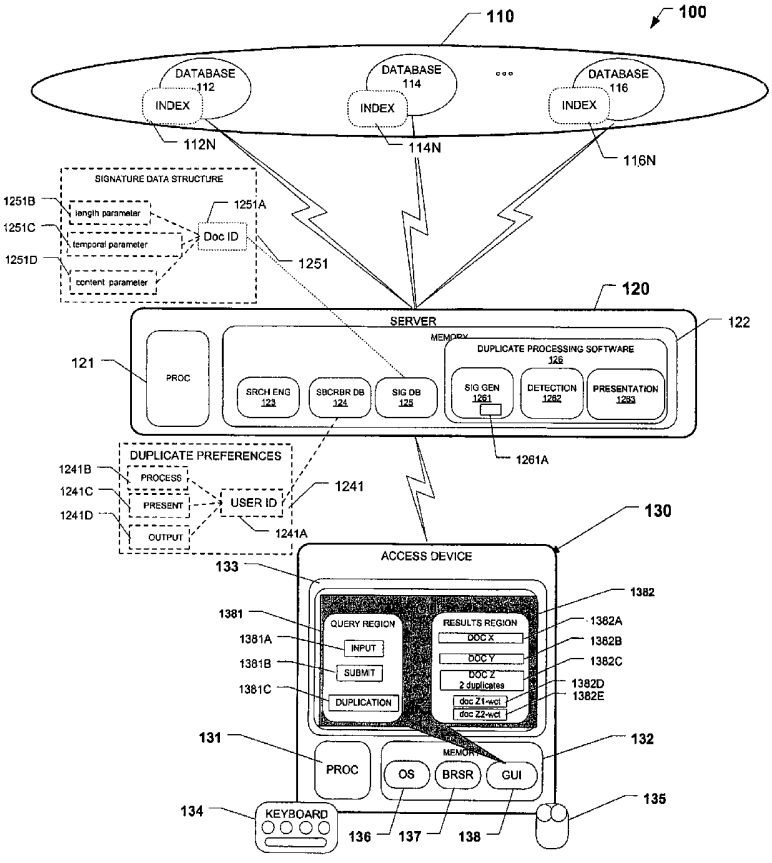
(74) Agents: **STEFFEY, Charles E.** et al.; Schwegman, Lundberg, Woessner & Kluth, P.A., P.O. Box 2938, Minneapolis, MN 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: DUPLICATE DOCUMENT DETECTION AND PRESENTATION FUNCTIONS



(57) Abstract: Many companies provide online search facilities that enable users to conduct computerized searches for documents. Unfortunately, these searches frequently provide results that include duplicate documents---that is, documents that are completely or substantially identical to each other. This problem is particularly vexing when searching news stories, for example. Moreover, the duplicate documents are intermixed in the search results, leaving users to manually manage the complexities of identifying and/or filtering them. Accordingly, the present inventors devised systems, methods, and software that facilitate the identification and/or grouping of duplicate documents in search results. One exemplary system includes a signature generation module which generates document signatures based on length, temporal, and/or content components; a real-time duplicate detection module which uses the document signatures to identify "exact" or "fuzzy" duplicate documents; and a user-interface or presentation module which controls how duplicate documents are presented or suppressed in search results.

WO 2006/023941 A1

FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, 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.

DUPLICATE DOCUMENT DETECTION AND PRESENTATION FUNCTIONS

Related Applications

This application claims priority to U.S. Provisional Application 60/603,762 (Attorney Docket 4962.030PRV) which was filed on August 23, 2004, and to U.S. Provisional Application 60/623,975 (Attorney Docket 4962.030PV2), which was filed on November 1, 2004. Both these applications are incorporated herein by reference.

Copyright Notice and Permission

Technical Field

Various embodiments of the present invention concern information-retrieval systems, such as those that provide news documents or other related content.

Background

Companies, such as Thomson Legal & Regulatory, Inc. of St. Paul, Minnesota (doing business as Thomson West), collect and store a vast spectrum of documents, including news, from all over the world, for online access in a system of databases and research tools, known as the Westlaw™ system. The Westlaw system empowers users to search over 100 million documents.

One problem recognized by the present inventors is that searches conducted against news or other databases frequently provide results that include

duplicate documents---that is, documents that are completely or substantially identical to each other. The problem stems from news providers, such as Associated Press (AP), selling their news stories for re-publication to multiple publishers around the world. This in turn means that systems, such as the Westlaw system, that provide users searchable access to collections of news stories from a wide array of publishers typically present users with many duplicate copies of news stories in their search results. Unfortunately, the duplicate stories are mixed generally according to relevance with other distinct stories, leaving users to manually manage the complexities of identifying and/or filtering them.

Accordingly, the present inventors recognized a need to effectively address how information-retrieval systems, such as the Westlaw system, handle the existence of duplicate documents in their document collections, and more importantly within the search results of their users.

Summary

To address this and other needs, the present inventors devised systems, methods, and software that facilitate the identification and/or grouping of duplicate documents in search results. One exemplary system includes three major components: 1) a signature generation module which generates document signatures based on length, temporal, and/or content components; 2) a duplicate detection module which uses the document signatures to identify "exact" or "fuzzy" duplicate documents; and 3) a user-interface (or presentation) module which allows users to control how duplicate documents are presented in their search results. For instance, users can select whether to exclude duplicates from their search results or to group duplicates together in the results presentation. In some embodiments, the identification and grouping ultimately streamline the process of users interpreting and accessing search results that contain duplicate documents.

Brief Description of Drawings

Figure 1 is a diagram of an exemplary information-retrieval system 100 corresponding to one or more embodiments of the invention;

Figure 2 is a flowchart corresponding to one or more exemplary methods of operating system 100 and one or more embodiments of the invention;

Figure 3 is a diagram of an exemplary document signature and IDF table corresponding to one or more embodiments of the invention;

Figure 4 is a facsimile of an exemplary query window 400 corresponding to one or more embodiments of the invention;

Figure 5 is a facsimile of an exemplary user interface 500 corresponding to one or more embodiments of the invention;

Figure 6 is a facsimile of an exemplary user interface 600 corresponding to one or more embodiments of the invention;

Figure 7 is a a facsimile of an exemplary HTML-formatted electronic message corresponding to one or more embodiments of the invention;

Figure 8 is a facsimile of an exemplary user interface 800 corresponding to one or more embodiments of the invention; and

Figure 9 is a diagram of an exemplary binning scheme that corresponds to one or more embodiments of the invention.

Detailed Description of Exemplary Embodiments

This description, which references and incorporates the above-identified Figures, describes one or more specific embodiments of an invention. These embodiments, offered not to limit but only to exemplify and teach the invention, are shown and described in sufficient detail to enable those skilled in the art to implement or practice the invention. Thus, where appropriate to avoid obscuring the invention, the description may omit certain information known to those of skill in the art.

Exemplary Definitions

The description includes many terms with meanings derived from their usage in the art or from their use within the context of the description. However, as a further aid, the following exemplary definitions are presented.

The term "document" refers to any addressable arrangement of machine-readable data, such as textual data.

The term "database" includes any logical arrangement of documents. In some embodiments, one or more of the databases share content with one or more other databases.

The term "duplicate" generally means any document having a substantial amount of content in common with at least one other document. Temporal, length, and/or content criteria are used to determine whether one document is a duplicate of another. In terms of content criteria, some embodiments identify a set of rarest terms or other features in each of a set of documents and deem those documents that have the same set of rarest terms in similar relative positions as "exact" duplicates. Other embodiments identify a set of rarest terms in each of the documents and deem those documents that surpass a threshold level of overlap among these sets of rarest terms as "fuzzy" duplicates.

Exemplary Information Retrieval System

Figure 1 shows an exemplary online information-retrieval system 100. System 100 includes one or more databases 110, one or more servers 120, and one or more access devices 130.

Databases 110 include representative databases 112, 114, and 116. In the exemplary embodiment, database 112 includes news stories, for example, business and finance information; database 114 includes science and technology information; and database 116 includes intellectual property information, for example U.S. and foreign patents. In some embodiments, one or more of the databases are organized in terms of financial, scientific, or health-care data.

Databases 112, 114, and 116, which take the exemplary form of one or more electronic, magnetic, or optical data-storage devices, include or are otherwise associated with respective indices 112N, 114N, and 116N. Each of the indices includes terms and phrases in association with corresponding document addresses, identifiers, and other conventional information.

Databases 112, 114, and 116 are coupled or couplable via a wireless or wireline communications network, such as a local-, wide-, private-, or virtual-private network, to server 120.

Server 120 is generally representative of one or more servers for serving data in the form of webpages or other markup language forms with associated applets, ActiveX controls, remote-invocation objects, or other related or analogous software and data structures to service clients of various "thicknesses." More particularly, server 120 includes a processor module 121, a memory module 122, a search module 123, a subscriber database 124, a signature database 125, and a duplicate-processing module 126.

Processor module 121 includes one or more local or distributed processors, controllers, or virtual machines. In the exemplary embodiment, processor module 121 assumes any convenient or desirable form.

Memory module 122, which takes the exemplary form of one or more electronic, magnetic, or optical data-storage devices, stores search module 123, subscriber database 124, signature database 125, and duplicate-processing module 126.

Search module 123 includes one or more search engines and related user-interface components, for receiving and processing user queries against one or more of databases 110. In the exemplary embodiment, one or more search engines associated with search module 123 provide Boolean or natural-language search capabilities. Subscription database 124 includes subscriber-related data for controlling, administering, and managing pay-as-you-go or subscription-based access to databases 110. In the exemplary embodiment, subscriber database 124 includes one or more duplicate preference data structures, of which data structure 1241 is representative. Data structure 1241 includes a customer or user identifier portion 1241A, which is logically associated with one or more duplicate-processing preferences, such as preferences 1241B, 1241C, and 1241D. Preference 1241B includes a default value governing whether duplicate detection is enabled or disabled when searching select databases. Preference 1241C includes a default value governing whether duplicates are included when externally outputting search results, for example during printing, saving, or emailing. Preference 1241D includes a default value governing which among two or more duplicate document definitions and/or detection algorithms are used during duplicate detection. In some embodiments, an additional preference is stored to control one or more aspects of a duplicate detection or presentation

function, for example, which of the longest, most recent, or most relevant of a set of duplicate documents to list first.

Signature database 125 includes one or more document signature data structures, such as a representative data structure 1251, for each document in one or more of databases 110. In the exemplary embodiment, data structure 1251 includes a document identifier field or portion 1251A that is logically associated with one or more length-related fields or portions 1251B, one or more time-related fields or portions 1251C, and/or one or more content-related portions 1251D. (As used herein, time-related fields broadly encompass time and/or date.) The structure and contents of these fields is defined by duplicate-processing module 126.

Duplicate-processing module 126 includes a signature-generation module 1261, a duplicate-detection module 1262, and a duplicate-presentation module 1263. (In some embodiments, the software modules or components thereof are distributed across multiple servers.) Signature-generation module 1261 includes one or more inverse-document-frequency (idf) tables, of which idf table 1261A is generally representative. The exemplary embodiment uses a binary-encoded idf table having approximately one million terms, with the terms selected from a one-third sampling of a combined set of relevant document collections. In the table, each idf term is associated with a corresponding three-byte (24-bit) representation of an ordinal number (or serial number), such that the idf term can be uniquely represented by its corresponding ordinal or serial number rather than actual text representations, thereby facilitating rapid processing and reducing storage requirements for the idf table. (The constant number of bytes used to represent each idf term is a function of the total terms in the table.) The exemplary idf table excludes numeric tokens, alphanumeric tokens, tokens with special characters, such as . , - \ & +; and tokens with less than three characters. However, some embodiments may elect to include these tokens. (Further description of the signature-generation, duplicate-detection, and duplicate-presentation modules and their exemplary operation is provided below with the aid of Figure 2.)

