Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2401170 A1 2001/09/07

(21) 2 401 170

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2001/02/28
(87) Date publication PCT/PCT Publication Date: 2001/09/07
(85) Entrée phase nationale/National Entry: 2002/08/22
(86) N° demande PCT/PCT Application No.: US 2001/006447
(87) N° publication PCT/PCT Publication No.: 2001/065416
(30) Priorité/Priority: 2000/02/28 (09/514,743) US

(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/30

(71) Demandeur/Applicant:
VALITY TECHNOLOGY INCORPORATED, US

(72) Inventeur/Inventor:
JARO, MATTHEW A., US

(74) Agent: SMART & BIGGAR

(54) Titre : MOTEUR D'APPARIEMENT PROBABILISTE
(54) Title: PROBABILISTIC MATCHING ENGINE

(57) Abrégé/Abstract:
The method and apparatus enable information to be retrieved from an electronic database based on a probabilistic approach and some query processing. In one aspect, records of a database are parsed into record tokens using a pattern action language before an index for the records is created. In another aspect, a table of index tokens is created wherein the table comprises a frequency of occurrence in the database for each index token and each index token comprises a phonetic equivalent for a respective record token. In one aspect, a query is parsed into query tokens using a pattern action language, a search token is generated from a query token, and the search token is used to access database records. In another aspect, a search token comprises a phonetic equivalent for a query token or a token that qualifies as similar to a query token and search token and a search token is used to access database records. The qualification of a token as similar to a query token is based on a comparison of the query token to a database dictionary using an information theoretic algorithm. In yet another aspect, a token is selected, the selected token is used to access database records, a likelihood of relevance to the query is calculated for each of the records, and the highest likelihood of relevance to the query is compared to a continuation threshold. If the continuation threshold is exceeded, no more records are accessed and the accessed records are output. If the continuation threshold is not exceeded, the selected search token is eliminated from the set of available search tokens, and a new token is selected for accessing database records.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 September 2001 (07.09.2001)

PCT

(10) International Publication Number
WO 01/65416 A2

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

(21) International Application Number: PCT/US01/06447

(22) International Filing Date: 28 February 2001 (28.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/514,743 28 February 2000 (28.02.2000) US

(71) Applicant: **VALITY TECHNOLOGY INCORPORATED** [US/US]; 100 Summer Street, 15th floor, Boston, MA 02110 (US).

(72) Inventor: **JARO, Matthew, A.**; 8 Litchfield Road, Lexington, MA 02420 (US).

(74) Agent: **FREEMAN, Kia, L.**; Testa, Hurwitz & Thibeault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).

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

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

Published:
— *without international search report and to be republished upon receipt of that report*

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

(54) Title: PROBABILISTIC MATCHING ENGINE

(57) Abstract: The method and apparatus enable information to be retrieved from an electronic database based on a probabilistic approach and some query processing. In one aspect, records of a database are parsed into record tokens using a pattern action language before an index for the records is created. In another aspect, a table of index tokens is created wherein the table comprises a frequency of occurrence in the database for each index token and each index token comprises a phonetic equivalent for a respective record token. In one aspect, a query is parsed into query tokens using a pattern action language, a search token is generated from a query token, and the search token is used to access database records. In another aspect, a search token comprises a phonetic equivalent for a query token or a token that qualifies as similar to a query token and search token and a search token is used to access database records. The qualification of a token as similar to a query token is based on a comparison of the query token to a database dictionary using an information theoretic algorithm. In yet another aspect, a token is selected, the selected token is used to access database records, a likelihood of relevance to the query is calculated for each of the records, and the highest likelihood of relevance to the query is compared to a continuation threshold. If the continuation threshold is exceeded, no more records are accessed and the accessed records are output. If the continuation threshold is not exceeded, the selected search token is eliminated from the set of available search tokens, and a new token is selected for accessing database records.

WO 01/65416 A2

PROBABILISTIC MATCHING ENGINE

Technical Field

The present invention generally relates to database information retrieval techniques. In particular, the present invention relates to database information retrieval based on record linkage theory with query expansion.

Background Information

Although the distinction is not always clear cut, information retrieval has traditionally been classified as belonging to one of two genres: browsing or querying. Browsing is typically more passive than querying. Browsing involves a user accessing a portion of a database through a simple mechanism, such as a menu topic, and then exploring the accessed information by navigating through it, often with some degree of information retrieval system guidance. Hypertext systems generally support a browsing approach to information retrieval. Although perceived as demanding less of a user, browsing is not necessarily the most efficient way to retrieve information from a large database.

In contrast to browsing, querying requires a user to specify the information that is of interest to him. Querying will only be successful when the information of interest is specified in a way that matches the database language. The match often requires a compromise in the selection of query terms. Querying can be perceived as taxing on a user, particularly if the user is untrained. Querying can also produce poor retrieval results. Querying itself has traditionally been classified as belonging to one of two genres: querying done in connection with Boolean retrieval and querying done in connection with probabilistic retrieval.

Querying in connection with Boolean retrieval is the most established form of information retrieval. It requires a user to create an appropriate combination of terms which match both the information of interest and the database language. Boolean searching requires a user to specify only a limited number of terms to achieve an acceptable number of retrieval results. Optimal Boolean searching requires the user to be familiar with the Boolean operators and with the effective ways to combine terms. Nonetheless, users rarely make explicit use of Boolean operators.

Querying in connection with probabilistic retrieval offers users a greater scope of retrieval. Retrieval results are typically compared to the query terms using an algorithm based on probability theory and rated on how closely they match the query terms. Terms that occur less frequently in a database are considered more discriminating and are typically given more weight in predicting a match. A user is not constrained in the number of query terms he may use because the rating of the retrieval results mitigates the problem of excessive retrieval results.

Nonetheless, problems remain in querying an electronic database with a probabilistic retrieval method. Misspellings and nonstandard spellings in the query or the database may cause relevant information to be overlooked in the retrieval process. Similarly, nonstandard formatting of information in the query or the database may cause relevant information to be overlooked. If retrieval speed is an issue, the specification of a large number of query terms may result in an unsatisfactorily slow response to a query. A user may equally become frustrated if he has to wait for search results because the database is tied up with another search. Conversely, a user may become frustrated by a fast search that returns poor results.

Summary Of The Invention

In one aspect the invention includes a method for indexing a database. Records of a database are input. Each record is parsed into record tokens using a pattern action language. An index to the record is created from the record tokens for each record.

In one embodiment, the parsing includes converting each record into original tokens, characterizing each original token, and converting the characterized original tokens into record tokens based on the pattern action language. In a related embodiment, the pattern action language is responsive to the domain with which the record token is associated.

