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

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 June 2000 (15.06.2000)

PCT

(10) International Publication Number
WO 00/34897 A1

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

(21) International Application Number: PCT/US99/28870

(22) International Filing Date: 6 December 1999 (06.12.1999)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/111,212 7 December 1998 (07.12.1998) US

(71) Applicant: **BLOODHOUND SOFTWARE, INC.** [US/US]; 100 Capitola Drive, Suite 304, Durham, NC 27713 (US).

(72) Inventors: **WHIPPLE, David**; 247 Morley Drive, Dianella, 6059 (AU). **CARSANARO, Joseph**; 1289 N. Fordham Boulevard, Suite A-110, Chapel Hill, NC 27514 (US). **YOUNG, Ken**; P.O. Box 219, Petrolia, CA 95558 (US).

(74) Agents: **KIRSCH, Gregory, J.** et al.; Needle & Rosenberg, P.C., Suite 1200, The Candler Building, 127 Peachtree Street, N.E., Atlanta, GA 30303-1811 (US).

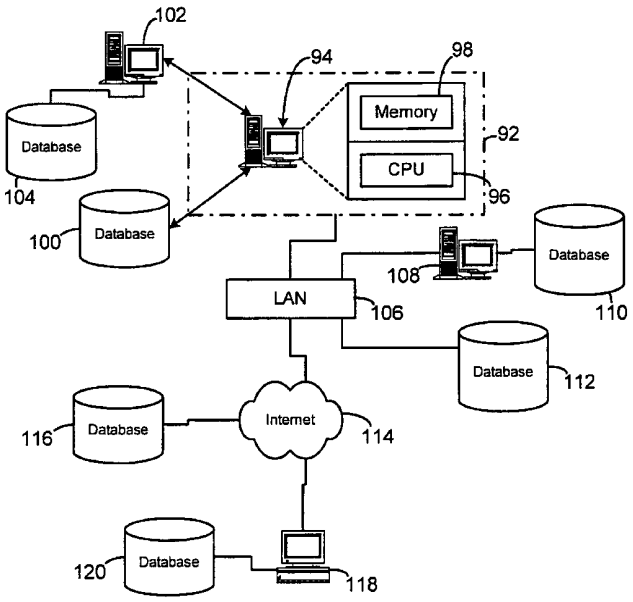
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL,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

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, 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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *With amended claims.*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR FINDING NEAR MATCHES AMONG RECORDS IN DATABASES



(57) Abstract: The present invention is a system and method for finding near matches among records in databases (104, 100, 116, 120, 112, 110) and data stores in computer systems. The system identifies near matches between records in the data store and a selected record having an associated coordinate set. The processor (96) creates identifiers which are associated with each record in the data store, maps each identifiers in a discriminant space associated with each record, and retrieves all records from the data store having associated coordinate set within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

WO 00/34897 A1

(48) Date of publication of this corrected version:
7 June 2001

(15) Information about Correction:
see PCT Gazette No. 23/2001 of 7 June 2001, Section II

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.

SYSTEM AND METHOD FOR FINDING NEAR MATCHES AMONG RECORDS IN DATABASES

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/111,212, filed on December 7, 1998.

BACKGROUND OF INVENTION

1. *Field of the Invention*

This invention generally relates to computer systems and the locating of records in databases and datastores. More specifically, this invention relates to a system and method for identifying near matches among records in a datastore based upon discriminant analysis.

2. *Description of the Related Art*

The invention addresses the problem of having duplicate and near duplicate records in database files, data marts, data warehouses or any data file. The duplication of information is difficult to find and can lead to wasted time and money. Processing duplicate claims, expense payments or other duplicate records can lead to cost over runs, customer service problems, inefficient processing time, manual intervention into automated systems, and wasted disk storage on computer systems. Unsynchronized data over multiple environments can lead to data duplicates, data replication and other data management problems. Furthermore, the inability to locate a near match in Internet searches can lead to lost sales opportunities, poor customer service problems and lost revenue.

Existing systems use standard procedures for indexing records and locating similar ("sims") or duplicate records. These records may then be removed, purged, flagged for future reference, extracted from the data set for viewing, or extracted for

use in additional statistical analysis. These procedures incorporate three basic steps: (1) creating a "keystring" for each record, where a keystring is a character string comprised of all or portions of some or all of the fields in a record; (2) sorting the keystrings, which is termed "indexing"; and (3) scanning the sorted list of keystrings for sims.

Conventionally, scanning the sorted list (step 3) is a single pass through the sorted list and comparing each successive pair of sorted keystrings to determine some measure of their similarity. Pairs of keystrings that are found to be similar within some pre-defined measure of similarity are flagged or one of the records is removed. Under this method, only mismatches in the least significant (right-most) character positions will be found.

For example, consider the following (sorted) keystrings in the following table:

	1	2	3	4	5	6
Keystring 1)	C	L	A	R	K	E
Keystring 2)	C	L	A	R	Y	S
Keystring 3)	C	L	E	R	K	E
Keystring 4)	D	L	A	R	K	E

"Clarke" and "Clarys" are mismatched in positions 5 and 6. "Clarys" and "Clerke" are likewise mismatched in positions 5 and 6 as well as in position 3. However, "Clarke" and "Clerke" are mismatched in only position 3. A sequential pass through the list of keystrings looking only at adjacent pairs of keystrings would miss this sim.

"Dlarke" and "Clarke" are also mismatched in only one position, namely position 1, and yet they are even further apart in the list. This ability to locate sims only in the right-most character positions is characteristic of a "linear indexing" scheme as it known in the art.

Accordingly, a system and method for sim identification that can perform a more accurate matching of the records in a data store would be advantageous. Thus, it

is to the provision of such an improved system and method that the present invention is primarily directed.

SUMMARY OF THE INVENTION

The present invention is a system and method for finding near-matches among records in one or more databases. In one embodiment, the system is for identifying near matches between records in a data store and a selected record having an associated coordinate set, and includes a data store for storing the records and a processor. The processor of the system performs the steps of creating one or more identifiers wherein each identifier is associated with a record in the data store, mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, and retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

In another aspect, the present invention provides a computer-readable storage device containing instructions that upon execution cause a processor to identify near matches between records in a data store and a selected record having an associated coordinate set. The device preferably performs the steps of creating one or more identifiers wherein each identifier is associated with a record in the data store, mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, and retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

The present inventive system accordingly provides a method for identifying near matches between records in a data store and a selected record that has an associated coordinate set. The method includes steps of creating one or more identifiers

wherein each identifier is associated with a record in the data store, mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, and retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record. The identifier associated with each record in the data store preferably comprises one or more characters.

The method preferably further includes the step of determining a set of records from the retrieved records that match the selected record. The step of determining a set of records from the retrieved records that match the selected record is preferably screening the coordinate sets associated with the retrieved records for sets within a second predetermined distance in the discriminant space from the coordinate set associated with the selected record. And then the method further includes the step of extracting, deleting, or otherwise modifying the determined set of records.