Server 120 is communicatively coupled or couplable via a wireless or wireline communications network, such as a local-, wide-, private-, or virtual-private network, to one or more access devices, such as access device 130.

Access device 130 is generally representative of one or more access devices. In the exemplary embodiment, access device 130 takes the form of a personal computer, workstation, personal digital assistant, mobile telephone, or any other device capable of providing an effective user interface with a server or database. Specifically, access device 130 includes a processor module 131, a memory 132, a display 133, a keyboard 134, and a graphical pointer or selector (or mouse) 135.

Processor module 131 includes one or more processors, processing circuits, or controllers. In the exemplary embodiment, processor module 131 takes any convenient or desirable form. Coupled to processor module 131 is memory 132.

Memory 132 stores code (machine-readable or executable instructions) for an operating system 136, a browser 137, and a graphical user interface (GUI)138. In the exemplary embodiment, operating system 136 takes the form of a version of the Microsoft Windows operating system, and browser 137 takes the form of a version of Microsoft Internet Explorer. Operating system 136 and browser 137 not only receive inputs from keyboard 134 and selector 135, but also support rendering of GUI 138 on display 133. Upon rendering, GUI 138 presents data in association with one or more interactive control features (or user-interface elements). (The exemplary embodiment defines one or more portions of interface 138 using applets or other programmatic objects or structures from server 120.)

More specifically, graphical user interface 138 defines or provides one or more display regions, such as a query region 1381 and a search-results region 1382. Query region 1381 is defined in memory and upon rendering includes one or more interactive control features (elements or widgets), such as a query input region 1381A, a query submission button 1381B, and a duplicate processing selection 1381C. Search-results region 1382 is also defined in memory and upon rendering includes one or more interactive control features, such as features 1382A, 1382B, 1382C, 1382D, and 1382E, for accessing or retrieving

one or more corresponding documents from one or more of databases 110 via server 120.

Each control feature includes a respective document identifier or label, such as DOC X, DOC Y, DOC Z, DOC Z1 and DOC Z2, identifying a corresponding document and associated with a corresponding link or whole or partial uniform resource locator (URL). (Some embodiments use a URL format, such as that taught in co-pending U.S. patent application 09/237,219 (Attorney Docket 962.002US1), which was filed on January 25, 1999 and which is incorporated herein by reference.) User selection of the control features results in retrieval and display of at least a portion of the corresponding document within a region of interface 138 (not shown in this figure.) Control features 1382D and 1382E are indented relative to control feature 1382C to indicate the status of their corresponding documents DOC Z1 and DOC Z2 as duplicates of DOC Z, the document corresponding to control feature 1362C. Control feature 1382C includes a label "2 duplicates" indicating existence of two duplicate documents. In the exemplary embodiment, each of these control features takes the form of a hyperlink or other browser-compatible command input, and provides access to and control of query region 1381 and search-results region. Although Figure 1 shows query region 1381 and results region 1382 as being simultaneously displayed, some embodiments present them at separate times.

Exemplary Methods of Operation

Figure 2 shows a flow chart 200 of one or more exemplary methods of operating a system, such as system 100. Flow chart 200 includes blocks 210-270, which, like other blocks in this description, are arranged and described in a serial sequence in the exemplary embodiment. However, some embodiments execute two or more blocks in parallel using multiple processors or processor-like devices or a single processor organized as two or more virtual machines or sub processors. Some embodiments also alter the process sequence or provide different functional partitions to achieve analogous results. For example, some embodiments may alter the client-server allocation of functions, such that functions shown and described on the server side are implemented in whole or in part on the client side, and vice versa. Moreover, still other embodiments

implement the blocks as two or more interconnected hardware modules with related control and data signals communicated between and through the modules. Thus, the exemplary process flow (in Figure 2 and elsewhere in this description) applies to software, hardware, and firmware implementations.

At block 210, the exemplary method begins automated generation of metadata (such as digital signatures) for one or more searchable documents of the online information-retrieval system. In the exemplary embodiment, this proceeds according to a batch process for documents in a select set of databases, such as news databases. (In some embodiments, the process is executed on a per-document, and/or real-time query-driven basis.) The batch process generally entails generating and storing, for each document, a document signature data structure. The exemplary embodiment uses one of two or more signature-generation processes generally represented by flow charts 210A and 210B. (In some embodiments, which may compute signatures in real-time, the choice of process is governed via a user preference; however in other embodiments, both processes are used to provide each document with two document signatures and user preferences are used during detection to determine which signature or detection method to use. In some embodiments, the choice of signature is an administrative decision.) Flow chart 210A shows generation of signatures that facilitate detection of duplicates according to a more exact duplicate standard, whereas flow chart 210B shows generation of signatures that facilitate detection of duplicates according to a less exact or "fuzzy" standard.

Exact Signature Generation

More particularly, flow chart 210A, which yields a signature having a length scalar and a fingerprint (for example, a hash value), includes process blocks 211A-216A. The process begins at block 211A which entails determining one or more document length features or values. To this end, the exemplary embodiment determines a length scalar, which is defined as the document length in tokens, excluding newspaper, title, author and other header information.

Next, block 212A entails determining or identifying and ranking one or more semantic or lexical (more generally, content) values for the document. In

the exemplary embodiment, this entails determining a 'fingerprint' or term vector which is defined to include the top X, for example six, unique highest-ranking inverse-document-frequency (idf) terms for the document (excluding title, author, and other header information or metadata.) The idf for a given term is defined as the reciprocal of the document frequency of the term, that is, the inverse of the number of documents in the collection under consideration that contain the term. Some embodiments use a normalized IDF which is defined as

$$IDF = \frac{\log(\frac{N+0.5}{n})}{\log(N+1.0)} \qquad (1)$$

where n denotes the number of documents containing the given term; N denotes the total number of documents in the collection; and the constants in the numerator and denominator serve as scaling factors where there is sparse data.

In specifically defining the vector, the exemplary embodiment excludes terms in the document title and other headings from consideration as a top idf term, because these may vary significantly in duplicate newspaper articles. Also, it excludes terms with an unusually high idf--for example terms having an idf greater than or equal to 0.8--from the top X idf terms because these tend to be textual aberrations, such as typos and misspellings. (Other embodiments may use greater or fewer numbers of terms and/or greater or lesser idf-exclusion criteria. Some embodiments may not use any idf-exclusion criteria. Some embodiments may even use phrases or combinations of terms, such as word pairs, rather than single terms or phrases in combination with single terms.) Once the content features are identified and ranked, execution continues at block 213A.

Block 213A entails determining relative positions of the idf terms within the document. The positions can be defined as absolute or relative positions. Absolute position is the position of a term relative to the first token in the document. One sample vector includes the following:

prevarication[76],
hostage[0],
conspicuous[25],

intransigence[121],
brutality[163],
theater[13]

wherein the terms are ranked and presented in descending order of idf value and the position, shown in brackets, is measured in tokens relative to a first token in the document.

Some embodiments measure position of each idf term relative to the position of the preceding highest-ranked idf term. And, some embodiments, which provide a relaxed or more tolerant definition of duplicate documents, round the relative positional offsets into bins. For instance, one embodiment rounds the positional offset for each corresponding top idf term into the closest of a series of ten-token "bins," with the number of bins determined by the size of the document in tokens divided by ten. This positional binning is effective in allowing this embodiment to handle cases where a document has been subjected to insertions or short substitutions and would still generally be regarded as a duplicate of another different, but otherwise identical document. The table below shows a sample set of six idf terms presented in descending rank order along with their original (or absolute) position within the document in the second column, their relative positions in the third column, and their binned (or rounded) relative positions in the fourth column.

idf term (descending rank order)	Original document position	position relative to preceding ranked idf term	"binned" relative position
irate	11	11-0 = 11	20
flabbergasted	35	35-11 = 24	30
dishonorable	62	62-35 = 27	30
disgraceful	67	67 – 62 = 5	10
outrageous	80	80 - 67 = 13	20
ignoble	120	120 – 80 = 40	40

Note that all rounding used in the positional binning is upward, and that differences ending in 0 are left untouched (for example, see binned relative position for ignoble in the table.) Also note that in this embodiment the terms present in the title of the document (and in any of the accompanying headings

and subheadings) do not participate (that is, are not counted in) the generation of the offsets within the document. After defining the fingerprint, execution advances to block 214A.

Block 214A entails determining a hash value (or other unique value) based on the fingerprint. Specifically, the exemplary embodiment concatenates the top idf-terms and positional information into a single string, such as "irate20flabbergasted30dishonorable30disgraceful10outrageous20ignoble40," and then hashes the resulting string according to the algorithm to determine the hash value. In the exemplary embodiment, this entails hashing the vector into a 20-byte key using National Institute of Standards and Technology's SHA1 hashing algorithm. Some embodiments may use other methods of determining a hash value.

Block 215A entails forming or defining a document signature (that is, data structure) based on one or more of the length values and the determined fingerprint number (such as hash value based on content.) In the exemplary embodiment, this entails aggregating the scalar length value and the fingerprint number into a data structure that is logically associated with the corresponding document, for example using a document identifier or pointer.

Block 216A entails storing the document signature data structure in a memory device. To this end, the exemplary embodiment stores the data structure in an index or metadata database, such as document signature database 125 in Figure 1.

Fuzzy Signature Generation

Flow chart 210B---which shows generation of a document signature data structure (or characteristic feature set) based on document temporal, length, and content components---includes process blocks 211B-215B.

Specifically, block 211B entails determining one or more temporal components or values for the document. In the exemplary embodiment, this determination entails extracting a publication date or time stamp from the dateline of a document, and then converting the date or time stamp to a simple integer representing the number of hours, days, weeks, or months relative to a reference date, such as January 1, 1950. Other documents may

use other dates associated with or contained in the document, such as a first or last occurring date in the document or a portion of the first or last occurring date. For example, some embodiments may extract and use the first occurring year in the document as a basis for the temporal component. Still others may determine a temporal component based on multiple dates within the document, for example an average or aggregate of two or more dates. Execution continues at block 212B.

Block 212B determines one or more length components or values for the document. In the exemplary embodiment the length value is based on length of the document, and is determined by extracting a document-length indicator from a predefined word-count field associated with the document. However, other embodiments independently determine a word count and use this as the length value. Execution proceeds to block 213B.

Block 213B entails determining one or more content values or features for the document. In the exemplary embodiment, this entails identifying one or more lexical features for the document and forming a 'fingerprint' or term vector. The fingerprint generally includes the top Y, for example 60, highest-ranking idf terms for the document (excluding title, author, and other header information or metadata).

More specifically, the exemplary embodiment tokenizes and parses a document into terms, and then sorts these terms by their associated idfs. Any terms that are not in the idf table plus any terms included on a list of stop words are excluded from the term (or feature) vector. If the number of idf terms y in a document is less than 10, no signature is created for the document. However, if y is between 10 and Y-1 inclusive, the exemplary embodiment pads the term vector with additional terms to ensure inclusion of Y terms.

Exemplary padding proceeds as follows. If the number of idf terms y for a document falls between 30 and 59 inclusive, the exemplary embodiment pads by adding up to 30 different alphanumeric terms to the vector, such as 'pad1', 'pad2', 'pad3', ..., 'pad*n*', where n = Y- y. If y is between 10 and 29 inclusive, the exemplary embodiment pads the term vector with some combination of n predetermined non-idf terms and m randomly selected non-

idf terms, such that y + n + m = Y. The n predetermined non-idf-tabled terms, in some embodiments, are (as above) alphanumeric terms, such as 'pad1', 'pad2', ..., 'padn', which have a common text portion and a sequenced numeric portion. The m randomly generated non-idf-tabled terms, in some embodiments, are alphanumeric terms, R1, R2, Rm, which represent terms that are not matchable with existing idf terms. The table below illustrates the padding scheme used in the exemplary embodiment.