In another embodiment, the index creation includes creating a list of unique index tokens from the record tokens for each record, calculating a frequency of occurrence in the database for each unique index token, and creating a table of index tokens. The table of index tokens contains the frequency of occurrence in the database for each unique index token. In a related embodiment, an index token comprises a phonetic equivalent for the respective record token. In further related embodiment, a list of unique record tokens is also created.

In another aspect, the invention includes a method for indexing a database. Records of a database are input and each record is parsed into record tokens. An index token is generated from a respective record token. The index token is a phonetic equivalent for the record token. A frequency of occurrence in the database is calculated for a unique index token. A table of index

tokens is created. The table of index tokens includes the frequency of occurrence for the unique index tokens.

In one embodiment, a list of unique record tokens is also created. In a related embodiment, each record is parsed into record tokens using a pattern action language. In another related embodiment, the parsing includes converting each record into original tokens, characterizing each original token, and converting the characterized original tokens into record tokens based on the pattern action language.

In some embodiments of the foregoing aspects, an index to the database is created from the record tokens for each record. Each record token is associated with a domain in the database, the pattern action language is responsive to the domain, the frequency of occurrence is calculated with respect to a domain in the database, and the index of unique record tokens list the frequency of occurrence by domain.

In one aspect, the invention relates to an apparatus for indexing a database. An input device accepts records of a database. A parser parses the records into record tokens, and an indexer generates an index of the record tokens in the database.

In one embodiment, the parser includes a tokenizer, a token characterizer, and a token converter. The tokenizer converts records into original tokens. The token characterizer characterizes each original token, and the token converter converts the characterized original tokens into record tokens based on a pattern action language. In a related embodiment, the pattern action language is responsive to the domain with which a record token is associated.

In another embodiment, the indexer includes a token comparator, a frequency calculator, and a table generator. The token comparator creates a list of unique index tokens from the record tokens. The frequency calculator calculates a frequency of occurrence in the database for the unique index tokens. The table generator generates a table containing a frequency of occurrence for the unique index tokens. In a related embodiment, an index token is a phonetic equivalent for the respective record token and the tokens comparator communications with the parser via the token generator. In another related embodiment, a record token comparator also creates a list of unique record tokens.

In another aspect, the invention relates to an apparatus for indexing a database. An input device accepts records of a database, and a parser parses the records into record tokens. A token generator generates an index token from a respective record token. The index token is a phonetic equivalent of the respective record token. A table generator generates a table containing for each

index token a frequency of occurrence of the index token in the database, calculated by a frequency calculator, and a pointer to all records containing the index token.

In one embodiment, a record token comparator creates a list of unique record tokens from the record tokens for each record. In a related embodiment, the table generator generates a table that contains a pointer to each record in the database that contains an index token corresponding to said unique index token. In another related embodiment, a record token comparator in communication with the parser also creates a list of unique record tokens.

In further related embodiment, the parser parses each record using a pattern action language. In one such embodiment, the parser further includes a tokenizer, a token characterizer, and a token converter. The tokenizer converts records into original tokens. The token characterizer characterizes each original token, and the token converter converts the characterized original tokens into record tokens based on the pattern action language.

In some embodiments, the original token, the respective record token, and all respective index tokens are all associated with the same domain in the database, the pattern recognition is responsive to the domain associated with a token, and the frequency of occurrence for an index token is calculated by domain. In one embodiment, A table generator generates a table containing for unique index tokens the frequency of occurrence and a pointer to each record in the database containing the corresponding record token.

In one aspect, the invention relates to a method of querying a database. A query is input and parsed into query tokens using a pattern action language. A search token is generated from a query token. The search token is looking up on an index table to access a record within the database.

In one embodiment, the parsing includes converting the query into original tokens, characterizing each original token, and converting the characterized original tokens into the query tokens based on the pattern action language. In a related embodiment, an original token and the resulting query token are associated with the same domain in the database. In a further related embodiment, the pattern action language is responsive to the domain with which the tokens are associated.

In a different further related embodiment, a search token is generated from a query token. and the search token is associated with the domain in the database with which the respective query token is associated.

In another aspect, the invention relates to a method of querying a database. A query is

input and parsed into query tokens. A search token is generated from a query token. Search token generation includes checking a list of unique record tokens for a token that is similar to the query token based on an information theoretic algorithm. It also includes translating query tokens and similar tokens into search tokens. The search tokens are phonetic equivalents for the query tokens or the similar tokens. A search token is looked up on an index table to access a record within the database.

In one embodiment, a search token is associated with the same domain in the database as the respective query token. In a related embodiment, the parsing is done using a pattern action language. In further related embodiment, the parsing includes converting the query into original tokens, characterizing each original token, and converting the characterized original tokens into query tokens based on the pattern action language.

In one aspect, the invention relates to an apparatus for querying a database. A query input device accepts a query as input. A parser parses the input into query tokens using a pattern action language. A generator generates search token from the query tokens. A database accessor accesses records in the database in response to a search token.

In one embodiment, the parser includes a tokenizer, a characterizer, and a converter. The tokenizer creates original tokens from the input, and the characterizer characterizes each of them. The converter converts the characterized original tokens into query tokens based on the pattern action language. In a related embodiment, the original token is associated with the same domain in the database as the respective query tokens and search token. In another related embodiment, the tokens are associated with the same domain in the database, and the pattern action language is responsive to the domain with which they are associated.

In another aspect, the invention relates to an apparatus for querying a database. A query input device accepts input, and a parser parses it into query tokens. A generator generates search tokens from the query tokens. The generator includes a query expander that adds tokens that qualify as similar to a query token based on an information theoretic algorithm. These are similar tokens. The generator also includes a translator that translates each query token and similar token into a phonetically-equivalent search token. A database accessor finds pointers to records in the database with a search token.

In one embodiment, each query token, respective similar token, and respective search token are all associated with the same domain in the database. In another embodiment, the parser parses uses a pattern action language. In a related embodiment, the parser includes a tokenizer, a

characterizer, and a converter.

In one aspect, the invention relates to a method for accessing data within a database. A token is selected from a set as the first token with which to search. A set of records is retrieved from the database in response to the selected token. A likelihood of relevance to the query is determined for each record in the set. The set of records is ordered by likelihood of relevance to the query. The highest likelihood of relevance to the query for the set is compared to a continuation threshold. If the threshold is exceeded, the search is terminated and the set of records is output. If not, a different token is selected for a new search.

In one embodiment, likelihood of relevance to the query is determined based on Record Linkage Theory. In a related embodiment, the set of records consists of more than one record and the output records are ordered by likelihood of relevance to the query.