The method also preferably includes the step of acquiring a mapping template, and the step of acquiring is preferably creating or receiving a mapping template. The method then preferably includes the step of refining the acquired mapping template.

The method can include the step of selecting an identifier format for use in the step of creating one or more identifier, and then further include the step of acquiring one or more mapping templates. The step of selecting an identifier format is then preferably evaluating the acquired one or more mapping templates.

The method further preferably includes the steps of receiving the selected record, creating an identifier associated with the selected record, and mapping the identifier associated with the selected record into a coordinate set in the discriminant space associated with the selected record. Additionally, the method can further include

the step of retrieving the coordinate set associated with the selected record from the data store.

According to the present invention, near matches to a selected record that has an associated coordinate set are identified among records in a data store. An identifier, preferably a keystring, is created for each record in the database. Each such identifier is then mapped into a set of coordinates in discriminant space. Records in the data store that are near matches to the selected record are retrieved by collecting all records with an associated coordinate set within a predetermined distance in discriminant space from the coordinate set associated with the selected record.

In a further embodiment, identical records may be identified by further selecting records from retrieved records or by setting the predetermined distance at a threshold guaranteeing only identical records are retrieved. In yet another embodiment, the records retrieved as near matches and/or identical matches may be automatically deleted from the data store as duplicative or may be outputted to an appropriate output device for further automated or manual processing.

The system and method for finding near matches among records in databases accordingly has industrial applicability in that the invention can be installed and practiced on existing computer systems to increase searching efficiency. Moreover, the inventive system can be created during the manufacture of computer systems having database record searching as a significant component of the system functionality.

Furthermore, the present invention therefore has a commercial advantage in that it defines a multi-dimensional indexing scheme that performs efficient database searching. In a multi-dimensional indexing scheme, the likelihood of finding mismatched characters is not dependent on the position of the character in the keystring. Tests suggest this method is more efficient at finding sims, identifying from

50% to 100% more sims than conventional methods known in the art. Consequently, sims are more likely to be identified through use of the present invention than through use of in existing linear indexing systems.

The above and other objects and advantages of the present invention will become more readily apparent after review of the hereinafter set forth Brief Description of the Drawings, Detailed Description of the Invention, and Claims.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart illustrating the mapping process of the present invention, specifically illustrating the record, key, and coordinate templates of the process.

FIG. 2 is a flowchart illustrating the process of detecting duplicate entries, and specifically illustrating the template definition, key optimization, and record location processes.

FIG. 3 is an illustration of a defined template created from the QWERTY keyboard wherein the likelihood of errors is assumed to occur from typing errors due to key proximity.

FIG. 4a is a flowchart illustrating base template generation.

FIG. 4b is a flowchart illustrating the optimization of the base template generated from the process of Fig. 4a.

FIG. 5 is a flowchart illustrating the process of key creation.

FIG. 6 is a representative diagram of a host computer environment in connection with other computers and databases though a local area network (LAN) and through the Internet.

DETAILED DESCRIPTION OF THE INVENTION

Referring to the drawings, in which like numbers indicate like elements throughout the views, Fig. 1 is a flowchart illustrating the present inventive method for

finding near-matches among records in one or more databases. The present invention is a result of extending and enhancing the concepts of multiple discriminant analysis to locate each record in discriminant space (hereafter referred to as sim-space). The system selects a record 12 in one or more databases to dimension, shown at step 10, and record 12, and a key applied to the record 12, shown at step 14, to create a keystring 16. Each character position in the keystring, such as keystring 16, preferably represents a dimension of the record, although other aspects of this invention can use each character position to represent two or more dimensions. The system then determines the coordinates for each character in the in keystring through a selected template, shown at step 18, such as template 20.

Pre-determined "templates," such as template 20, define the actual coordinates of a record in sim-space. These templates provide a conformal mapping for each character in the keystring to a coordinate (or coordinates) in sim-space. Although shown here as ASCII characters, other characters that can be used in the present invention include the full English alphabet, numbers, words, special characters such as ñ (Spanish), ö (German), or ø (Norwegian), or can consist of entire non-Roman alphabets such as Greek, Russian, Arabic, Hebrew, Chinese, or Japanese characters.

Multidimensional indexing finds application in two main areas, firstly in searching an indexed list for a near match of a given record, and secondly in the detection of groups of similar records in lists. Finding a possible match for a given record in a previously indexed database can therefore be achieved by the following steps: generating the keystring for the record to be matched, determining the location in sim-space by applying template to the keystring, and searching the locations neighborhood, i.e. one or more databases or datastores, for sims by applying a nearest neighbor algorithm. Identifying sims within a previously indexed data base can be achieved by the following

steps preformed on the system: randomly selecting a keystring in sim-space, pulling a pre-determined number of nearest neighbors, and checking all possible pairs for sims within the set of neighbors.

With reference to Fig.2, there is illustrated a flowchart for the process of detecting duplicate entries, and the template definition, key optimization, and record location processes are specifically illustrated. The template defining process (TDP) defines templates are defined in such a manner so as to assign characters commonly substituted erroneously, to near-by coordinates. The system thus creates mapping templates, as shown at step 24.

The key optimization process (KOP) creates a keystring 16 for each record 12 in the database or datastore. The system creates a key to apply to the particular dataset under review in the program, shown at step 26, wherein the dataset can be one or more databases or data stores, and then the system creates a keystring 16 for each record in the dataset, shown at step 28. Each character position in the keystring is given an evaluation based on its ability to discriminate between records. The evaluation nominally lies between 0 and 1 (preferably expressed as a percentage) with 1 being the best discrimination (desired).

For example, if a particular position in a keystring always contained the same character, it would have no value in finding duplicate records. This could occur for example, with a mailing list database for California. The first digit of the zip code would always be "9" and would have no value in discriminating between records.

One simple method for assessing the discrimination for a given character position is to assign a "-1" to any coordinate lying below the median coordinate and a "+1" to any coordinate lying on or above the median coordinate. An evaluation of zero would result when the absolute value of the sum of the assigned -1's and +1's (a's) is

equal to the number of records (n). An evaluation of 100 would result when the sum of the assigned a's is equal to zero. Thus, the position evaluation (PE) is given by:

$$PE = (n - \sum a_i) / n.$$

Another method is to calculate the standard deviation of the coordinates for a given character position. However, such method has a disadvantage in that there is no easily defined "best" position evaluation. It should be noted that in the creation of a key, the order of the characters in the keystring is irrelevant for the present invention, unlike keys created for conventional indexing methods.