# of idf terms, y	**# of non-random pad terms, n**	**# of non-matchable terms, m**
y >= 60	0	0
30<= y <= 59	pad1, pad2, ..., padn, n=Y-y	0
28 or 29	pad1, pad2, ..., padn, n=Y-y-1	1
26 or 27	pad1, pad2, ..., padn, n=Y-y-2	2
24 or 25	pad1, pad2, ..., padn, n=Y-y-3	3
22 or 23	pad1, pad2, ..., padn, n=Y-y-4	4
20 or 21	pad1, pad2, ..., padn, n=Y-y-5	5
18 or 19	pad1, pad2, ..., padn, n=Y-y-6	6
16 or 17	pad1, pad2, ..., padn, n=Y-y-7	7
14 or 15	pad1, pad2, ..., padn, n=Y-y-8	8
12 or 13	pad1, pad2, ..., padn, n=Y-y-9	9
10 or 11	pad1, pad2, ..., padn, n=Y-y-10	10
1<=y<=9	no signature created	no signature

Next, block 214B encodes the term vector. To this end, the exemplary embodiment encodes each term separately, with the encoding based on position of the term within the ranked idf table. Specifically, the exemplary embodiment encodes each term vector token as a unique three-byte (24 bit) serial number or index that not only uniquely corresponds with the token in the idf table (which includes about one million entries), but also (in the exemplary embodiment) indicates the rank of the term in the idf table. (The exemplary embodiment organizes the terms in the vector from highest idf value to lowest to accelerate mismatch count and thus reduce computation time.) After encoding the term vector, execution advances to block 215B.

Block 215B entails storing the document signature data structure in a memory device. To this end, the exemplary embodiment stores the signature in a metadata database, such as signature database 125 (in Figure 1). (Some embodiments append the document signature data to the document.)

Some embodiments address issues surrounding the maintenance of collection statistics and idf table updates. For example, one embodiment, which creates signatures according to flow chart 210A and/or 210B, recognizes the sensitivity of these document signatures to updates of the idf table that typically occur with addition, deletion, or correction of documents in a given collection or database. This embodiment provides a pair of signatures for those documents published close to an idf-table update date (for example, within a two-month window centered on the update date.) One signature, a pre-update signature, is based on the idf table prior to update, and the other, a post-update signature, is based on the idf table after update. (In use, for example at block 250, if at least one of the two signatures matches at least one of the signatures for another document, the two documents are regarded as duplicates.)

Figure 3 shows how a document signature 300 is related to an idf table 340 via its content component 310. Specifically, document signature 300 includes a length component 310, a temporal component 320, and a content component 330. Content component 330 takes the exemplary form of a 60-term vector 330', which includes terms T0-T59. As shown in Figure 3 one or more of the terms, for example all of the terms, map to terms in idf table 340, which has one million terms and corresponding idf values.

After generating and storing document signatures for all the documents of the select databases according to one or both of the methods shown in flow charts 210A and 210B, exemplary execution eventually proceeds to block 220.

Block 220 entails presenting a search interface to a user. In the exemplary embodiment, this entails a user directing a browser in a client access device to an internet-protocol (IP) address for an online information-retrieval system, such as the Westlaw system, and then logging onto the system. Successful login results in a web-based search interface, such as interface 138 in Figure 1 or interface 300 in Figure 4 (or one or more portions thereof) being output from server 120, stored in memory 132, and displayed by client access device 130.

As Figure 4 shows, interface 400 includes a number of interactive control features, including a query input region 410, a query field restriction region 420, a duplicate directive region 430, and a query submit command 440. Query input

region 410 receives textual input defining a query. Query targeting region 420 allows the user to target the query to a specific subsection of documents, the headlines and lead paragraphs, in a fielded database. (Other embodiments may have one or more other selectable subsections.) Duplicate-directive region 430 allows the user to specifically enable identification of duplicate documents within search results for the query being defined. The initial state of this directive region is determined by a default user preference value stored in a subscriber database, such as database 124. Changing the state of the directive region, in the exemplary embodiment, changes the directive for the current query; the default preference value is unaffected unless changed at a higher control level.

Using interface 138 or 400, the user can define or submit a query and cause it to be output to a server, such as server 120. In other embodiments, a query may have been defined or selected by a user to automatically execute on a scheduled or event-driven basis. In these cases, the query may already reside in the memory of a server for the information-retrieval system (such as done for clipping services), and thus need not be communicated to the server repeatedly. Execution then advances to block 230 (in Figure 2.)

Block 230 entails receiving a query. In the exemplary embodiment, the query includes a query string and/or a set of target databases, which includes one or more of the select databases. In some embodiments, the query string includes a set of terms and/or connectors, and in other embodiment includes a natural-language string. Also, in some embodiments, the set of target databases is defined automatically or by default based on the form of the system or search interface. Also in some embodiments, the received query may be accompanied by other information, such as information defining whether to check for duplicate documents as discussed above. In any case, execution continues at block 240.

Block 240 entails identifying a set of documents or search results based on or in response to the received query. In the exemplary embodiment, this entails the server or components under server control or command, executing the query against the targeted set of databases and identifying documents that satisfy the query criteria. Execution proceeds to block 250.

Block 250 entails identifying sets of duplicate documents in the search results. (In some embodiments, execution of the duplicate identification block is contingent on a default or selected user option that specifies whether to identify duplicate documents. Some embodiments allow users or administrators to select which of two or more duplicate-detection techniques or algorithms to use.) In the exemplary embodiment, duplicate identification generally entails comparing one or more aspects of one or more document signatures to corresponding aspects of other document signatures and determining whether documents are duplicates based on "exact" or "fuzzy" (less exact) standard of what constitutes a duplicate document.

More specifically, the exemplary embodiment follows the method shown in flow chart 250A, which includes process blocks 251A-254A for the exact or more exact detection algorithm, or the method shown in flow chart 250B, which includes process blocks 251B-255B for a "fuzzy" detection algorithm. Some embodiments which use both types of signatures described above use both corresponding methods.

Exact Duplicate Detection

In flow chart 250A, the exemplary method begins at block 251A, which entails selecting two or more documents of the search results for comparison. In the exemplary embodiment, this entails retrieving document signature data structures for each document in the search results based on their document identifiers and defining a number of document pairs for real-time duplicate detection or comparison. Defining the document pairs entails selecting a primary document and pairing the primary document (or more precisely its document signature) with each of the other documents in the search results, then selecting a second primary document and pairing it with all other documents with which it has not already been paired. Similarly, each document can be selected as a primary document and paired with all other documents with which it has not already been paired, ultimately defining a complete set of unique document pairings for comparison. (In some embodiments, the primary documents are selected in order of their relevance ranking within the search results. Also, some embodiments restrict application of the duplicate detection

process to documents that exceed a certain relevance threshold or that have a certain minimum ranking.) Execution then advances to block 252A.

Block 252A entails determining whether the length criteria for the set of documents selected for comparison are satisfied. In the exemplary embodiment, this entails determining whether the length scalars of a selected pair of document signature data structures are within a predetermined range, such as ±40 tokens or ±10% of each other. The fixed or relative range allows for potential differences in documents that are near the header material, for example, Dateline: Amsterdam. If the determination is that the length criteria (or condition(s)) are not satisfied, execution returns to block 251A for selection of another set of documents for comparison. However, if the length criteria are satisfied, execution advances to block 253A.

Block 253A determines whether content criteria for the selected documents are satisfied. In the exemplary embodiment, this entails comparing the document fingerprints of the selected documents to each other. If the two fingerprints are not identical, execution returns to block 251A for selection of another set of documents for comparison. If the fingerprints are identical, execution advances to block 254A.

Block 254A entails marking the selected documents as duplicates of each other. In the exemplary embodiment, this marking entails storing the document identifiers for the documents deemed to be duplicates in a duplicate-set buffer. (In other embodiments, the marking includes adding the document identifiers for the selected documents to a master duplicate document database or to the respective document signature data structures for the selected documents, which may be reused for special duplicate-document queries.)

Execution then returns to block 251A for the selection of the next set of documents for comparison. In the exemplary embodiment, block 251A includes logic for terminating the comparison process after all selected sets of documents are processed.

Fuzzy Duplicate Detection

Flow chart 250B depicts an alternative detection process, which generally entails a real-time multilevel processing of the signature data structures

for the documents identified in the search results. (Some embodiments may perform duplicate detection prior to, rather than in response to a user query.) Flow chart 250B includes process blocks 251B-255B.

At block 251B, the process ensues with retrieval of at least two document signature data structures for documents identified in the search results. In the exemplary embodiment, this entails retrieving signature data structures from signature database 125 (shown in Figure 1) for each document in the search results in a manner similar to that described for block 251A. Once the signatures are retrieved, a set of two or more document signatures are selected for use in determining whether their corresponding documents are duplicates. Execution then continues at block 252B.

Block 252B determines whether the temporal criteria for components associated with the two or more document signatures are within a certain time period of each other. In the exemplary embodiment, this entails determining whether the temporal components of the selected document signatures are within 30 days of each other. (Some embodiments use smaller or larger temporal windows.) A negative determination results in the documents being deemed as non-duplicates and execution returning to block 251B to get another set of document signatures for comparison, whereas a positive determination extends processing to block 253B.

Block 253B determines whether the length components of the signature data structures for one or more corresponding documents are within some range of each other. In the exemplary embodiment, this entails determining in real-time whether the length components are within ±20% of each other. If the length determination is negative, the documents are considered as non-duplicates and execution returns to block 251B to select another set of document signatures for comparison. However, if the determination is positive, indicating that the document lengths are sufficiently close, execution continues at block 254B.

Block 254B entails determining whether the set of document signatures satisfy the fingerprint or content criteria for duplicate documents. In the exemplary embodiment, this entails determining whether at least 80% of the terms in the term vector for one document are contained within the term vector for the other document. Some embodiments may use alternative thresholds, for

example, 60, 65, 70, 75, 80, 85, 90, or 95%. Other embodiments may use dynamic thresholds that vary based on type of documents or database; indeed some embodiments allow users to select the threshold.

More particularly, in determining whether the content criteria are met, the exemplary embodiment performs a real-time term-by-term comparison, incrementing a mismatch counter for every term mismatch and a match counter for every match. The term-by-term comparison is terminated when the mismatch counter exceeds a non-duplicate threshold, such as 12, which indicates that the content criteria cannot be satisfied (since, for example, more than 20% of the terms have not matched). Alternatively, the term-by-term comparison could terminate when the match counter meets a duplicate threshold, such as 49, which indicates that more than 80% of the terms have matched. In any event, if the content criteria are not satisfied, execution returns to block 251B. And, if the content criteria is satisfied, execution proceeds to block 255B.

Block 255B entails marking the documents in the document sets as duplicates. In the exemplary embodiment, this marking entails storing the document identifiers for the documents deemed to be duplicates in a duplicate-set buffer. After identifying duplicates within the search results, execution proceeds to block 260.

Block 260 entails presenting search results to a user. In the exemplary embodiment, this entails outputting the search results in the form of a ranked list to a client access device, such as client access device 130 in Figure 1. Specifically, the exemplary embodiment provides the list within the context of a graphical user-interface, as exemplified by interface 138, specifically region 1382 in Figure 1 or alternatively by interface 500 in Figure 5.