In another embodiment, a frequency of occurrence in a database is identified for each token, the tokens are ordered by frequency of occurrence, and the token having the lowest frequency of occurrence is selected as the first search token. If the continuation threshold is not exceeded, the token with the next lowest token is selected as the next search token. In a related embodiment, the frequencies of occurrence relate to domains in the database and tokens are each associated with a domain. In such an embodiment, tokens are ordered and the token having the lowest frequency of occurrence in the associated domain is the first selected token. In a related embodiment, a likelihood of relevance to the query is determined for each record based on Record Linkage Theory. In further related embodiment, if a buffer of retrieved records overflows, the buffer is cleared and a new search is begun for records contain all of the tokens.

In another aspect, the invention relates to an apparatus for accessing data within a database. A database accessor retrieves a set of records from the database in response to the token selected as the first token on which to search by the token selector. A relevance determiner determines the likelihood of relevance to the query for each record in the set or records. A relevance comparator orders each record in the set by likelihood of relevance and a threshold comparator compares a continuation threshold to the highest likelihood of relevance. If the continuation threshold is exceeded, the relevance comparator terminates the search. If not, the relevance comparator removes the selected token and allows the token selector to select another token. An output device returns the set of records when the threshold comparator terminates the search.

In one embodiment, the likelihood of relevance to the query is determined based on

Record Linkage Theory. In a related embodiment, the database accessor retrieves more than one record and the output device returns the records ordered by likelihood of relevance to the query.

In another embodiment, a frequency comparator identifies a frequency of occurrence in the database for each token and orders the tokens by the frequency of occurrence. The token selector selects the token having the lowest frequency of occurrence as the first token on which to search. In a related embodiment, the frequency comparator identifies a frequency of occurrence in the domain in the database with which the token is associated and selects the token having the lowest frequency of occurrence in the associated domain as the first token. In another related embodiment, the relevance determiner determines a likelihood of relevance to a query based on Record Linkage Theory. In further related embodiment, a buffer overflow arrestor clears a buffer when it overflows and sends an overflow signal to the token selector. The database accessor then retrieves the set of records from the database that contain all of the tokens.

Brief Description Of The Drawings

In the drawings, like reference characters generally refer to the same parts throughout the different figures. Also, emphasis is generally being placed upon illustrating the principles of the invention.

FIG. 1 is a functional block diagram of the information retrieval process as known to the prior art.

FIG. 1A describes an embodiment of the evolution of records throughout the indexing process in accordance with the invention.

FIG. 1B describes an embodiment of the evolution of a query throughout query processing in accordance with the invention.

FIG. 1C describes an embodiment of the interaction of the search token and record in the information accessing process in accordance with the invention.

FIG. 2 is a functional block diagram of an embodiment of the information indexing portion of the information retrieval process performed in accordance with the invention.

FIG. 3 is a functional block diagram of an embodiment of the query processing portion of the information retrieval process performed in accordance with the invention.

FIG. 4 is a functional block diagram of an embodiment of the information accessing portion of the information retrieval process performed in accordance with the invention.

Description

In brief the present invention relates in general to the information retrieval process for an electronic database as illustrated in FIG. 1. The information retrieval process is a process by which a query is used to access existing reference data in a database. In the present invention, probability theory is used to select records in a database according to a user query and retrieve them. The information retrieval process can generally be separated into three steps as illustrated in FIG. 1: indexing the reference data, processing the query, and accessing the reference data. The last two steps of the information retrieval process may be considered the search phase.

A database generally includes many records, each of which may be referred to by record number. Each record generally includes several domains. Similarly, each domain generally includes several fields. Each field may further contain free form text. For example, an Internal Revenue Service database may contain a separate record for each taxpayer. The taxpayer record may be numbered and may include separate domains for the home and work address of the taxpayer. Each address domain may contain a street field, a town field, a zip code field, and other fields. The street field, for example, may accept free form text such as "10910 Way Thru The Woods" or "71 Camino De Gracia." Databases typically do not require that every field or domain include information. For example, a taxpayer working as a freelance photographer may not have a work address so that taxpayer's work address domain may not include any data.

Other database arrangements are possible and the information retrieval process of the present invention can easily be applied to those situations. Nonetheless, for the purposes of this application, the reference data in a database is presumed to include a number of records, each record including a number of domains, each domain including one or more fields, each field containing free form text. With the presumed arrangement of the reference data, the present invention operates on free form text residing in fields.

In brief overview, the first step in the information retrieval process, a block diagram of which is illustrated in FIG. 1, is to index (STEP 10) the reference data. Indexing reference data may be considered preparation of the reference data for the search phase of the information retrieval process. FIG. 1A illustrates the evolution of a database record during the indexing process according to one embodiment of the present invention. To begin the indexing, the elements 42 of each record 44 are parsed into a set of record tokens (T_{R_n}) 46. The parsing process in some embodiments includes elimination of some portions of the record and

standardization of other portions of the record. In the embodiment shown in FIG. 1A, index tokens (T_{I_n}) 62 are then generated from record tokens (T_{R_n}) 46.

To conclude the indexing, the index tokens (T_{I_n}) 62 and record tokens (T_{R_n}) 46 are analyzed to facilitate later searching. In one embodiment, a list of unique record tokens (T_{R_n}) 46 contained in the reference data is created. In one embodiment, a table 96 of unique index tokens (T_{I_n}) 62 is created. In a related embodiment, the table 96 includes the frequency of occurrence (ν_n) 92 in the database for each unique index token (T_{I_n}) 62. In another related embodiment, the table 96 includes pointers 94 to the records in the reference data that contain the tokens. In the embodiment shown in FIG. 1A, there is one comprehensive table containing all of the available indexing information. In another embodiment, there are numerous tables containing portions of the available indexing information.

The second step in the information retrieval process illustrated in FIG. 1 is to process (STEP 20) the query. Processing the query may be considered preparation of the query for use in the information accessing phase of the information retrieval process. FIG. 1B illustrates the evolution of a query 54 during query processing according to one embodiment of the present invention. Elements 52 of the query 54 are parsed into a set of query tokens (T_{Q_n}) 56. In the embodiment shown in FIG. 1B, the parsing process includes elimination of some portions of the query 54 and standardization of other portions of the query. In one embodiment, any token from a list of record tokens (T_{R_n}) 46 that qualifies as similar to a query token (T_{Q_n}) 56 based on an information theoretic algorithm is added to the set of query tokens. In one embodiment, search tokens (T_{S_n}) 72, that can be used to access records in the reference data, are generated from query tokens (T_{Q_n}) 56 and similar tokens. In some embodiments, the processing of a query corresponds to the processing of the records in the reference data.