In the record location process (RLP) the system takes a template generated with a keystring 16 and locates matching records for the template, which represents near matches to the exact keystring generated from the record. The system maps the keystring into sim-space using the specified template, shown at step 30, and then the system examines the neighborhood, or database/datastore, for matches to the templated keystring, shown at step 32. Locating a record requires creating the keystring for that record and then determining the coordinates for that record using a template. Any number of processes may be utilized to make a key as are known in the art.

One preferred method of creating a mapping template that assigns characters which are commonly substituted erroneously is simply to "stretch" the QWERTY keyboard under the assumption that errors are commonly produced by typing a letter adjacent to the desired letter on the keyboard. This extrapolation of the QWERTY keyboard creates the template shown in Fig. 3.

Fig. 3 illustrates a template having a section 36 for the ASCII letter characters 38 with the template coordinates 40, and a section 42 for ASCII numbers 44 with corresponding template coordinates 46. Once the template has been used to identify

sims in a sample data base, the substitution error frequency can be directly determined for that type of data and data entry method.

Any number of methods can then be used to construct more optimal templates. Once the key is created, determining the coordinates of the record is a simple matter of substitution. For example, using the QWERTY template in Fig. 3, the keystring "CLARKE" would have coordinates of {11, 25, 3, 9, 22, 5}.

With reference to Figs. 4a and 4b, a "hill-climber" algorithm is employed to construct a more optimal template. A template evaluation function (TE) is defined as the sum of the error frequency (f) divided by the coordinate distance between each pair of characters (x_i, x_j) as, $TE = \sum f/(x_i - x_j)$, as shown in step 50. Characters are then randomly assigned coordinates and each set of assignments is evaluated, as shown in step 52. This step is repeated a pre-determined number of times and the set with the best (highest) score is saved and becomes the basis for the next step.

This set of coordinates is then "shuffled" by switching the coordinates for a randomly selected pair of characters that lie within a variable coordinate distance 'm.' A comparison is then made to determine if the switch produces a better evaluation, shown at decision 54. If the switch has made a better evaluation, the new set is saved and becomes the basis for continuing optimization, as shown at step 56, and a decision is made, decision 58, as to whether the process has been repeated a sufficient amount of times. If there is no improvement after a pre-determined number of switches, i.e. the score is not greater that the previous highest score, m is decreased by 1 and a decision is again made as to whether the process has been repeated the requisite amount of time, decision 58, and step 52 is repeated.

After an optimal template is produced, another variable is defined that represents a distance metric within the coordinate system, shown here as having an

initial value 'm', as shown at step 60. The value 'm' is initially chosen such that it completely includes all the set members in the current template configuration. Then a pair of characters in the co-ordinate space lying within 'm' units of each other are randomly selected, shown at step 62, and their coordinates are switched.

The Template evaluation function is then applied, shown at step 64, and the resulting error is compared to the current optimal templates, shown at decision 66. Should this template configuration yield a higher score, it is flagged as the new optimal template and set as the current template, shown at step 68. If the template does not yield a higher score, then a decision is made as to whether the template evaluation process has been repeated a predetermined number of times, shown at decision 70. If the evaluation process has not been repeated the requisite number of times, the pair of characters in the co-ordinate space lying within 'm' units of each other are again randomly selected, shown at step 62, and the process is repeated.

Once the optimal template has been selected, step 68, or the template evaluation process has been repeated the predetermined number of times, decision 70, then the distance m has 1 subtracted, shown at step 72, and a decision is made if m then equals 0, shown at decision 74. As 'm' is now encloses a smaller region in the coordinate system, there are less pairs within this new region for comparison. If the resultant score does not improve on the current optimal score, another pair of points is chosen and their coordinates are again swapped, step 62, for a maximum P retries. If m=0, the algorithm is complete and the optimal template has been determined.

Several templates may be used in a single purge operation. Each template should be designed to be independent of previously created templates. This can be accomplished by setting the error frequency of pairs of characters that have adjacent coordinates to zero and running the hill-climber algorithm again. Given a set of

templates, these templates can be used to evaluate the method used to create a keystring.

With reference to Fig. 5, the preferred process of key creation is illustrated. The system selects fields of the dataset, such as the local database, that provide a suitable level of discrimination, shown at step 78, and then all neglection of textual attributes such as vowels, numbers, punctuation and spaces are specified and preferably applied, shown at step 80. Then any logical field groupings are identified, shown at step 82, and any source field substitutions are specified, shown at step 84, should the source field be blank. Then all field weightings are specified, shown at step 86, and all composite fields that can be analyzed by subfield partitioning, such as addresses, are identified, shown at step 88. A decision is then made, decision 90, as to whether the key test results show a high level of discrimination. If the key test does show a high level of discrimination, then the process ends. If the key test does not show a high level of discrimination, then the process is begun anew, with new dataset fields again selected at step 78.

Fig. 6 illustrates a host computer environment 92 comprised of a host computer 94 having a local memory 98 and a central processing unit (CPU) 96. The host computer environment 92 is thus a system for identifying near matches between records in a data store, such as local memory 98, or a directly connected database 100, as example of which is a hard disk for the host computer 94. Accordingly, the CPU 96 of the host computer 94 preferably performs the steps of: creating one or more identifiers, such as a keystring 16, wherein each identifier is associated with a record in the data store; mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, such as creating the template 20; and retrieving all records from the data store having associated coordinate

sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record, as set forth above.

In another aspect, the present invention provides a computer-readable storage device, such as memory 98 or local database 100, containing instructions that upon execution cause a processor (CPU 96) to identify near matches between records in a data store (e.g. local database 100) and a selected record having an associated coordinate set. The device preferably performs the steps of creating one or more identifiers wherein each identifier is associated with a record in the data store, such as a keystring 16, mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, such as with template 20, and retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

With reference again to Fig.6, the host computer environment 92 and host computer 94 can be connected to any manner of computer or database and perform the method of finding near matches therein. As an example, host computer 94 is in direct connection with another computer 102 having a database 104, and the CPU 96 can access the data either resident on the directly connected computer 102, or the other database 104. Further, the host computer environment 92 can be connected to a local area network (LAN) 106 as are common in the art, and through the LAN 106, the host computer 94 can be in communication with one or more networked computers 108, each of which can have an attached database 110 that is accessible by the host computer 94. The host computer 94 can also be in communication through the LAN 106 with one or more networked databases 112, and can perform the record searching upon the data therein.

The host computer environment 92 can either directly, or through the LAN 106 as shown in Fig. 6, be connected to the Internet 114, or other wide area network (WAN). Thus, the host computer 94 can thereby access one or more databases 116 in communication with the Internet 114, and can also access other computers 118 in communication with the Internet 114 and any databases 120 accessible to the other computers 118 on the Internet 114. It should be appreciated that the present inventive system can therefore be used in any environment having a processor and a datastore as are known in the art, and is not to be limited to the host computer environment 92 and connective environments disclosed in Fig. 6.