Interface 500 shows a results list 510 of selectable document citations or hyperlinks and window 520 for displaying text of at least one of the documents in results list 510. Results list 510 includes one or more sets or groupings of duplicate document identifiers or citations, of which a duplicate set 512 is representative. In the exemplary embodiment, the position of set 512 within results list 510 is determined based on reverse chronological listing, relevance rank, or score of the highest ranked document in or associated with the set. For example, if the original set includes documents 3

and 5 (in rank order) that are duplicates and document 5 is the primary document, the entire duplicate set is displayed in the number 3 position in the cite list. Set 512 includes a primary document identifier 5121, duplicate count indicator 5122, and a duplicate identifier 5123.

Primary document identifier 5121 identifies a primary document, which in the exemplary embodiment is defined as the longest of the documents in the duplicate set. If two or more duplicate documents are the same length (that is, have the same word count), then the more recent document is listed as the primary document. In some embodiments, the primary document identifier corresponds to the document having the highest relevance score or the most recent publication date.

Duplicate count indicator 5122 indicates the number of documents in the search results that are considered to be duplicates of the primary document---that is, to include substantial amounts of content that are duplicative of content of the primary document. In the example shown, the primary document is indicated to have one duplicate in the results list.

Duplicate listing 5123 lists one or more selectable document citations or identifiers, such as document citation or identifier 5123A, each associated with a document (or document URL) deemed relevant to the search query and considered to include a substantial amount of content that is duplicative of content in the primary document. (In the exemplary embodiment, duplicates are determined in accord with the techniques and principles presented herein; however, other embodiments may use alternative techniques and/or principles.) In the duplicate listing, the order of listing multiple duplicates can be determined by length, publication date, document relevance, or access frequency. In some embodiments, the labels for these links contain the date of the document, publication, and word count.

Rather than presenting the duplicate listing in the results lists, some embodiments display a container icon, such as a duplicates folder, in association with the primary document identifier. The duplicates folder is a user-selectable icon that when selected opens a window to display a listing of the duplicate documents (or corresponding URLs) along with bibliographic information.

From presentation of search results at block 260, execution advances to block 270, which entails outputting one or more select documents from the search results. In the exemplary embodiment, this output entails printing, emailing, or saving one or more of the identified documents to memory associated with a client access device, such as access device 130, in response to a user selection. To facilitate user control or direction of this output, the exemplary embodiment presents the user a graphical user interface, such as interface 600 in Figure 6, which is incorporated and accessible via interface 138 in Figure 1.

Interface 600 includes an output-destination region 610, a range-definition region 620, a content-definition region 630, and a request-submission region 640. Output-destination region 610 allows a user to specify printer, email, or memory destinations for one or more portions of the search results. (See Figure 7 and supporting text for sample email message.) Range-definition region 620 allows a user to identify the documents within the search results that are to be output to the destination(s) identified in region 610.

Content-definition region 630, which allows the user to identify specific portions of documents selected in range-definition region 620 portion for output, includes, among other things, a duplicate-output control feature 631. This feature allows a user to indicate and control whether to include any documents that are duplicates of the identified documents as part of the output. If the user has not modified her corresponding preference, such as preference setting 1241D in subscriber database 124 of Figure 1, the default is to exclude duplicate documents from the output, meaning that the duplicate-output control region initially indicates that duplicates are to be excluded from the output.

In the exemplary embodiment, selection or invocation of the include-duplicates feature not only causes output of duplicate documents for all options in the range definition region except for the selected documents or citations option, but also affects displays within interface 600. The "all documents" range option in region 620 includes a label indicating the number of primary documents in the search result, with the set of primary documents being the entire set of search results minus any duplicates. If the search results include 30 total documents and documents 1, 5 and 7 are duplicates, the label next to the

"all documents" option in region 620 ordinarily would indicate "28 documents." However, if the user invokes the duplicate-inclusion feature (manually or by default) the label would indicate both primary and duplicate documents, or pursuant to the example, "30 documents."

Figure 7 shows an exemplary HTML-formatted electronic message 700 which includes among other things a results list that identifies duplicate documents. Specifically, message 700, which can also be sent in response to automatic running of a query defined for repeated, periodic or event-driven execution, includes a header 710, a query region 720, and a result listing region 730.

Header 710 includes from, sent, to, and subject regions, which respectively identify the sender, time-sent, recipient, and subject of the email. Query region 720 includes a database-identification field 722 and a query field 724. Database identification field 722 identifies the database(s) that were searched, and query field 724 lists text of a query. In the exemplary embodiment, the particular query takes the form of a Boolean or natural-language query.

Results listing region 730 includes one or more document citations or hyperlinks, such as identifiers 732 and 734, which are selectable to invoke retrieval and display of all or a portion of each corresponding document. In some embodiments, selection of one of the hyperlinks immediately causes retrieval of the corresponding document in a browser window. In others the selection results in display of a sign-on screen in the browser window, prompting the user to enter appropriate login and/or client-matter-identification data prior to document display.) Associated with document identifier 732 is a set 7322 of one or more selectable duplicate-document identifiers or links. In the exemplary embodiment, labels for these links contain respective publication dates and word counts.

Exemplary Options-Control Interface

Figure 8 shows an exemplary options-control interface 800, which functions as a portion of interface 138 in Figure 1 and allows users to set values for preferences in subscriber database 123, such as those related to duplicate

processing and/or presentation. In the exemplary embodiment, interface 800 includes an identify-duplicates control feature 810, duplicate-inclusion-or-exclusion control features 820, primary-duplicate-selection features 830, and a save-command feature 840.

Identify-duplicates control feature 810, a check box in this embodiment, enables users to set the default for whether duplicate processing is performed on eligible search results.

Duplicate-inclusion-or-exclusion control elements 820 include control features 821 and 822. In this embodiment, features 821 and 822 take the form of radio buttons and respectively allow users to select whether duplicates are to be included in the displayed results list or excluded from the displayed results list.

Primary-duplicate-selection features 830 include features 831 and 832, which also take the form of radio buttons and which allow users to specify which document in a set of duplicate documents to display as a primary document in a results list. In this embodiment, the user can select to have the longest document or the most recent or most relevant document as the primary document. If duplicates are to be excluded from the search results, this option governs which document of a set of duplicates is displayed in the results list, whereas if documents are to be included within the list, this option governs which document is to be displayed first in the list.

Save command feature 840 enables a user to save changes made via control features 810, 820, 830 to subscriber database 124 for use during the remainder of a current search session and during future search sessions.

Use of Temporal and Length Bins

Some embodiments integrate the temporal and length comparison of blocks 252B and 253B (in Figure 2) by defining sets or bins of potentially duplicate documents. For example, some embodiments retrieve a set of corresponding signature data structures as defined in flow chart 210B from signature database 124 and sort them in reverse chronological order based on their respective temporal components.

After this temporal sorting, these embodiments define one or more temporal sets or bins, by "moving" a fixed-temporal window down the sorted

document list. The first temporal bin includes the first sorted document and all documents having a temporal value within a time period, such as 30 days, of the first sorted document. The second temporal bin includes the second sorted document and all documents having a temporal value within 30 days of the temporal value for the second sorted document. Additional bins are defined similarly, moving down the list of sorted documents. (Some embodiments define mutually exclusive sets or bins of documents.)

Once these temporal sets or bins are defined, these embodiments define one or more length-based bins or sets of signatures within each of the defined temporal bins. This entails sorting the signatures in each of the temporal bins in descending order of the length components in their corresponding signatures and then "moving" a length window down the sorted list to define one or more length bins or sets. The first length bin within the first temporal bin includes the first signature in the length-ranked list in that bin (that is, the longest document in the temporal bin) and all documents within the first temporal bin that are no more than 20% shorter than the length of the longest document.

In other words, each subsequent signature has a length value at least 80% of that of each other signature in the first length bin. The second length bin includes the second longest document in the first temporal bin and all documents that are no more than 20% shorter than it. Subsequent length bins are defined similarly until all the documents in each temporal bin are assigned to a length bin. Thus, two documents are members of the same length bin if their temporal features (for example, publication dates) are within the same 30-day window and the length of the shorter of the documents is no less than 80% the length of the longer document. Further length bins for the first temporal bin and any other temporal bin are defined similarly.

To illustrate this binning, Figure 9 shows a diagram 900. Diagram 900 includes a reverse chronologically sorted list 910 of document identifiers or document signatures D1-D20, which has been organized as three temporal bins or sets TB1, TB2, and TB3, with each bin identifying or corresponding to a set of documents that are published within 30 days of each other. Temporal bin TB1 includes document identifiers or document signatures D1-D7. Sorting the contents of temporal bin TB1 based on the length components of its document

signatures yields a length sorted list 920. List 920 is shown organized as two length bins LB1 and LB2, with each bin identifying or corresponding to a set of documents that are no more than 20% shorter than the length of the longest document in the bin. Once all the length bins for each temporal bin are defined, all unique pairs of documents in each length bin are compared pursuant to the content-comparison process outlined for block 254B (in Figure 2.)

Some embodiments omit actual defining of the length bins, instead comparing the length of each document to each other document in the current temporal bin and performing the content comparison only for those pairs of documents that have lengths within ±20% of each other. In effect, these embodiments define length bins in a virtual manner.

Conclusion

In furtherance of the art, the present inventors have not only recognized a need to effectively address how information-retrieval systems handle the existence of duplicate documents in their document collections, but also presented herein systems, methods, and software that facilitate the identification and/or grouping of duplicate documents in search results, in accordance with user preferences. This identification and grouping ultimately streamline the process of users accessing and reviewing search results that contain duplicate documents.

The embodiments described above are intended only to illustrate and teach one or more ways of making and using the present invention, not to restrict its breadth or scope. The actual scope of the invention, which embraces all ways of practicing or implementing the teachings of the invention, is defined only by one or more issued patent claims and their equivalents.

Claims

1. An information-retrieval system comprising:

one or more databases; and

one or more servers for facilitating client access to the databases over a network, with each server including:

query-definition means for facilitating user submission of a query and user selection of an option related to identification of search-result documents that include content duplicative of one or more other search-result documents; and

duplicate-determination means for determining whether one or more of the search-result documents include content duplicative of content in one or more other search-result documents, with the duplicate-determination means including:

means for comparing first and second feature vectors for respective first and second documents, with each feature vector including a plurality of equal-length binary representations of features selected from the respective document, and with each binary representation based on sequential position of a corresponding one of the features within an inverse-document-frequency (idf) table for one or more of the databases; and

means for determining whether the first and second documents are duplicates based on results of comparing the first and second feature vectors;

means for controlling display of search-result documents based on the selected option, with at least one of the displayed results indicated as including content duplicative of content in one or more other documents within the results.

2. The system of claim 1, wherein the server further comprises means for comparing first and second lengths of the respective first and second

documents, and the means for comparing first and second feature vectors for respective first and second documents compares the feature vectors only in response to the first and second lengths having a predetermined relationship.

3. The system of claim 1, wherein each feature vector includes at least 30 terms selected from its respective document, and wherein the duplicate-determination means determines that documents are duplicates when at least 80 percent of the terms in the first and second feature vectors match.

4. The system of claim 1, wherein the idf table is sorted in descending order of idf value.

5. The system of claim 1, wherein each means comprises one or more sets of machine-readable instructions.

6. A method of operating an information-retrieval system, comprising:
comparing first and second feature vectors for the respective first and second documents, with each feature vector including a plurality of binary representations of features selected from the respective document, and with each binary representation based on sequential position of a corresponding one of the features within an inverse-document-frequency (idf) table; and
determining whether the first and second documents are duplicates based on results of comparing the first and second feature vectors.

7. The method of claim 6, further comprising:
comparing first and second lengths of the respective first and second documents identified in response to a user query;
wherein the comparison of the first and second documents occurs in response to the comparison of the first and second lengths indicating that the first and second lengths have a predetermined relationship.