The third step in the information retrieval process illustrated in FIG. 1 is to access (STEP 30) the reference data. Accessing the reference data is the culmination of the preparation of the reference data and the query. FIG. 1C illustrates the accessing process according to one embodiment of the present invention. In one embodiment, in accord with a probabilistic search model, a search token (T_{S_n}) 72 is selected from the set of search tokens based on the selectivity of the search token. Records 44 from the reference data containing the search token (T_{S_n}) 72 are retrieved using a token table 96. In one embodiment, a weight is calculated for each record representing the likelihood that it is relevant to the user query 54. In a related embodiment, the weight calculation is based on Record Linkage Theory. In one embodiment, the maximum

weight for a set of retrieved records is compared to a threshold to determine whether the search should continue or be terminated. In one embodiment, the retrieved records are ordered and returned to the user. In some embodiments, the weight of each record is returned to the user alone or in association with the record. The final result of the information retrieval process is the user having a list or records and, in some embodiments, weights to evaluate each record's relevance to the query.

Referring now to FIG. 2, the figure illustrates a detailed block diagram of the process of indexing reference data according to one embodiment. The first step is to parse (STEP 40) a record 44 of the reference data. Parsing the record into tokens includes separating the data in the record into a set of tokens. In some embodiments, the developer of the reference data defines a set of individual characters to be used as the basis for separation of the contents of a record into tokens. In some such embodiments, these developer-defined characters are used alone. In other such embodiments, these developer-defined characters are used in addition to default characters as the basis for separation. In other embodiments, the developer allows default characters to be used as the sole basis for the separation. A group of characters may be used as a basis for separation. In some embodiments, the separation characters themselves become tokens.

For example, a record containing "big;bad.wolf and redriding hood" becomes "<big><;><bad><.><wolf and redriding hood>" where the semicolon and period are defined as the individual separation characters and the "<" and ">" indicates token boundaries. Similarly, a record containing "big;bad.wolf and redriding hood" becomes "<big;bad.wolf><and><redriding><hood>" where the a space is defined as the separation character and the "<" and ">" again indicate token boundaries. In other embodiments, the separation characters are eliminated in the separation process. In some embodiments, different characters are used as the basis for separation in different fields or domains.

In some embodiments, parsing the record includes eliminating some tokens. In some embodiments, the developer defines a set of tokens to be eliminated after the separation of the contents of a record into tokens. In one embodiment, the developer defined tokens are the sole tokens that are eliminated. In another embodiment, the developer defined tokens are eliminated in addition to the default tokens. In other embodiments, the developer simply allows default tokens to be eliminated. A token to be eliminated need not consist of a single character. For example, "<big><;><bad><.><wolf and redriding hood>" becomes "<big><bad><wolf and redriding hood>" where the semicolon and period are defined as tokens to be eliminated. In

some embodiments, the developer defines different tokens to be eliminated in different fields or domains.

In some embodiments, parsing the record includes examining the set of tokens that results from the separation process for patterns and acting upon one or more tokens in a recognized pattern. In such embodiments, the attributes of each token are determined once a record has been separated into tokens. In one embodiment, the attributes include class, length, value, abbreviation, and substring. In other embodiments, additional attributes are determined. In other embodiments, different attributes are determined. In yet other embodiments, fewer attributes are determined. In any embodiment, the determination of some attributes of a token may negate the requirement to determine other attributes of the token. In one embodiment which uses the class attribute, classes include Numeric, Alphabetic, Leading Numeric followed by one or more letters, Leading Alphabetic followed by one or more numbers, Complex Mix containing a mixture of numbers and letters that do not fit into either of the two previous classes, and Special containing a special characters that are not generally encountered. In another embodiment which uses the class attribute, other classes are defined. In one embodiment, the Alphabetic classification is case sensitive. In some embodiments, additional developer defined classes are used in conjunction with default classes. In other embodiments, developer defined classes are used to the exclusion of the default classes. For example, in one embodiment, the token <aBcdef> has the attributes of an Alphabetic class token with a length of 6 characters and a value of "aBcdef" where the Alphabetic tokens are case sensitive.

In embodiments in which parsing includes modifying a recognized pattern of tokens in some way, a pattern must be defined for recognition based on the possible attributes of the tokens. In some such embodiments, a pattern is defined for action only if it occurs in a specific domain. In other such embodiments, a pattern is defined for action if it occurs anywhere in the set of record tokens. Pattern matching begins with the first token and proceeds one token at a time. There may be multiple pattern matches to a record. A pattern is defined by any of the attributes of a token, a portion of a token, or a set of tokens. In one embodiment, a pattern is defined by the attributes of a single token. In another embodiment, a pattern is defined by the attributes of a set of tokens. For example, in one embodiment, the pattern is defined as a token with length of less than 10, a substring of "ANTI" in the first through fourth characters of a token, and a substring of "CS" without constraint on where it occurs in the token. In the example embodiment, the tokens <ANTICS> and <ANTI-CSAR> will be recognized for action. In

contrast, in the example embodiment, the token <ANTIPATHY> will not be recognized due to failure to meet the second substring constraint and the token <ANTIPOLITICS> will not be recognized due to failure to meet the length constraint.

In embodiments in which parsing includes modifying a recognized pattern of tokens in some way, a number of actions may be taken to modify the pattern. In one embodiment, the action taken in response to a recognized pattern is to change one of the attributes of the pattern. In another embodiment, the action taken in response to a recognized pattern is to concatenate a portion of the pattern. In yet another embodiment, the action taken is response to a recognized pattern is to print debugging information. In other embodiments, other actions are taken. Some embodiments take an action with respect to a substring of a token. Some embodiments take a number of actions in response to a recognized pattern. For example, in one embodiment the command "SET the value of <token> to (1:2) <token>" is defined for execution upon recognition of the pattern of an alphabetic class token of length 7 with a substring "EX" in the first two characters. In the example embodiment, the token <EXAMPLE> is recognized as fitting the pattern and the command is executed resulting in the value of the token changing to the first two characters of the original token or "EX". In other embodiments, the value of noise words, such as "at", "by", and "on", which are not typically helpful to a search, are set to zero so that they are excluded from the list of unique index tokens. As shown in FIG. 1A, parsing converts a database record 44 into record tokens (T_{R_n}) 46.

The second step in the process of indexing reference data illustrated in FIG. 2 is to identify (STEP 50) the unique record tokens. Identifying the unique record tokens allows a list of unique record tokens to be created. Such a list may be described as a dictionary of database terms. In one embodiment, certain fields are excluded from contributing to the list. In another embodiment, certain domains are excluded from contributing to the list. In one embodiment, tokens are excluded from contributing to the list of unique tokens based on their class. In another embodiment, tokens are excluded from contributing to the list of unique tokens based on their class and another attribute. In some embodiments, the excluded classes or other attributes are designated with respect to a domain. In some embodiments, the excluded classes or other attributes are designated with respect to records as a whole. For example, in one embodiment, a developer excludes all numeric tokens with a length of more than 5 characters from the list of unique tokens. In another embodiment, STEP 50 is skipped. In yet another embodiment, STEP 50 is done later in the process of indexing reference data illustrated in FIG. 2.