The present inventive system accordingly provides a method for identifying near matches between records in a data store and a selected record that has an associated coordinate set. The method includes steps of creating one or more identifiers wherein each identifier is associated with a record in the data store, such as keystrings 16, mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store, and retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record. The identifier associated with each record in the data store preferably comprises one or more characters, such as keystring 16.

The method preferably further includes the step of determining a set of records from the retrieved records that match the selected record. The step of determining a set of records from the retrieved records that match the selected record is preferably screening the coordinate sets associated with the retrieved records for sets within a second predetermined distance (such as 'm' in Fig. 4b) in the discriminant space from the coordinate set associated with the selected record. And then the method further

includes the step of extracting, deleting, or otherwise modifying the determined set of records.

The method also preferably includes the step of acquiring a mapping template, such as the template in Fig. 3, and the step of acquiring is preferably creating or receiving a mapping template. The method then preferably includes the step of refining the acquired mapping template, as show in the processes of Figs. 4a and 4b.

The method can include the step of selecting an identifier format for use in the step of creating one or more identifier, and example of which is the process of Fig.5, and then further includes the step of acquiring one or more mapping templates. The step of selecting an identifier format is then preferably evaluating the acquired one or more mapping templates.

The method further preferably includes the steps of receiving the selected record, creating an identifier associated with the selected record, and mapping the identifier associated with the selected record into a coordinate set in the discriminant space associated with the selected record, as discussed above. Additionally, the method can further include the step of retrieving the coordinate set associated with the selected record from the datastore(s).

It should also be noted that the present inventive system and method can be implemented on any category of computer devices including the four main categories of digital computers: supercomputers, mainframe computers, minicomputers and microcomputers. The structures, processes, methods and system as disclosed herein can also be implemented on handheld computers, and Personal Digital Assistant (PDA) and Personal Information Management (PIM) devices including, but not limited to, cellular/mobile phones, Personal Organizers, Windows CE devices or hybrid devices such as a smart phone that may be deployed over a fixed or wireless network.

Moreover, the invention can be implemented on a variety of computing platforms and operating systems. For example, the present invention may be implemented on a standard personal computer (PC) operating under an operating system such as Windows, Windows NT, Unix, Linux, or other operating system. Standard development tools, languages and compilers all can be used to implement the processes described herein, under programming languages and development tools such as Java, C, C++, XML (Extensible Markup Language), Visual Basic, PowerBuilder, and other languages as known in the art.

Database files, either standard, relational or multidimensional, are the preferred file type to implement the invention. The databases can exist alone, in a data warehouse or in a data mart. These databases are operated upon by the processes may be created, managed, transformed and/or consolidated using a variety of database systems as are know in the art. These systems include but are not limited to Oracle, Sybase, Informix, Access, SQL, ODBC, Foxpro, XML schema or any other traditional or relational databases and/or database access tools.

Typical uses of this invention include locating duplicate records, locating near duplicate records, locating records with similar characteristics, and enhancing search capabilities in a database, data mart, or data warehouse. The invention also can used in locating duplicate URLs over the Internet and/or locating correct URLs when URLs are misspelled or typed incorrectly. The invention could further enhance Internet search capabilities in locating similar URLs or products on an e-business site. Locating similar products for an e-business site is another use of this invention. The methods and processes of the invention would be able to solve failed searches by providing a list of 'projectors' based on the Internet search for 'projecters.'

The present invention further can be applied in locating and extracting duplicate or near duplicate records in a customer or supplier database such as duplicate customer's name and address, customer order, and/or customer payment information. The methods and processes of the invention are no limited to test searches. Such capability can also locate duplicate or near duplicate customers/prospects in a Direct Marketing campaign or Sales Force Automation where there is data consolidation. The methods and processes used in this invention would allow one to compare all similar record sets to determine if duplicate data exists. This will then allow one to extract current customers from the prospected database.

A further application of the present invention is locating similar or near duplicate records that are possibly fraudulent in e-commerce applications which are conducted over the Internet 114. E-business fraud can include any electronic credit card or other transactions where similar records are fraudulently used as a unique record. For example, in e-business that given benefits for signing up, the present invention can detect new members that sign up multiple times by changing name slightly.

This invention can further synchronize database files. For example, Wireless devices are small and prone to input/data entry errors. As Personal Information Management (PIM) devices increase in popularity more data will exist in a variety of data sources that need to be synchronized. Data existing on LAN, WAN, PIM, Internet and Mainframe systems can be out of synchronization and this invention can be used to clean the synchronized data.

While there has been shown a preferred and alternate embodiments of the present invention, it is to be understood that certain changes may be made in the forms and arrangement of the elements and performance of the steps as set forth herein

without departing from the spirit of the invention as particularly set forth in the claims appended herewith. In addition, all means-plus-function language is intended to cover all equivalent structures, materials, and acts as known to one of skill in the art providing the elements or performing the steps as set forth in the elements of the claims.

CLAIMS

What is claimed is:

1. A method for identifying near matches between records in a data store and a selected record that has an associated coordinate set, the method comprising the steps of:

   (a) creating one or more identifiers, wherein each identifier is associated with a record in the data store;

   (b) mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store; and

   (c) retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

2. The method of claim 1, further comprising the step of determining a set of records from the retrieved records that match the selected record.

3. The method of claim 2, further comprising the step of extracting the determined set of records.

4. The method of claim 2, further comprising the step of deleting the determined set of records from the data store.

5. The method of claim 2, further comprising the step of modifying the determined set of records from the data store.

6. The method of claim 2, wherein the step of determining a set of records from the retrieved records that match the selected record comprises screening the coordinate sets associated with the retrieved records for sets within a second predetermined distance in the discriminant space from the coordinate set associated with the selected record.

7. The method of claim 1, further comprising the step of extracting the retrieved records.

8. The method of claim 1, further comprising the step of deleting the retrieved records from the data store.

9. The method of claim 1, further comprising the step of modifying the determined set of records from the data store.

10. The method of claim 1, further comprising the step of acquiring a mapping template.

11. The method of claim 10, wherein the step of acquiring a mapping template comprises creating a mapping template.

12. The method of claim 10, wherein the step of acquiring a mapping template comprises receiving a mapping template.

13. The method of claim 10, further comprising the step of refining the acquired mapping template.

14. The method of claim 1, further comprising the step of selecting an identifier format for use in the step of creating one or more identifier.

15. The method of claim 14, further comprising the step of acquiring one or more mapping templates.

16. The method of claim 15, wherein the step of selecting an identifier format comprises evaluating the acquired one or more mapping templates.

17. The method of claim 1, further comprising the steps of:

(d) receiving the selected record;

(e) creating an identifier associated with the selected record; and

(f) mapping the identifier associated with the selected record into a coordinate set in the discriminant space associated with the selected record.

18. The method of claim 1, further comprising the steps of retrieving the coordinate set associated with the selected record from the data store.

19. The method of claim 1, wherein the identifier associated with each record in the data store comprises one or more characters.