8. The method of claim 6, further comprising comparing first and second temporal values associated respectively with the first and second documents.

9. The method of claim 6, wherein the comparison of first and second features vectors occurs in real-time response to a query submitted over the Internet to the system.

10. The method of claim 7, wherein the determination of whether the first and second documents are duplicates is affirmative if and only if the first feature vector has at least a threshold number of features in common with the second feature vector.

11. The method of claim 6, wherein the binary representations are of equal length and each feature is selected from the respective document based on relative magnitude its corresponding idf value in the idf table.

12. The method of claim 11, wherein the idf table is sorted in descending order of idf value and excludes features having an idf value greater than 0.8.

13. A machine-readable medium comprising instructions for:
comparing first and second feature vectors for the respective first and second documents, with each feature vector including a plurality of equal-length binary representations of features selected from the respective document, and with each binary representation based on position of a corresponding one of the features within an inverse-document-frequency (idf) table; and
determining whether the first and second documents are duplicates based on results of comparing the first and second feature vectors.

14. The medium of claim 13, further comprising instructions for:

comparing first and second lengths of the respective first and second documents identified in response to a user query;

wherein the comparison of the first and second documents occurs in response to the comparison of the first and second lengths indicating that the first and second lengths have a predetermined relationship.

15. The medium of claim 13, further comprising instructions for comparing first and second temporal values associated respectively with the first and second documents.

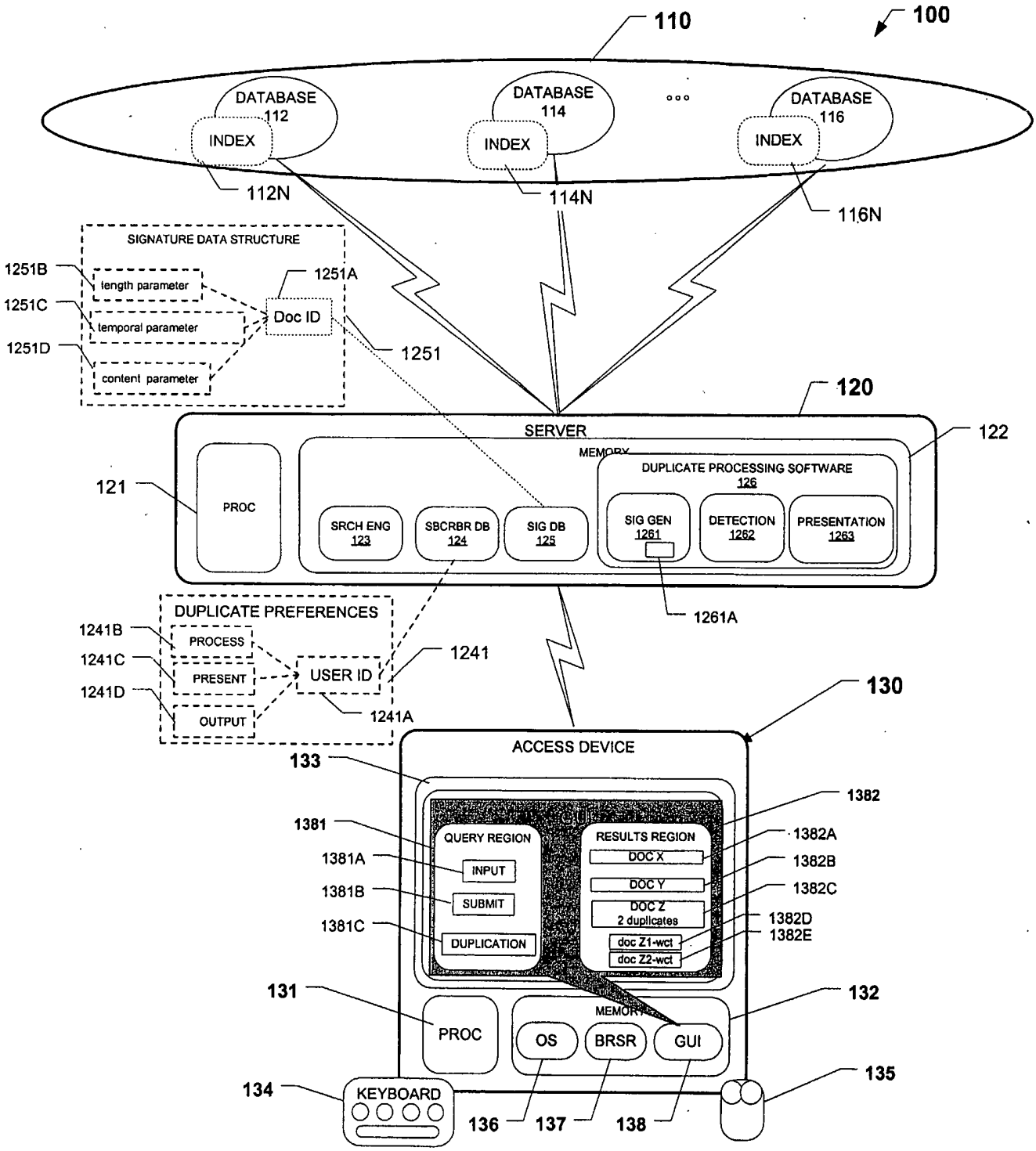

110
100
DATABASE 112
INDEX
DATABASE 114
INDEX
DATABASE 116
INDEX
112N
114N
116N
SIGNATURE DATA STRUCTURE
1251B
length parameter
1251A
1251C
temporal parameter
Doc ID
1251D
content parameter
1251
120
SERVER
122
MEMORY
121
PROC
DUPLICATE PROCESSING SOFTWARE 126
SRCH ENG 123
SBCRBR DB 124
SIG DB 125
SIG GEN 1261
DETECTION 1262
PRESENTATION 1263
1261A
DUPLICATE PREFERENCES
1241B
PROCESS
1241C
PRESENT
USER ID
1241
1241D
OUTPUT
1241A
130
ACCESS DEVICE
133
1382
1381
GUI
QUERY REGION
RESULTS REGION
1382A
1381A
INPUT
DOC X
1382B
1381B
SUBMIT
DOC Y
1382C
DOC Z
2 duplicates
1381C
1382D
DUPLICATION
1382E
doc Z1-wct
doc Z2-wct
131
132
MEMORY
PROC
OS
BRSR
GUI
135
134
KEYBOARD
136
137
138

FIGURE 1

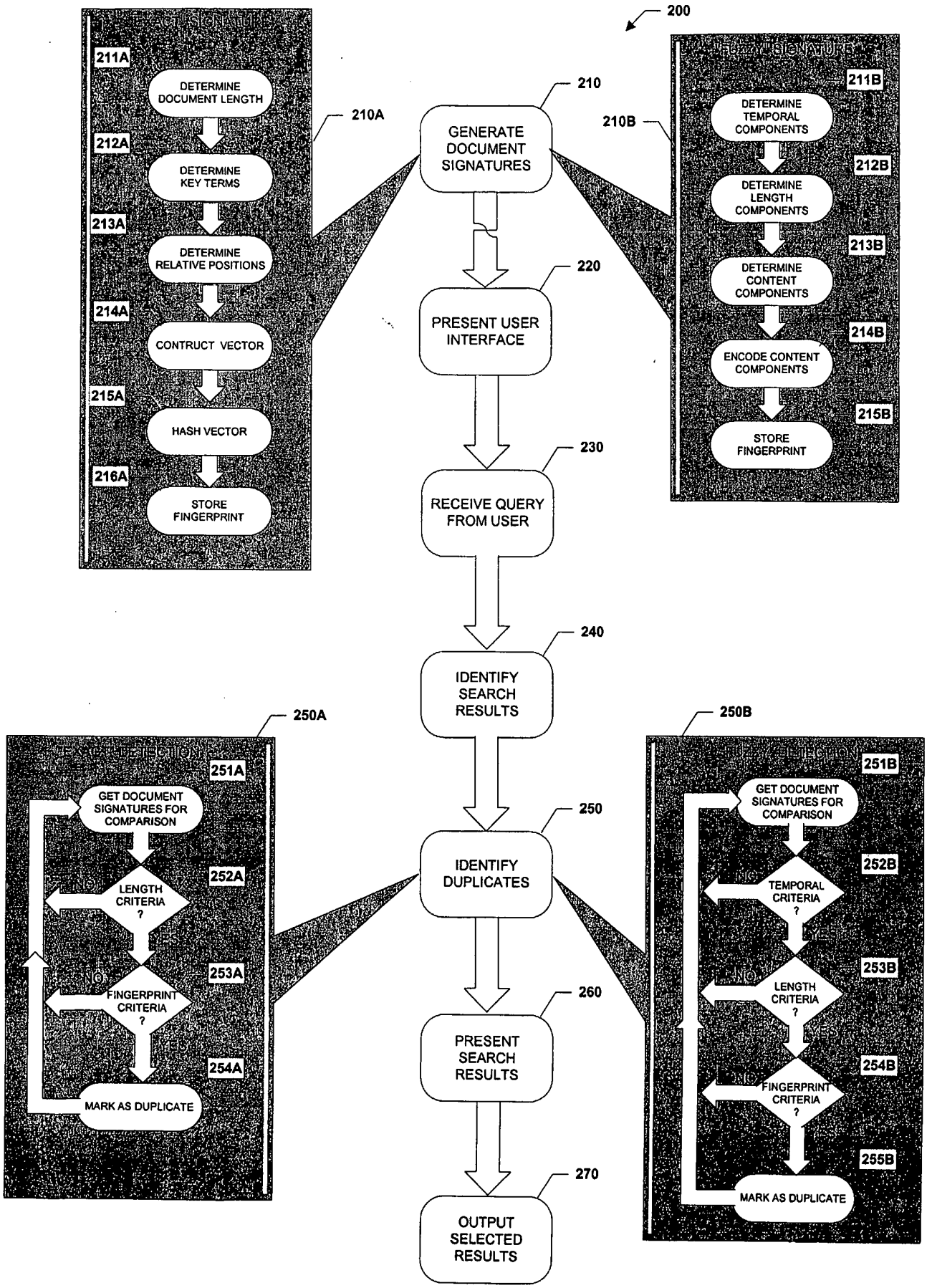

200
210
GENERATE DOCUMENT SIGNATURES
210A
211A
DETERMINE DOCUMENT LENGTH
212A
DETERMINE KEY TERMS
213A
DETERMINE RELATIVE POSITIONS
214A
CONTRUCT VECTOR
215A
HASH VECTOR
216A
STORE FINGERPRINT
210B
211B
DETERMINE TEMPORAL COMPONENTS
212B
DETERMINE LENGTH COMPONENTS
213B
DETERMINE CONTENT COMPONENTS
214B
ENCODE CONTENT COMPONENTS
215B
STORE FINGERPRINT
220
PRESENT USER INTERFACE
230
RECEIVE QUERY FROM USER
240
IDENTIFY SEARCH RESULTS
250
IDENTIFY DUPLICATES
250A
251A
GET DOCUMENT SIGNATURES FOR COMPARISON
252A
LENGTH CRITERIA ?
253A
FINGERPRINT CRITERIA ?
254A
MARK AS DUPLICATE
250B
251B
GET DOCUMENT SIGNATURES FOR COMPARISON
252B
TEMPORAL CRITERIA ?
253B
LENGTH CRITERIA ?
254B
FINGERPRINT CRITERIA ?
255B
MARK AS DUPLICATE
260
PRESENT SEARCH RESULTS
270
OUTPUT SELECTED RESULTS