The third step in the process of indexing reference data illustrated in FIG. 2 is to generate (STEP 60) index tokens (T_{I_n}) 62 from record tokens (T_{R_n}) 46. Step 60 is also shown in FIG. 1A. In some embodiments, the index tokens are the record tokens themselves. In the foregoing embodiments, STEP 70 is duplicative of STEP 50. In other embodiments, as shown in FIG. 1A, the index tokens (T_{I_n}) 62 are phonetic equivalents of the record tokens (T_{R_n}) 46. In those embodiment, the index tokens are generated by translating a record token into a phonetic language. In one such embodiment, the phonetic language is NYSIIS. In another such embodiment, the phonetic language is SOUNDEX. In still other such embodiments, the phonetic equivalence is based on another phonetic language or variation thereof. In one embodiment, there are multiple sets of index tokens, each based on different phonetic language or variation thereof. In one embodiment, only record tokens in the alphabetic class are translated and other classes of tokens are not used to generate index tokens. In another embodiment, record tokens in the alphabetic class and other classes generate index tokens, but only the alphabetic portion of the record tokens are translated into index tokens.

The fourth step in the process of indexing reference data illustrated in FIG. 2 is to identify (STEP 70) the unique index tokens. STEP 70 is very similar to STEP 50. Identifying the unique index tokens allows a list of unique index tokens to be created. Such a list may be described as a dictionary of index terms. In one embodiment, certain fields are excluded from contributing to the list. In another embodiment, certain domains are excluded from contributing to the list. In one embodiment, a token is excluded from contributing to the list of unique tokens based on its class. In another embodiment, a token is excluded from contributing to the list of unique tokens based on its class and another attribute. In some embodiments, the excluded classes and attributes are designated with respect to a domain. In some embodiments, the excluded classes and attributes are designated with respect to records as a whole. For example, in one embodiment, a developer excludes all alphabetic tokens with a length of less than 5 characters from contributing to the list of unique tokens. In another embodiment, STEP 70 is skipped. In yet another embodiment, STEP 70 is done after STEP 80. In another embodiment, STEP 70 is done as part of STEP 80.

The fifth step in the process of indexing reference data illustrated in FIG. 2 is to check (STEP 80) for additional records. This step is simply a check step which determines when it is appropriate to calculate the frequency of occurrence of index tokens. If there are additional records, the next record will be processed before this step will be repeated. If there are no

additional records, the indexing process continues on to STEP 90. In one embodiment, the check for additional records comprises simply looking for an end of file flag.

The sixth and final step in the process of indexing reference data illustrated in FIG. 2 is to calculate (STEP 90) the frequency of occurrence of the tokens in the database. Frequency of occurrence is also known as collection frequency or document frequency. Assuming independence of tokens, a lower frequency of occurrence indicates a more selective token. Tokens are not necessarily independent. For example, phrases containing specific groups of tokens may be included repeatedly in a database. Nonetheless, independence of tokens is an acceptable approximation of reality. Frequency of occurrence may be calculated for any type of token that can be associated with a record. For example, in one embodiment, frequency of occurrence is calculated for index tokens. Frequency of occurrence may be calculated for multiple different types of tokens that can be associated with a record. For example, in another embodiment, frequency of occurrence is calculated for index tokens and record tokens.

In one embodiment, a frequency of occurrence is calculated for each unique index token with respect to the database as a whole. In another embodiment, a frequency of occurrence is calculated for each unique index token with respect to each domain in the database. In another embodiment, a frequency of occurrence is calculated for each unique index token with respect to each field in each domain in the database. Other levels of specificity for the calculation are also possible. In some embodiments, no frequency of occurrence is calculated for some unique index tokens. In one embodiment, such index tokens include noise words such as <the> and <and>. Creating a list of index tokens while calculating their respective frequency of occurrence makes the frequency calculation more efficient.

When the frequency of occurrence is calculated, it is efficient to create and save a token table 96 that includes pointers 94 to records containing the token in the respective location in the database. The table 96 prevents duplicative searches for records containing the token from being required. In one embodiment, as shown in FIG. 1A, the pointers 94 are included in a comprehensive table 96. In another embodiment, the pointers are included in a separate table and associated with the respective token.

Referring now to FIG. 3, the figure illustrates a block diagram of query processing according to one embodiment. The first step in processing a query shown in FIG. 3 is to parse (STEP 40) the query. Query parsing can be done using the same process and variations thereto as used for parsing (STEP 40) a record from a database. The only difference is that, whereas

parsing a record 44 results in record tokens (T_{R_n}) 46, parsing a query 54 results in query tokens (T_{Q_n}) 56 as shown in FIG. 1B.

The second step in processing a query as illustrated in FIG. 3 is to expand (STEP 90) the query. In some embodiments, the query is expanded by adding similar tokens to the query tokens. In one such embodiment, similar tokens are selected from the list of unique record tokens. In choosing which tokens in the list of unique record tokens to add to the query tokens, various comparisons of a query token and a candidate record token may be considered. Here, for ease of understanding the list of unique record tokens may be considered a dictionary of database terms. Similarly, the comparisons of a query token and candidate record tokens may be considered a spelling check for the query. In one embodiment, the following comparisons are considered: the number of mismatched characters; the number of transpositions; and the lengths of the character strings. In another embodiment, a subset of the above comparisons are considered. In yet other embodiments, other comparisons are considered instead of or in addition to the named comparisons.

In some embodiments, the entire set of tokens from the list of unique record tokens are used for comparison to a query token. In other embodiments, a smaller subset of tokens from the list of unique record tokens are used for comparison. For example, in one such embodiment, the subset of record tokens that have the same first two characters as the query token are used for comparison with an individual query token. In the example embodiment, if the list of unique record tokens includes no record tokens with the same first two characters as the query token <XENITH>, no further comparison is done and no record token is added to the set of query tokens for the query token <XENITH>.

In embodiments that expand queries by comparing candidate record tokens to a query token, a threshold is set to determine which candidate record tokens are added to the set of query tokens and which are not. In some embodiments, the threshold is based on the similarity of the candidate record tokens in comparison to a query token. In one such embodiment, the threshold is a minimum similarity required for inclusion of the candidate record token. In other embodiments, the threshold is based on the dissimilarity of the candidate record tokens in comparison to a query token. In one such embodiment, the threshold is a maximum dissimilarity required for exclusion of the candidate record token. In another embodiment, the threshold is a combination of the similarity and the dissimilarity.