20. A system for identifying near matches between records in a data store and a selected record having an associated coordinate set, the system comprising:

(a) a data store for storing the records; and

(b) a processor for performing the steps of:

(c) creating one or more identifiers, wherein each identifier is associated with a record in the data store;

(d) mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store; and

(e) retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

21. A computer-readable storage device containing instructions that upon execution cause a processor to identify near matches between records in a data store and a selected record having an associated coordinate set by performing the steps comprising of:

(d) creating one or more identifiers, wherein each identifier is associated with a record in the data store;

(e) mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store; and

(f) retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

22. A system for identifying near matches between records in a data store and a selected record having an associated coordinate set, the system comprising:

(d) storing means for storing one or more records;

(e) creating means for creating one or more identifiers, wherein each identifier is associated with a record in the storing means;

(f) mapping means for mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the storing means; and

(g) retrieving means for retrieving all records from the storing means having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

AMENDED CLAIMS

[received by the International Bureau on 16 May 2000 (16.05.00);
original claims 20-22 amended; new claims 23-38 added;
remaining claims unchanged (3 pages)]

(a) a data store for storing the records; and

(b) a processor for performing the steps of:

(i) creating one or more identifiers, wherein each identifier is associated with a record in the data store;

(ii) mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store; and

(iii) retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

21. A computer-readable storage device containing instructions that upon execution cause a processor to identify near matches between records in a data store and a selected record having an associated coordinate set by performing the steps comprising of:

(a) creating one or more identifiers, wherein each identifier is associated with a record in the data store;

(b) mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the data store; and

(c) retrieving all records from the data store having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

22. A system for identifying near matches between records in a data store and a selected record having an associated coordinate set, the system comprising:

(a) storing means for storing one or more records;

(b) creating means for creating one or more identifiers, wherein each identifier is associated with a record in the storing means;

(c) mapping means for mapping each of the one or more identifiers into a set of coordinates in a discriminant space associated with each record in the storing means; and

(d) retrieving means for retrieving all records from the storing means having associated coordinate sets within a predetermined distance in the discriminant space from the coordinate set associated with the selected record.

23. The method of claim 1, wherein each record comprises a URL.

24. The method of claim 23, wherein the step of creating identifiers comprises creating identifiers based upon the URL in each record.

25. The method of claim 1, wherein each record comprises contact information associated with a person or entity.

26. The method of claim 25, wherein the contact information contains at least one type of information selected from the group consisting of name, address, identification number and telephone number.

27. The system of claim 20, wherein each record comprises a URL.

28. The system of claim 27, wherein the step of creating identifiers comprises creating identifiers based upon the URL in each record.

29. The system of claim 20, wherein each record comprises contact information associated with a person or entity.

30. The system of claim 29, wherein the contact information contains at least one type of information selected from the group consisting of name, address, identification number and telephone number.

31. The storage device of claim 21, wherein each record comprises a URL.

32. The storage device of claim 31, wherein the step of creating identifiers comprises creating identifiers based upon the URL in each record.

33. The storage device of claim 21, wherein each record comprises contact information associated with a person or entity.

34. The storage device of claim 33, wherein the contact information contains at least one type of information selected from the group consisting of name, address, identification number and telephone number.

35. The system of claim 22, wherein each record comprises a URL.

36. The system of claim 35, wherein the creating means comprises means for creating identifiers based upon the URL in each record.

37. The system of claim 22, wherein each record comprises contact information associated with a person or entity.

38. The system of claim 37, wherein the contact information contains at least one type of information selected from the group consisting of name, address, identification number and telephone number.

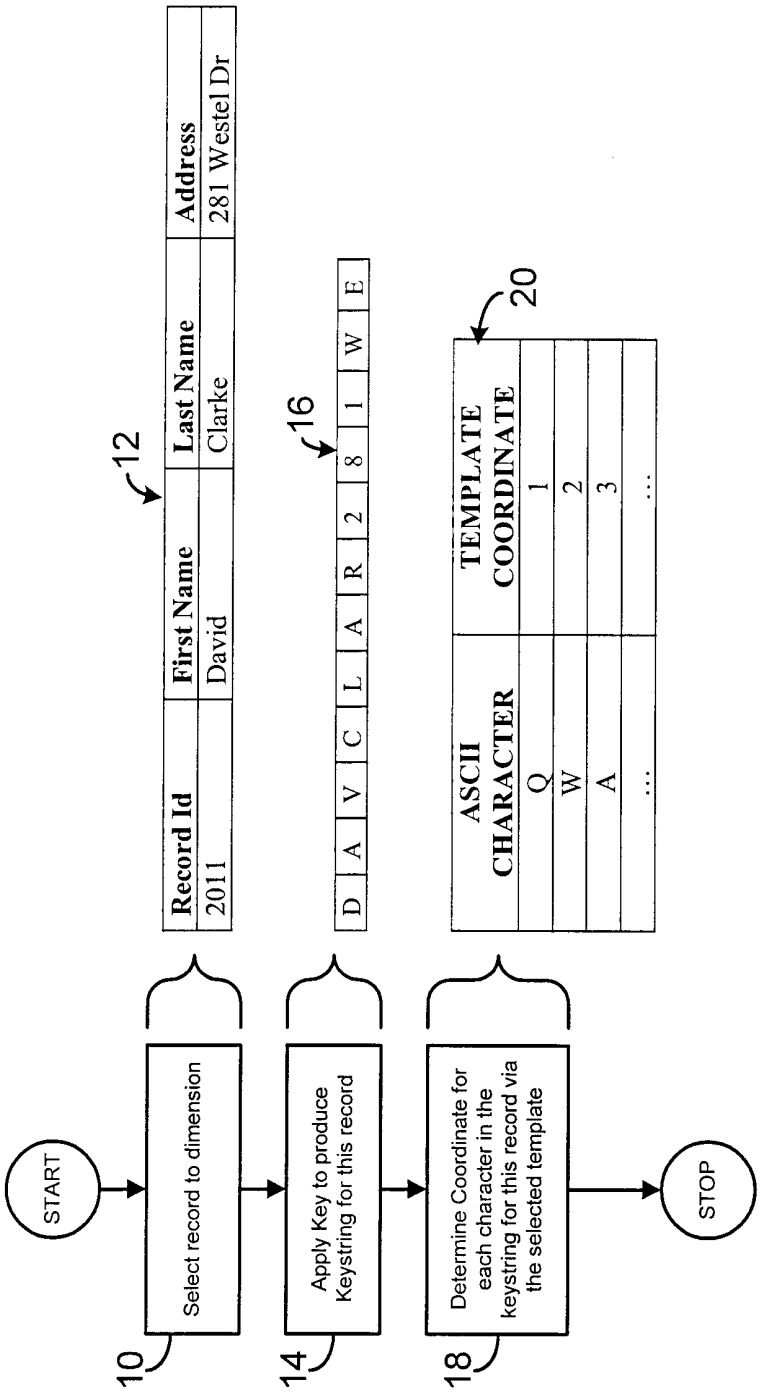

START
Select record to dimension
Apply Key to produce Keystring for this record
Determine Coordinate for each character in the keystring for this record via the selected template
STOP
10
14
18
12
16
20
Record Id
First Name
Last Name
Address
2011
David
Clarke
281 Westel Dr
D A V C L A R 2 8 1 W E
ASCII CHARACTER
TEMPLATE COORDINATE
Q 1
W 2
A 3
... ...