FIGURE 2

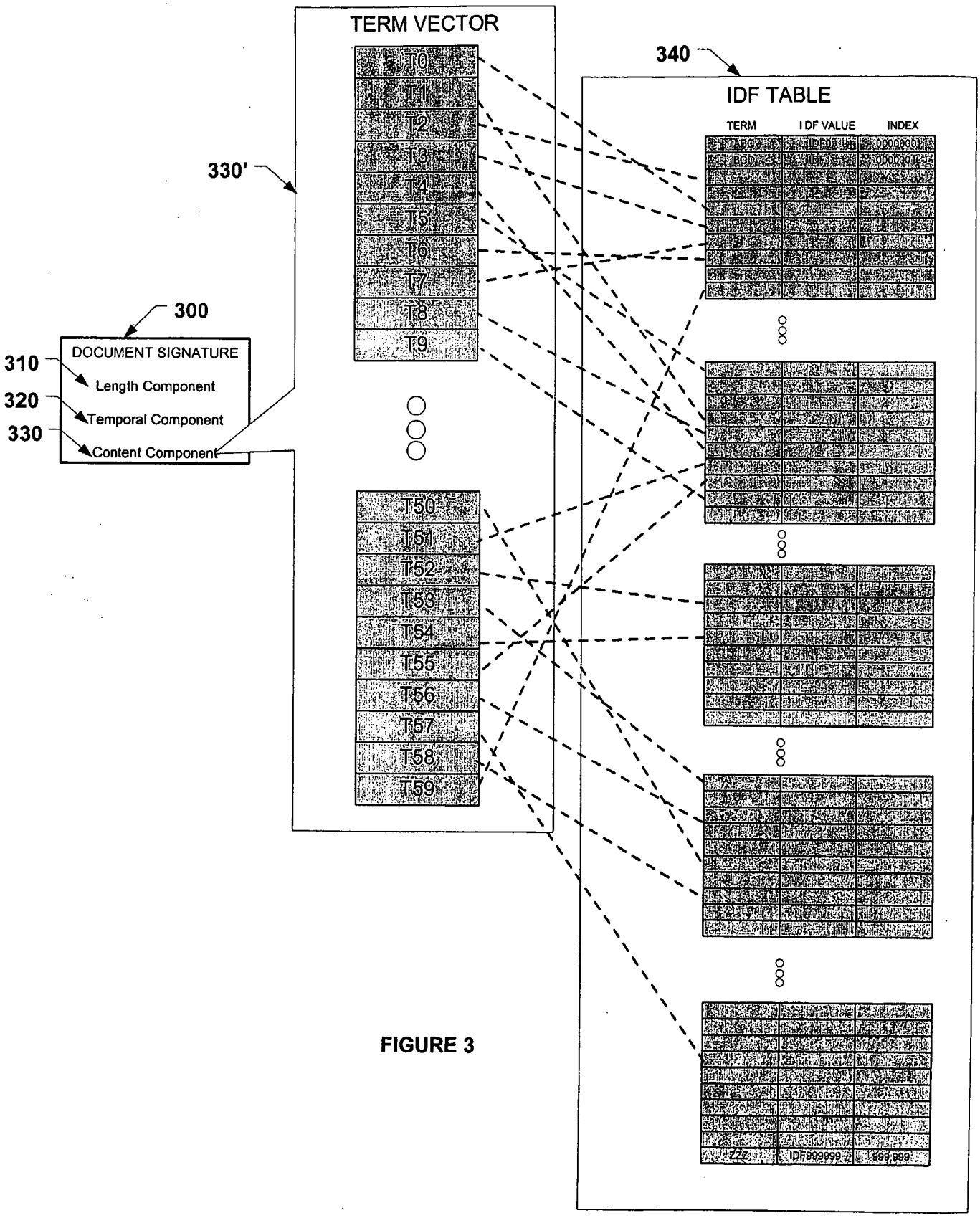

TERM VECTOR
T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T50
T51
T52
T53
T54
T55
T56
T57
T58
T59
330'
300
DOCUMENT SIGNATURE
310
Length Component
320
Temporal Component
330
Content Component
340
IDF TABLE
TERM
IDF VALUE
INDEX

FIGURE 3

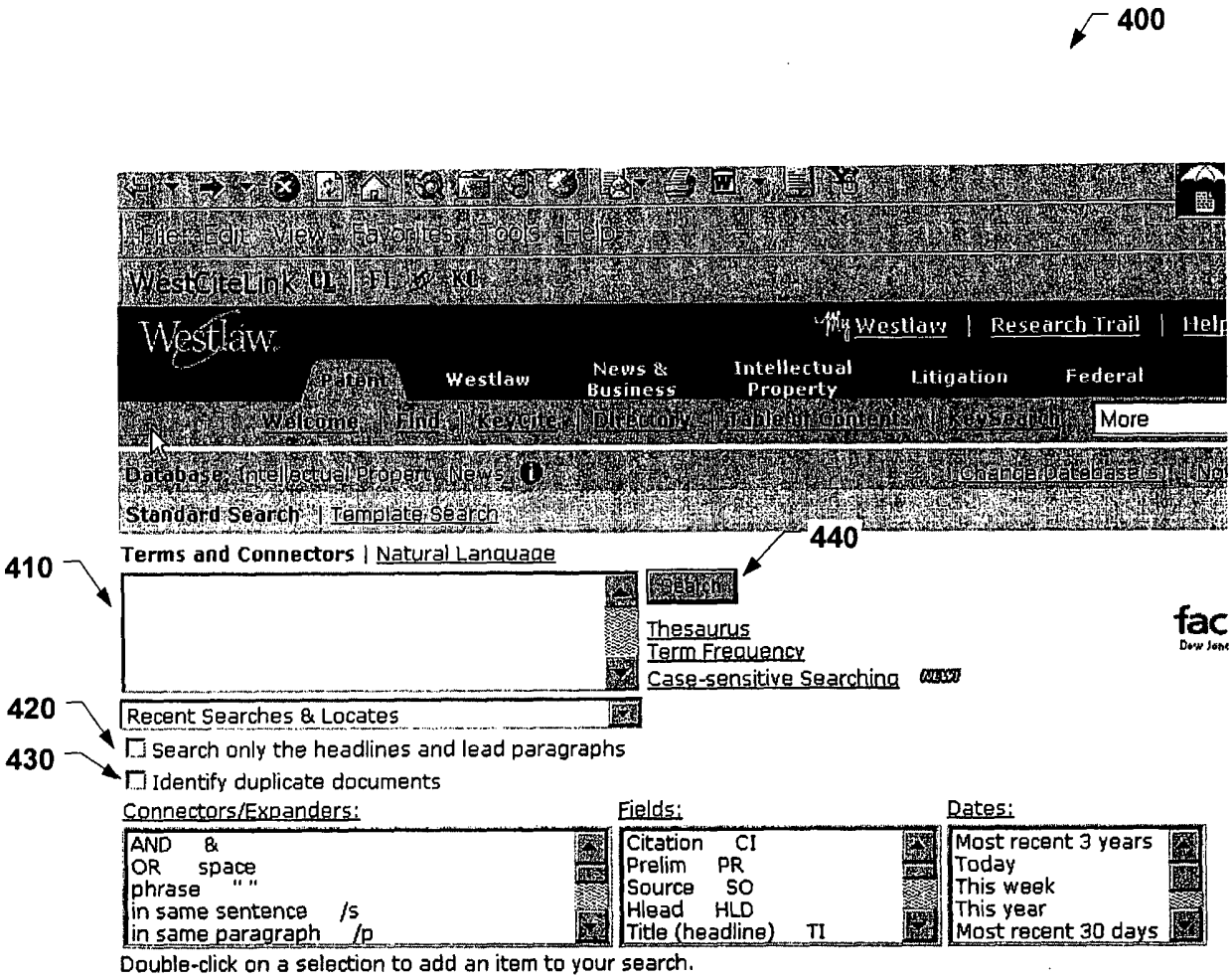

400
410
420
430
440
Westlaw
Research Trail
Patent
Westlaw
News & Business
Intellectual Property
Litigation
Federal
Standard Search | Template Search
Terms and Connectors | Natural Language
Search
Thesaurus
Term Frequency
Case-sensitive Searching
Recent Searches & Locates
Search only the headlines and lead paragraphs
Identify duplicate documents
Connectors/Expanders:
AND &
OR space
phrase " "
in same sentence /s
in same paragraph /p
Fields:
Citation CI
Prelim PR
Source SO
Hlead HLD
Title (headline) TI
Dates:
Most recent 3 years
Today
This week
This year
Most recent 30 days
Double-click on a selection to add an item to your search.

FIGURE 4

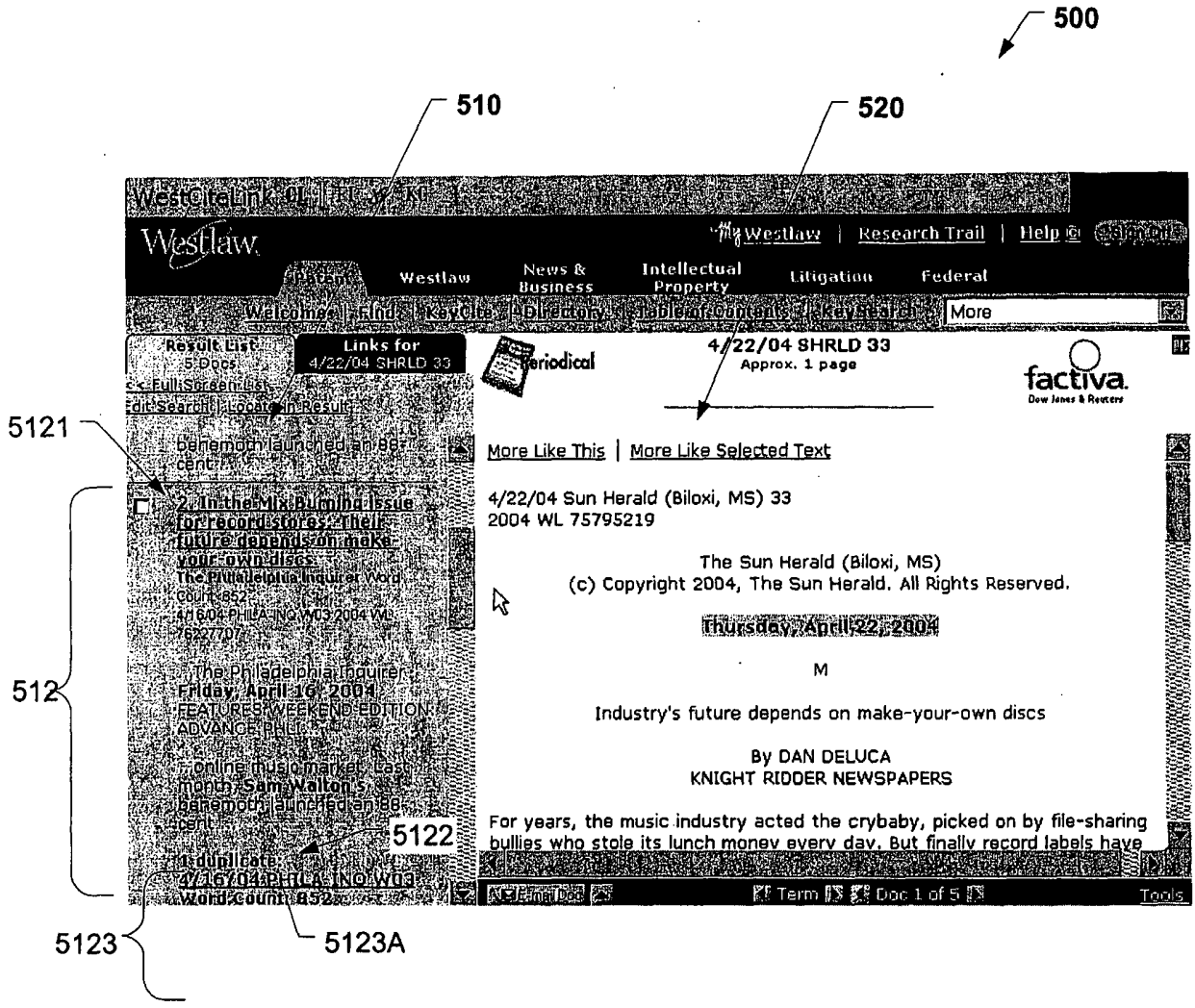

500
510
520
5121
512
5122
5123
5123A
Westlaw
Research Trail
Help
Westlaw
News & Business
Intellectual Property
Litigation
Federal
More
Result List
5 Docs.
Links for 4/22/04 SHRLD 33
4/22/04 SHRLD 33
Approx. 1 page
factiva
Dow Jones & Reuters
More Like This | More Like Selected Text
4/22/04 Sun Herald (Biloxi, MS) 33
2004 WL 75795219
The Sun Herald (Biloxi, MS)
(c) Copyright 2004, The Sun Herald. All Rights Reserved.
Thursday, April 22, 2004
M
Industry's future depends on make-your-own discs
By DAN DELUCA
KNIGHT RIDDER NEWSPAPERS
For years, the music industry acted the crybaby, picked on by file-sharing
Term
Doc 1 of 5
Tools