Various calculations of similarity and dissimilarity are possible depending on the comparisons between the query tokens and record tokens that are used. Similarity may be calculated as follows where each S is a weighting factor, c is the number of characters in common with both the query token and the candidate record token, d is the length of the query token, r is the length of the candidate record token, and t_r is the number of transpositions of characters found by comparing the query token to the candidate record token.

(1) $\text{Similarity} = (S_{cd} * (c / d)) + (S_{rd} * (c / r)) + (S_{tr} * ((c - t_r) / c))$

With respect to the similarity weighting factors S, S_{cd} is the weight factor for the percentage of characters in the query token consisting of characters in common with the candidate record token, S_{rd} is the weight factor for the percentage of characters in the candidate record token consisting of characters in common with the query token, and S_{tr} is the weight factor for the percentage of characters in common with the query token and the candidate record token that are not transposed. In one embodiment, all of the similarity weighting factors are set to a value of 300 and the candidate records are added to the set of query tokens if their calculated similarity exceeds a minimum similarity.

Dissimilarity may be calculated as follows where each D is a weighting factor, u_{cd} is the number of characters in the query token that are not in the candidate record token, d is the length of the query token, u_{rd} is the number of characters in the candidate record token that are not in the query token, r is the length of the candidate record token, t_r is the number of transpositions of characters found by comparing the query token to the candidate record token, and c is the number of letters in common with both the query token and the candidate record token.

(2) $\text{Dissimilarity} = (D_{cd} * (u_{cd} / d)) + (D_{rd} * (u_{rd} / r)) + (D_{tr} * (t_r / c))$

With respect to the dissimilarity weighting factors D, D_{cd} is the penalty factor for the percentage of characters in the query token that are not in the candidate record token, D_{rd} is the penalty factor for the percentage of characters in the candidate record token that are not in the query token, and P_{tr} is the penalty factor for the percentage of characters in common with the query token and the candidate record token that are transposed.

In one embodiment, the query is further expanded by generating search tokens (T_{S_n}) 72 from the query tokens (T_{Q_n}) 56 and the similar tokens. Search token generation can be done using the same process and variation thereto as used for generating (STEP 60) index tokens from record tokens. The only difference is that, whereas index tokens (T_{I_n}) 62 are generated from record tokens (T_{R_n}) 46, search tokens (T_{S_n}) 72 are generated from query tokens (T_{Q_n}) 56.

In another embodiment, as shown in FIG. 1B, the query is expanded by generating search tokens (T_{S_n}) 72 from the query tokens (T_{Q_n}) 56 alone. Again, search token generation can be done using the same process and variation thereto as used for generating (STEP 60) index tokens from record tokens. Again, the only difference is that, whereas index tokens (T_{I_n}) 62 are generated from record tokens (T_{R_n}) 46, search tokens (T_{S_n}) 72 are generated from query tokens (T_{Q_n}) 56.

Referring now to FIG. 4, the figure illustrates a block diagram of the process of accessing the reference data according to one embodiment. The first step in the process of accessing the reference data shown in FIG. 4 is to select (STEP 100) the first search token. In one embodiment, the first search token is selected at random from the search tokens. In another embodiment, the first search token is selected by the given order within the search tokens. In some embodiments, the first search token is the most selective search token. In some embodiments, search tokens are ordered by selectivity. In one such embodiment, selectivity is determined by frequency of occurrence in an indexed database record set. In another such embodiment, selectivity is determined by frequency of occurrence in a specific domain within an indexed database record set. In another such embodiment, selectivity is determined by frequency of occurrence in a specific field in a domain within an indexed database record set. In one embodiment, the first search token is the most selective search token in the domains corresponding to the domains specified in the query. In another embodiment, the most selective search token is identified by comparing frequencies of occurrence reported in a table of unique index tokens.

The second step in the process of accessing the reference data illustrated in FIG. 4 is to access (STEP 110) reference data. In some embodiments, a new search of the database record set for the selected token is initiated. In other embodiments, once the first search token has been selected, the selected token is looked up on a token table. In one such embodiment, as shown in FIG. 1C, the token table 96 will directly return a set of pointers 94 to records within the database containing the selected token (T_{S_3}) 72. In another such embodiment, the token table will indirectly return a set of pointers to records within the database containing the selected token. The pointers may be used to access the records within the database.

The third step in the process of accessing the reference data illustrated in FIG. 4 is to calculate (STEP 120) relevance. In some embodiments, each accessed record is evaluated by calculating a weight representing its likelihood of relevance to the query. In some such

embodiments, the weight is calculated by comparing the query tokens to the record tokens. In another such embodiment, the weight is calculated by comparing the query tokens to the record tokens in the domains specified by the query.

Record linkage is the process of examining records and locating pairs of records that match on some combination of fields. Record Linkage Theory is the probabilistic basis for considering a pair of records to match or be relevant to each other. The present invention applies the Theory in some embodiments to matching a query to individual records within a database record set. A query is defined as a record from the set A of records. A record from the reference data that is a candidate for matching the query is defined as a record from the set B of records. Each pair of records includes one record from set A, in effect the query, and one record from set B. Each pair of records is either a member of the set of matching pairs M or a member of the set of non-matching pairs U.

Under Record Linkage Theory, the power of a field to identify a match depends on the selectivity of the contents of the field and the accuracy of the contents of the field. Selectivity is a measure of the power of the contents of the field to discriminate amongst records. For example, where the field is surnames, the token <Humperdinck> is likely to be much more selective than the token <Smith> because there are likely to be many more records containing <Smith> in the surname field than <Humperdinck>. Selectivity u_i is defined as the probability that two records have the same contents in a field when the pair of records is a member of the set of non-matching pairs U. This is expressed mathematically as follows:

$$u_i = P(fields\_agree \mid \rho \in U).$$

Accuracy is a measure of the reliability of the data in the field. For example, field information which is entered more carefully or checked after entry is more likely to agree in a matched pair than field information which is less carefully entered or not checked after entry. Accuracy m_i is defined as the probability that two records have the same contents in a field when the pair of records is a member of the set of matching pairs M. This is expressed mathematically as follows where $P(\alpha|\beta)$ is the probability of α being true given the condition β:

$$m_i = P(fields\_agree \mid \rho \in M).$$

These measures can be quantified and applied mathematically to predict the likelihood that a record within the reference data is of interest to the user based on the user's query. We consider the pairs of records in which the first record is from the A set of records and the second record is from the B set of records. A and B share a number of common fields. Each pair of

records ρ is a member of the set of matches M or the set of non-matches U. For each pair of records ρ and each domain common to both sets of records i, we define the following quantities:

Agreement Weight W_A is the log of the ratio of the accuracy m_i to the selectivity u_i.

$$(3) \qquad W_A = \log_2\left(\frac{m_i}{u_i}\right)$$

In some embodiments, Agreement Weight W_A is added to the likelihood of relevance of a candidate record when the candidate record contains a token equivalent to the query token in the respective domain i. In other embodiments, Agreement Weight W_A is added to the likelihood of relevance of a candidate record when the candidate record contains a token equivalent to the query token in the respective field i. In other embodiments, i represents another level of specificity of location of data.