Fig. 1

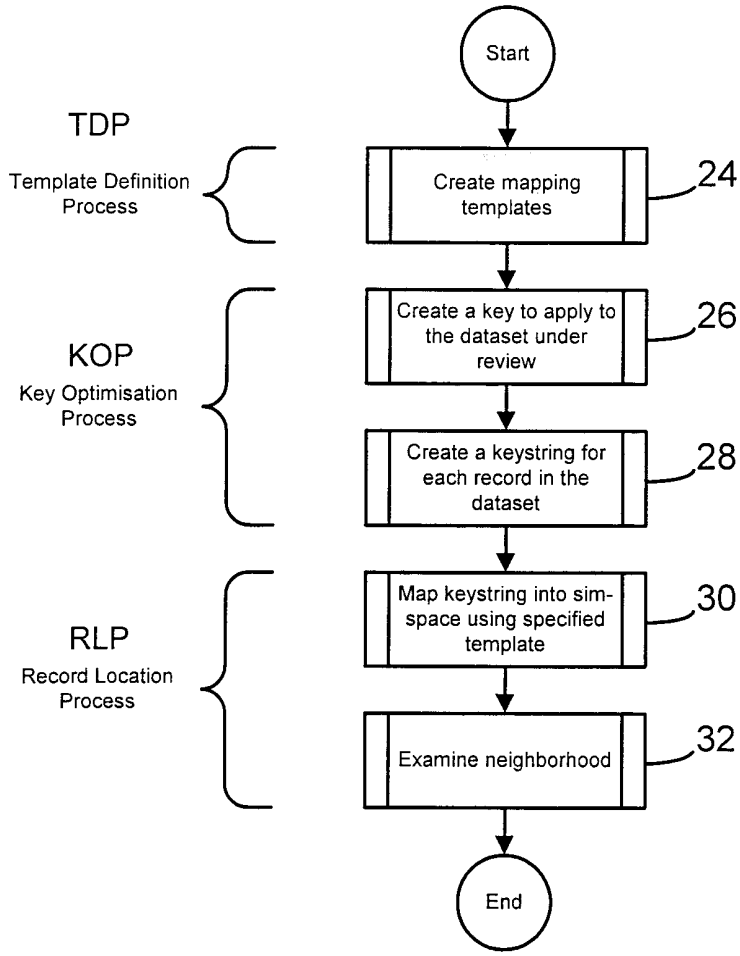

Start
TDP
Template Definition Process
Create mapping templates
24
KOP
Key Optimisation Process
Create a key to apply to the dataset under review
26
Create a keystring for each record in the dataset
28
RLP
Record Location Process
Map keystring into sim-space using specified template
30
Examine neighborhood
32
End

Fig. 2

38 40

ASCII CHARACTER	TEMPLATE COORDINATE
Q	1
W	2
A	3
S	4
E	5
D	6
Z	7
X	8
R	9
F	10
C	11
T	12
G	13
V	14
Y	15
H	16
B	17
U	18
J	19
N	20
I	21
K	22
N	23
O	24
L	25
P	26

36

44 46

ASCII CHARACTER	TEMPLATE COORDINATE
0	1
1	2
4	3
2	4
7	5
5	6
3	7
8	8
6	9
9	10

42

Fig. 3

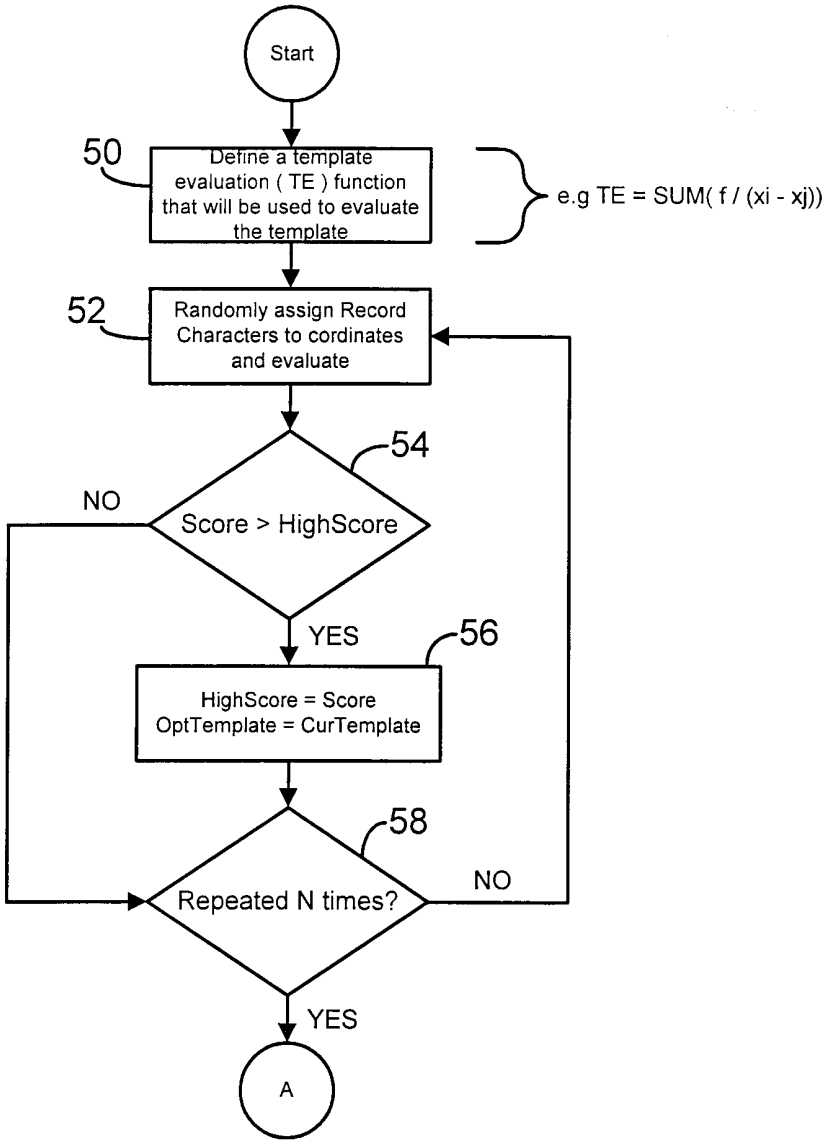

Start
50
Define a template evaluation (TE) function that will be used to evaluate the template
e.g TE = SUM(f / (xi - xj))
52
Randomly assign Record Characters to cordinates and evaluate
54
Score > HighScore
NO
YES
56
HighScore = Score
OptTemplate = CurTemplate
58
Repeated N times?
NO
YES
A