FIGURE 5

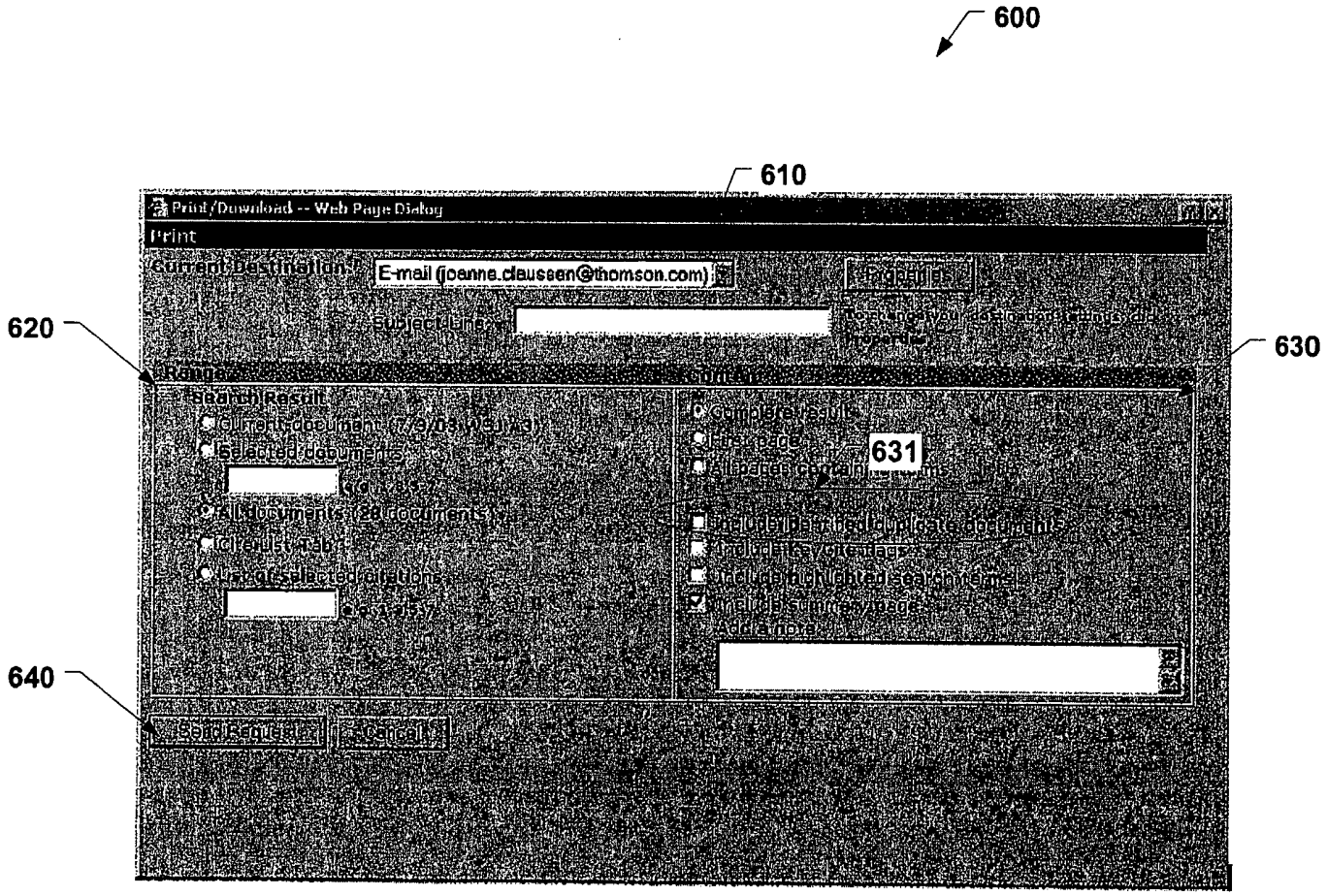

600
610
620
630
631
640
Print/Download -- Web Page Dialog
Print
Current Destination:
E-mail (joanne.clausen@thomson.com)
Properties
Subject Line:
Range
Search Result
Current document
Selected documents
All documents
Current Tab
Cancel

FIGURE 6

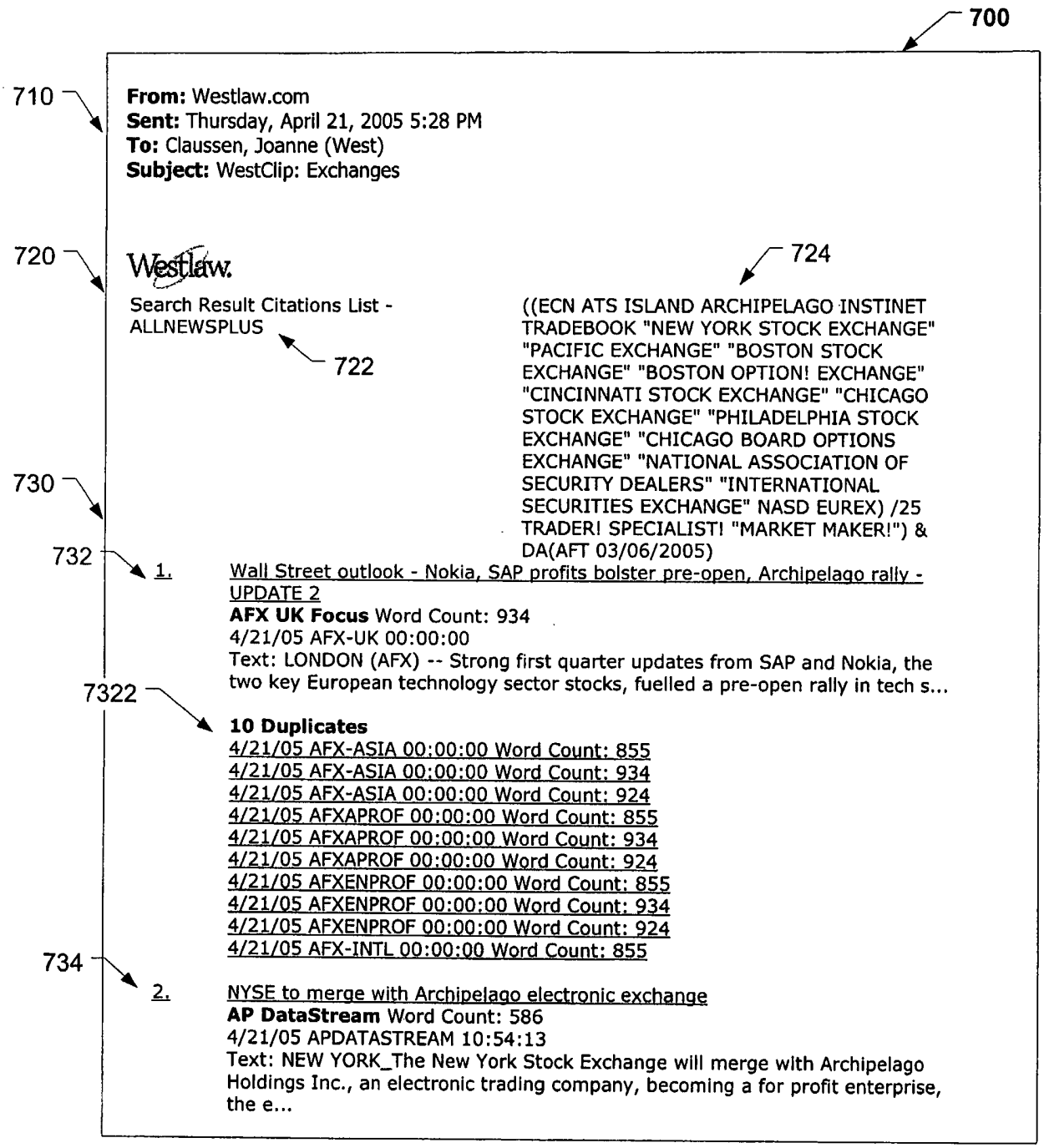

700
710
From: Westlaw.com
Sent: Thursday, April 21, 2005 5:28 PM
To: Claussen, Joanne (West)
Subject: WestClip: Exchanges
720
Westlaw.
724
Search Result Citations List - ALLNEWSPLUS
722
((ECN ATS ISLAND ARCHIPELAGO INSTINET TRADEBOOK "NEW YORK STOCK EXCHANGE" "PACIFIC EXCHANGE" "BOSTON STOCK EXCHANGE" "BOSTON OPTION! EXCHANGE" "CINCINNATI STOCK EXCHANGE" "CHICAGO STOCK EXCHANGE" "PHILADELPHIA STOCK EXCHANGE" "CHICAGO BOARD OPTIONS EXCHANGE" "NATIONAL ASSOCIATION OF SECURITY DEALERS" "INTERNATIONAL SECURITIES EXCHANGE" NASD EUREX) /25 TRADER! SPECIALIST! "MARKET MAKER!") & DA(AFT 03/06/2005)
730
732
1. Wall Street outlook - Nokia, SAP profits bolster pre-open, Archipelago rally - UPDATE 2
AFX UK Focus Word Count: 934
4/21/05 AFX-UK 00:00:00
Text: LONDON (AFX) -- Strong first quarter updates from SAP and Nokia, the two key European technology sector stocks, fuelled a pre-open rally in tech s...
7322
10 Duplicates
4/21/05 AFX-ASIA 00:00:00 Word Count: 855
4/21/05 AFX-ASIA 00:00:00 Word Count: 934
4/21/05 AFX-ASIA 00:00:00 Word Count: 924
4/21/05 AFXAPROF 00:00:00 Word Count: 855
4/21/05 AFXAPROF 00:00:00 Word Count: 934
4/21/05 AFXAPROF 00:00:00 Word Count: 924
4/21/05 AFXENPROF 00:00:00 Word Count: 855
4/21/05 AFXENPROF 00:00:00 Word Count: 934
4/21/05 AFXENPROF 00:00:00 Word Count: 924
4/21/05 AFX-INTL 00:00:00 Word Count: 855
734
2. NYSE to merge with Archipelago electronic exchange
AP DataStream Word Count: 586
4/21/05 APDATASTREAM 10:54:13
Text: NEW YORK_The New York Stock Exchange will merge with Archipelago Holdings Inc., an electronic trading company, becoming a for profit enterprise, the e...