Disagreement Weight W_D is the log of the ratio of one minus the accuracy m_i to one minus the selectivity u_i.

$$(4) \qquad W_D = \log_2\left(\frac{(1-m_i)}{(1-u_i)}\right)$$

In some embodiments, Disagreement Weight W_D is subtracted from the likelihood of relevance of a candidate record when the candidate record does not contain a token equivalent to the query token in the respective domain i. In other embodiments, Disagreement Weight W_D is subtracted from the likelihood of relevance of a candidate record when the candidate record does not contain a token equivalent to the query token in the respective field i. In other embodiments, i represents another level of specificity of location of data.

In some embodiments, Adjacency Weight is added to the likelihood of relevance weight of a candidate record if the candidate record contains more than one token equivalent to more than one query token and the relevant record tokens are immediately adjacent to each other. In some embodiments, Semi-Adjacency Weight is added to the likelihood of relevance weight of a candidate record if the candidate record contains more than one token equivalent to more than one query token and the relevant record tokens are located near each other. In one embodiment, Semi-Adjacent Weight is added if search tokens are separated by one intervening token. In other embodiments, Semi-Adjacent is added if search tokens are separated by more than one intervening tokens. In one embodiment, the Adjacency and Semi-Adjacency Weight is a factor of the weights of the relevant search tokens. Various weighting schemes for nearness are available.

In one embodiment, for example, the likelihood of relevance of a candidate record is calculated by summing the Agreement Weight W_A , the Adjacency Weight, and the Semi-Adjacency Weight of all the record tokens in the candidate record with respect to the query tokens. In the example, Semi-Adjacency Weight is only added only when there is one intervening token between the record tokens in the candidate record that are equivalent to query tokens.

The fourth step in the process of accessing the reference data illustrated in FIG. 4 is to compare (STEP 130) the calculated relevance to a threshold. In some embodiments, the weight of each accessed record is compared to one or more thresholds. In other embodiments, the candidate records are ordered by their likelihood of relevance weight so that weights for the set of accessed records are more efficiently compared to one or more thresholds.

In some embodiments, the weight is compared to a continuation threshold. In such an embodiment, the search is terminated if the continuation threshold is exceeded. At that point, all accessed records are output. In such an embodiment, failure to exceed the continuation threshold will trigger (STEP 140) a different search. The token that was used as the basis for the previous search is eliminated from the set of available search tokens. The first step in the new search is to select a different token with which to access reference data. In such an embodiment, if the most selective token has already been used to access data, the second most selective token is used in the subsequent search. The process is repeated until the continuation threshold is exceeded or all search tokens have been used to access data.

In some embodiments, the weight of accessed records is compared to a presentation threshold. In such an embodiment, a portion of the accessed records are output. In embodiments using a presentation threshold, the output records are limited to the those records whose likelihood of relevance weight exceeds the presentation threshold.

In some embodiments, a highest possible likelihood of relevance weight is calculated for each query. The highest possible likelihood of relevance weight depends on the weighting scheme that is selected. In some embodiments, the developer chooses to have additional tokens reduce the weight of a candidate record. For example, in embodiments that use only Agreement Weight W_A, the highest possible likelihood of relevance weight is the weight a candidate record would have if it included every query token in the respective domain. For another example, in embodiments that use Agreement Weight W_A and Adjacency Weight, the highest possible

likelihood of relevance weight is the weight a candidate record would have if it included every query token in the respective domain and in the query arrangement.

In some embodiments, the continuation threshold weight used as a basis for terminating a search is a percentage of the highest possible weight. In other embodiments, the continuation threshold weight is an absolute weight. In some embodiments, the presentation threshold weight used as a criteria for presenting a record accessed in a search is a percentage of the highest possible weight. In other embodiments, the presentation threshold weight is an absolute weight.

In some embodiments, the accessed records are ordered for output by likelihood of relevance weight. In other embodiments, the accessed records are output in the order in which they are retrieved. In still other embodiments, the accessed records are output in another order.

Some embodiments include a step in the database accessing process not shown in the embodiment of FIG. 4. In this step, the amount of information accessed is compared to an overflow threshold. If the overflow threshold is exceeded in such embodiments, the current search is terminated. The memory or buffer is cleared. In one such embodiment, a new search is triggered. The new search is based on all search tokens connected together with a Boolean AND. If the overflow threshold triggers a new search, the continuation threshold is then disabled. Otherwise, the records accessed in the new search are handled the same as in a regular search. In some embodiments, the overflow threshold used as a basis for terminating a search and triggering a different search is as a software error or warning regarding available memory space or buffer space.

Finally, in one embodiment, in addition to the regular search, the developer elects to have a search based on all search tokens connected together with a Boolean AND run for each query.

Having described embodiments of the invention, it will be apparent to those of ordinary skill in the art that other embodiments incorporating the concepts disclosed herein can be used without departing from the spirit and the scope of the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive. Therefore, it is intended that the scope of the present invention be only limited by the following claims.

Claims

What is claimed is:

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

(a) inputting records of a database;

(b) parsing each record into a plurality of record tokens using a pattern action language; and

(c) creating an index to the record from the plurality of record tokens for each record.

2. The method of claim 1 wherein step (b) comprises parsing each record into a plurality of record tokens using a pattern action language, said parsing comprising the steps of:

(i) converting each record into a plurality of original tokens;

(ii) characterizing each original token; and

(iii) converting the plurality of characterized original tokens into said plurality of record tokens based on the pattern action language.

3. The method of claim 2 wherein the pattern action language is responsive to a domain with which each of said plurality of record tokens is associated.

4. The method of claim 1 wherein step (c) comprises creating an index to the record from the plurality of record tokens for each record, said creating comprising the steps of:

(i) creating a list of unique index tokens from the plurality of record tokens for each record;

(ii) calculating a frequency of occurrence in the database for each unique index token; and

(iii) creating a table of index tokens, said table of index tokens containing for each unique index token the frequency of occurrence in the database.

5. The method of claim 4, further comprising, prior to step (c), the step of generating an index token from a respective record token, said index token comprising a phonetic equivalent for the respective record token.

6. The method of claim 5, further comprising the step of creating a list of unique record tokens.

7. A method for indexing a database, the method comprising the steps of:

(a) inputting records of a database;

(b) parsing each record into a plurality of record tokens;

(c) generating an index token from a respective record token, said index token

comprising a phonetic equivalent for the respective record token;

(d) calculating a frequency of occurrence in the database for each unique index token; and

(e) creating a table of index tokens, said table of index tokens comprising, for each unique index token, the frequency of occurrence.