Fig. 4a

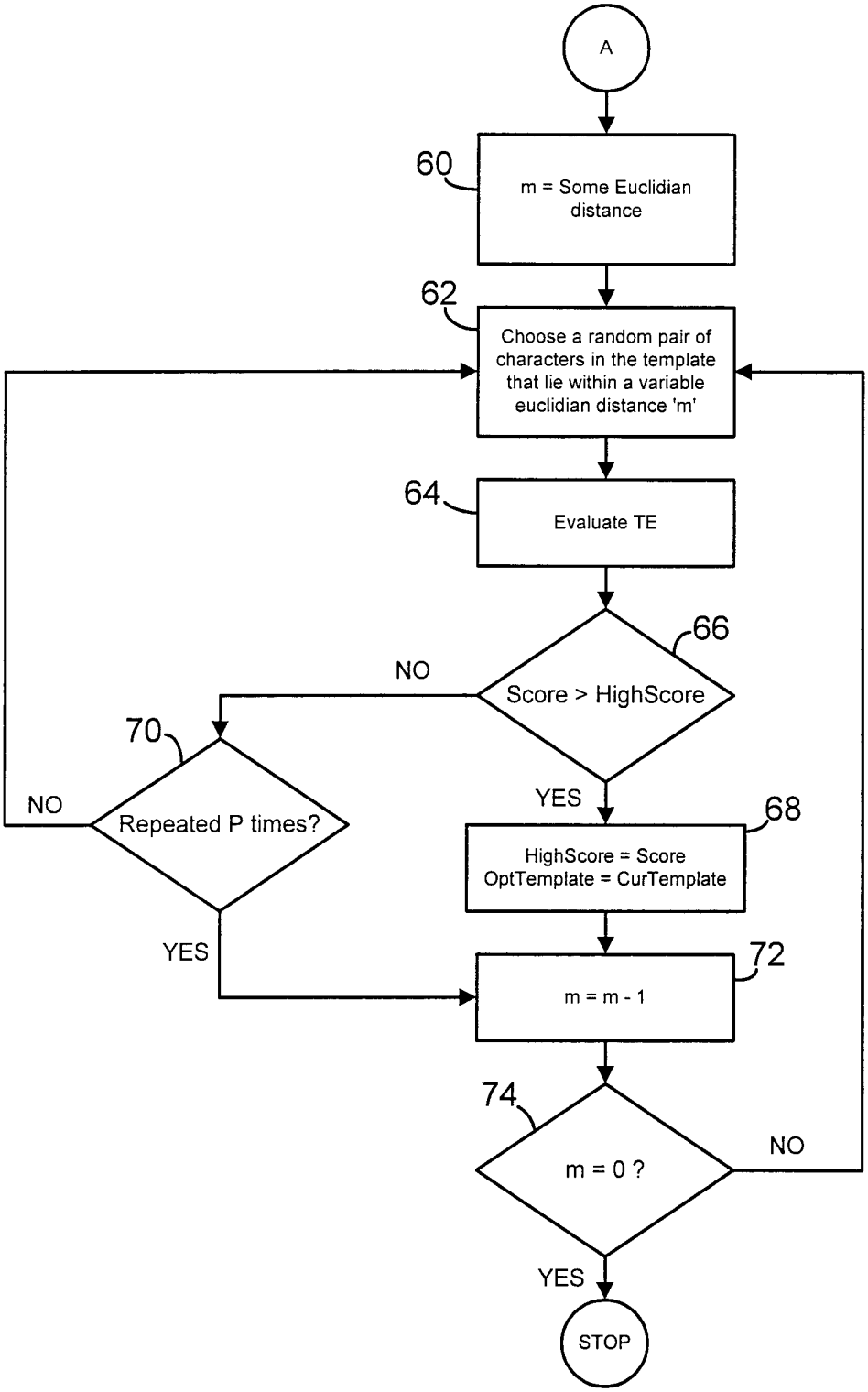

A
60
m = Some Euclidian distance
62
Choose a random pair of characters in the template that lie within a variable euclidian distance 'm'
64
Evaluate TE
66
Score > HighScore
NO
YES
70
Repeated P times?
NO
YES
68
HighScore = Score
OptTemplate = CurTemplate
72
m = m - 1
74
m = 0 ?
NO
YES
STOP