FIGURE 7

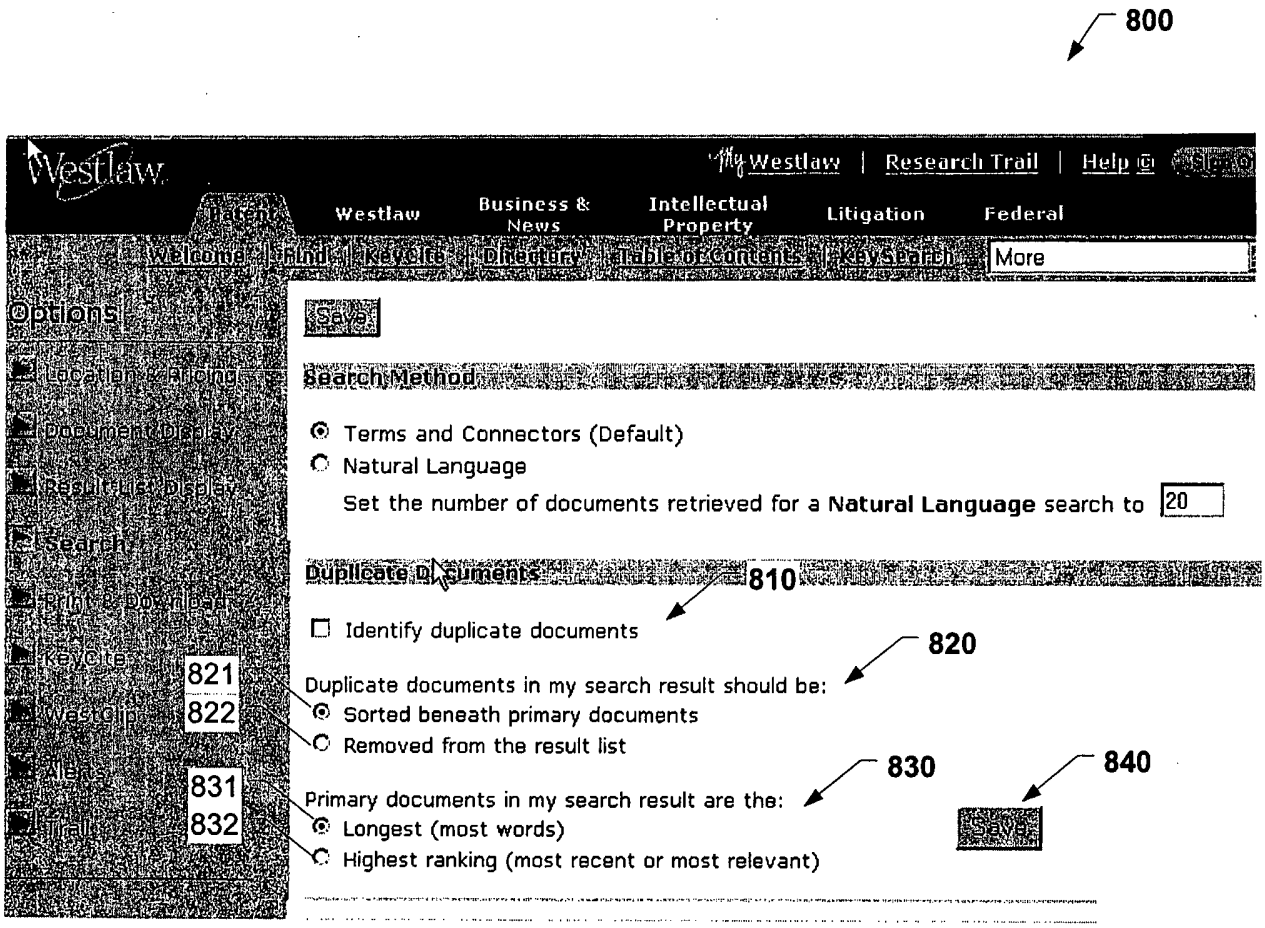

800
Westlaw
My Westlaw | Research Trail | Help
Patent
Westlaw
Business & News
Intellectual Property
Litigation
Federal
Welcome | Find | KeyCite | Directory | Table of Contents | KeySearch | More
Options
Location & Filtering
Document Display
Result List Display
Search
Print & Download
KeyCite
821
WestClip
822
Alerts
831
Trails
832
Save
Search Method
Terms and Connectors (Default)
Natural Language
Set the number of documents retrieved for a Natural Language search to 20
Duplicate Documents
810
Identify duplicate documents
820
Duplicate documents in my search result should be:
Sorted beneath primary documents
Removed from the result list
830
Primary documents in my search result are the:
Longest (most words)
Highest ranking (most recent or most relevant)
840
Save

FIGURE 8

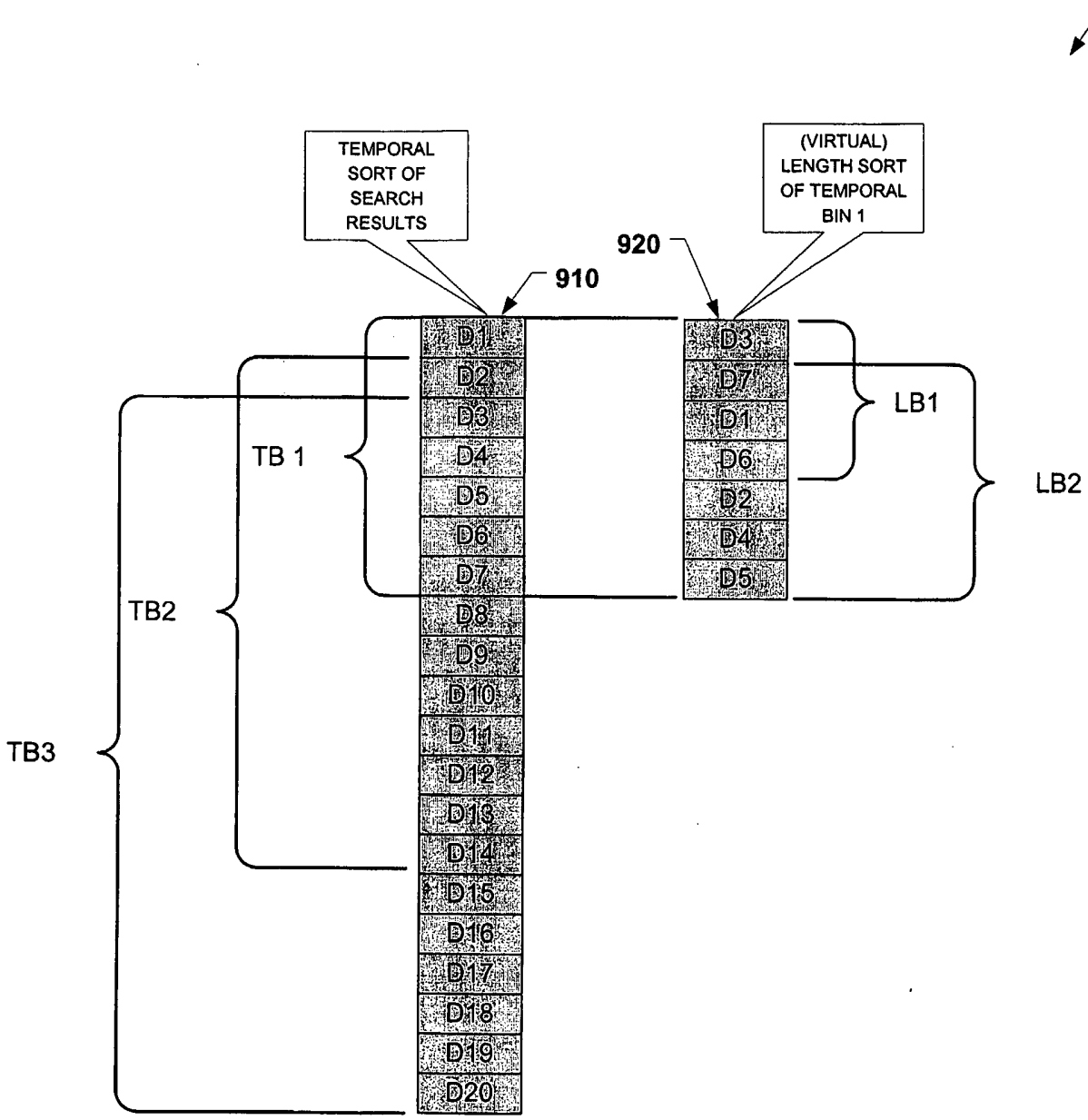

900
TEMPORAL SORT OF SEARCH RESULTS
(VIRTUAL) LENGTH SORT OF TEMPORAL BIN 1
920
910
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D3
D7
D1
D6
D2
D4
D5
LB1
LB2
TB 1
TB2
TB3

FIGURE 9

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/030024

A. CLASSIFICATION OF SUBJECT MATTER
G06F17/30

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)
G06F G06K G06N

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 practical, search terms used)

EPO-Internal, WPI Data, INSPEC, PAJ, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/172063 A1 (GUTTA SRINIVAS ET AL) 11 September 2003 (2003-09-11) page 1, paragraph 8 - page 5, paragraph 54 figures 1-10	1-15
A	FRIEDER O ET AL: "On scalable information retrieval systems" NETWORK COMPUTING AND APPLICATIONS, 2003. NCA 2003. SECOND IEEE INTERNATIONAL SYMPOSIUM ON 16-18 APRIL 2003, PISCATAWAY, NJ, USA,IEEE, 16 April 2003 (2003-04-16), pages 241-245, XP010640257 ISBN: 0-7695-1938-5 the whole document	1-15

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"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)

"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

"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

"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

"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.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 November 2005	16/12/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Warry, L

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/030024

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 513 652 A (SIEMENS AKTIENGESELLSCHAFT) 19 November 1992 (1992-11-19) page 4, line 18 - page 10, line 26	1-15
A	TONELLA P ET AL: "Using keyword extraction for web site clustering" WEB SITE EVOLUTION, 2003. THEME: ARCHITECTURE. PROCEEDINGS. FIFTH IEEE INTERNATIONAL WORKSHOP ON 22 SEPT. 2003, PISCATAWAY, NJ, USA,IEEE, 2003, pages 41-48, XP010659541 ISBN: 0-7695-2016-2 the whole document	1-15
A	US 6 654 739 B1 (APTE CHIDANAND ET AL) 25 November 2003 (2003-11-25) column 2, line 65 - column 9, line 20	1-15
A	HOPPENBROUWERS J ET AL: "Invading the fortress: how to besiege reinforced information bunkers" ADVANCES IN DIGITAL LIBRARIES, 2000. PROCEEDINGS. IEEE WASHINGTON, DC, USA 22-24 MAY 2000, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 22 May 2000 (2000-05-22), pages 27-35, XP010501102 ISBN: 0-7695-0659-3 the whole document	1-15
A	MISHRA R K ET AL: "KhojYantra: an integrated metasearch engine with classification, clustering and ranking" DATABASE ENGINEERING AND APPLICATIONS SYMPOSIUM, 2000 INTERNATIONAL SEPT. 18-20, 2000, PISCATAWAY, NJ, USA,IEEE, 18 September 2000 (2000-09-18), pages 122-131, XP010519020 ISBN: 0-7695-0789-1 the whole document	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US2005/030024

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003172063 A1	11-09-2003	AU 2003206064 A1	16-09-2003
		CN 1639712 A	13-07-2005
		EP 1485823 A2	15-12-2004
		WO 03075181 A2	12-09-2003
		JP 2005519396 T	30-06-2005
EP 0513652 A	19-11-1992	US 5461698 A	24-10-1995
US 6654739 B1	25-11-2003	GB 2365576 A	20-02-2002