Fig. 4b

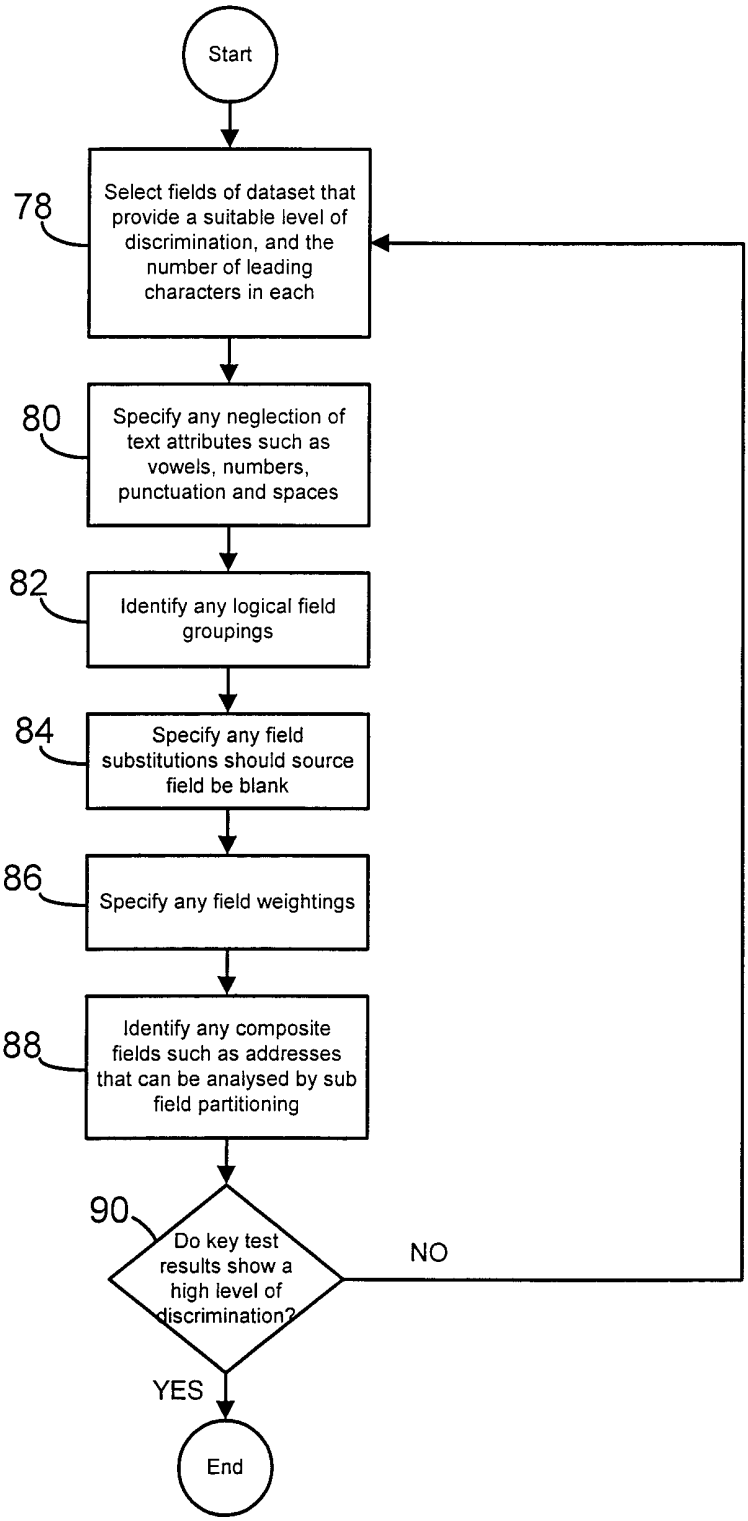

Start
78
Select fields of dataset that provide a suitable level of discrimination, and the number of leading characters in each
80
Specify any neglection of text attributes such as vowels, numbers, punctuation and spaces
82
Identify any logical field groupings
84
Specify any field substitutions should source field be blank
86
Specify any field weightings
88
Identify any composite fields such as addresses that can be analysed by sub field partitioning
90
Do key test results show a high level of discrimination?
NO
YES
End

Fig. 5

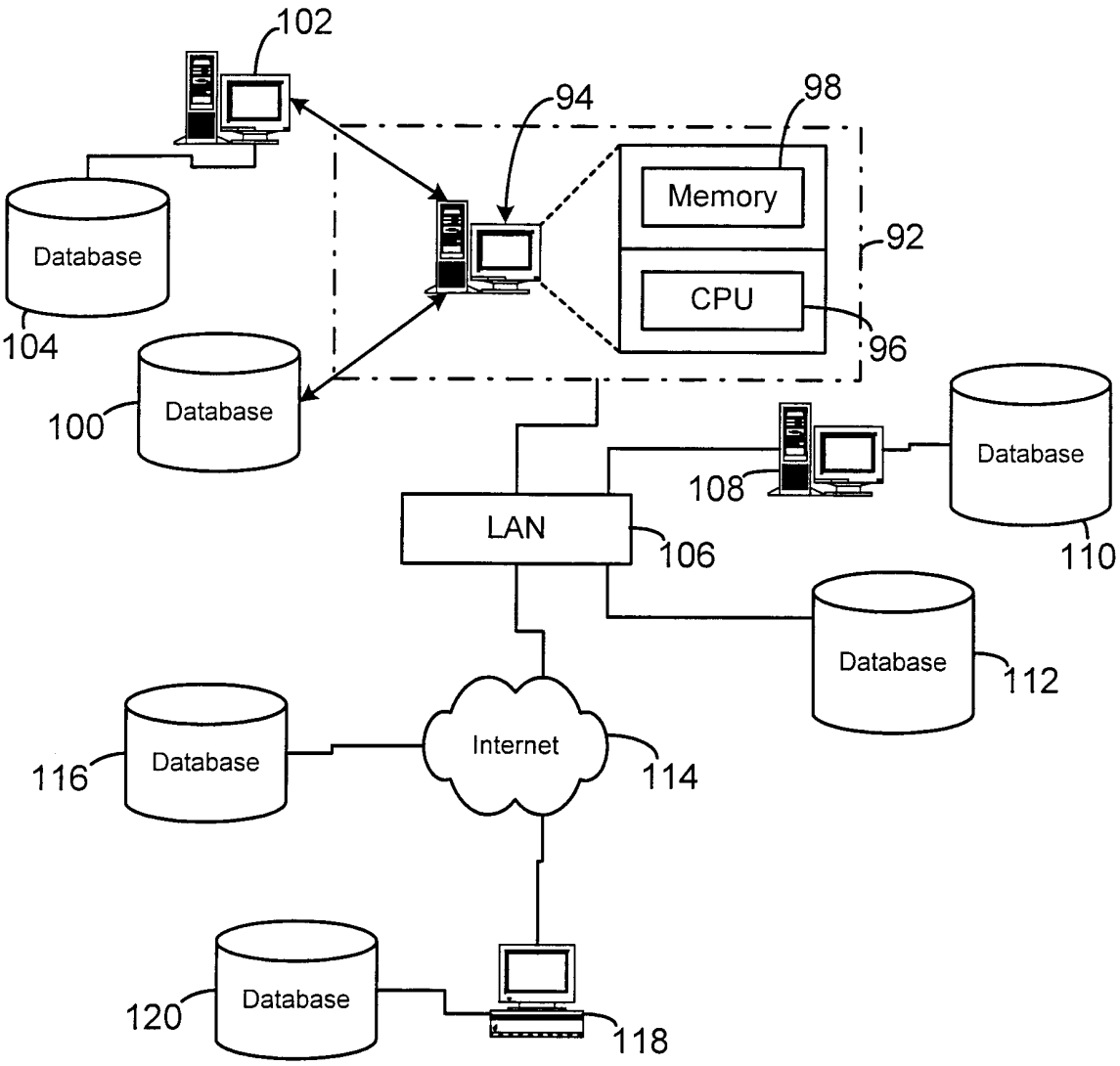

102
94
98
Memory
92
CPU
96
Database
104
Database
100
108
Database
110
LAN
106
Database
112
Internet
114
Database
116
Database
120
118

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/28870

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :GO6F 17/30
US CL :707/6

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. : 707/6, 3, 4, 5, 501, 530, 531, 536

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)

West, CAS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,E	US 6,026,398 A [BROWN ET AL.] 15 FEBRUARY 2000, SEE FIG 2	1-20
Y	US 5,649,183 A [BERKOWITZ ET AL.] 15 JULY 1997, SEE FIG 3.	1-20
A,E	US 6,029,167 A [EVANS] 22 FEBRUARY 2000, SEE ABSTRACT	1
A	US 5,465,353 A [HULL ET AL.] 07 NOVEMBER 1995, SEE FIG 1	1

☐ Further documents are listed in the continuation of Box C. ☐ See patent family 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 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
24 MARCH 2000	14 APR 2000
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 SANJIV SHAH James R. Matthews Telephone No. (703) 305-8355

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