8. The method of claim 7, further comprising the step of creating a list of unique record tokens.

9. The method of claim 8 wherein step (b) comprises parsing each record into a plurality of record tokens using a pattern action language.

10. The method of claim 9 wherein step (b) comprises parsing each record into a plurality of record tokens using a pattern action language, said parsing comprising the steps of:

(i) converting each record into a plurality of original tokens;

(ii) characterizing each original token; and ;

(iii) converting the plurality of characterized original tokens into said plurality of record tokens based on the pattern action language.

11. A method for indexing a database, the method comprising the steps of:

(a) inputting records of a database;

(b) parsing each record into a plurality of record tokens using a pattern action language, each of said plurality of record tokens being associated with a respective domain in the database, said parsing comprising the steps of:

(i) converting each record into a plurality of original tokens;

(ii) characterizing each original token; and

(ii) converting the plurality of characterized original tokens into said plurality of record tokens based on the pattern action language, said pattern action language being responsive to the respective domain with which each of said plurality of record tokens is associated;

(c) creating a list of unique record tokens;

(d) generating an index token from a respective record token, said index token being associated with the domain in the database with which the respective record token is associated, said index token comprising a phonetic equivalent for the respective record token;

(e) creating a list of unique index tokens;

(f) calculating a frequency of occurrence in each domain of the database for each

unique index token;

(g) creating a table of index tokens, said table of index tokens comprising, for each unique index, token the frequency of occurrence in each domain of the database; and

(h) creating an index to the database from the plurality of record tokens for each record.

12. An apparatus for indexing a database, the apparatus comprising:

an input device, said input device accepting records of a database;

a parser in signal communication with the input device, said parser parsing each record of the database into a plurality of record tokens using a pattern action language; and

an indexer in signal communication with the parser, said indexer generating an index of the plurality of record tokens in the database.

13. The apparatus of claim 12, wherein the parser comprises:

a tokenizer in signal communication with the input device, said tokenizer converting each record into a plurality of original tokens;

a token characterizer in signal communication with the tokenizer, said token characterizer characterizing each original token; and

a token converter in signal communication with the token characterizer, said token converter converting a plurality of characterized original tokens into a plurality of record tokens based on the pattern action language.

14. The apparatus of claim 13, wherein said token converter converts a plurality of characterized original tokens into a plurality of record tokens based on the pattern action language, said pattern action language being responsive to a domain with which each of said plurality of record tokens is associated.

15. The apparatus of claim 12, wherein the indexer further comprises:

a token comparator in signal communication with the parser, said token comparator creating a list of unique index tokens from the plurality of record tokens for each record;

a frequency calculator in signal communication with the token comparator, said frequency calculator calculating a frequency of occurrence in the database for each unique index token on the list of unique index tokens; and

a table generator in signal communication with the frequency calculator, said table generator generating a table containing for each unique index token the frequency of occurrence calculated by the frequency calculator.

16. The apparatus of claim 15, the apparatus further comprising:

a token generator in signal communication with the parser, said token generator generating at least one index token for a respective record token, said at least one index token comprising a phonetic equivalent for the respective record token; and

wherein the token comparator is in signal communication with the parser via the token generator.

17. The apparatus of claim 16, the apparatus further comprising:

a record token comparator in signal communication with the parser, said record token comparator creating a list of unique record tokens from the plurality of record tokens for each record.

18. An apparatus for indexing a database, the apparatus comprising:

an input device, said input device accepting records of a database;

a parser in signal communication with the input device, said parser parsing each record of the database into a plurality of record tokens;

a token generator in signal communication with the parser, said token generator generating at least one index token for a respective record token, said at least one index token comprising a phonetic equivalent for the respective record token;

a frequency calculator in signal communication with the token generator, said frequency calculator calculating a frequency of occurrence in the database for each unique index token; and

a table generator in signal communication with the frequency calculator, said table generator generating a table containing for each unique index token the frequency of occurrence calculated by the frequency calculator and a pointer to each record in the database that contains an index token corresponding to said unique index token.

19. The apparatus of claim 18, the apparatus further comprising:

a record token comparator in signal communication with the parser, said record token comparator creating a list of unique record tokens from the plurality of record tokens for each record.

20. The apparatus of claim 18 wherein the parser parses each record of the database into a plurality of record tokens using a pattern action language.

21. The apparatus of claim 20 wherein the parser further comprises:

a tokenizer in signal communication with the input device, said tokenizer converting each record into a plurality of original tokens;

a token characterizer in signal communication with the tokenizer, said token characterizer characterizing each original token; and

a token converter in signal communication with the token characterizer, said token converter converting a plurality of characterized original tokens into a plurality of record tokens based on the pattern action language.

22. An apparatus for indexing a database, the apparatus comprising:

an input device, said input device accepting records of a database;

a parser in signal communication with the input device, said parser parsing each record of the database into a plurality of record tokens using a pattern action language, said parser further comprising:

a tokenizer in signal communication with the input device, said tokenizer converting each record into a plurality of original tokens;

a token characterizer in signal communication with the tokenizer, said token characterizer characterizing each original token; and

a token converter in signal communication with the token characterizer, said token converter converting a plurality of characterized original tokens into a plurality of record tokens based on the pattern action language, said pattern action language being responsive to the respective domain with which each of said plurality of record tokens is associated;

a record token comparator in signal communication with the parser, said record token comparator creating a list of unique record tokens from the plurality of record tokens for each record;

a token generator in signal communication with the parser, said token generator generating at least one index token for a respective record token, said at least one index token comprising a phonetic equivalent for the respective record token;

an index token comparator in signal communication with the token generator, said token comparator creating a list of unique index tokens from the at least one index token for the respective record token;

a frequency calculator in signal communication with the token generator, said frequency calculator calculating a frequency of occurrence in a domain in the database for each unique index token; and

a table generator in signal communication with the frequency calculator, said table generator generating a table containing for each unique index token the frequency of occurrence

calculated by the frequency calculator and a pointer to each record in the database that contains a record token corresponding to said unique index token.

23. A method of querying a database, the method comprising the steps of:

(a) inputting a query;

(b) parsing the query into a plurality of query tokens using a pattern action language;

(c) generating at least one search token from a respective query token; and

(d) looking up at least one search token on an index table to access at least one record within a database.

24. The method of claim 23 wherein step (b) comprises parsing the query into a plurality of query tokens using a pattern action language, said parsing comprising the steps of:

(i) converting the query into a plurality of original tokens;

(ii) characterizing each original token; and

(iii) converting the plurality of characterized original tokens into said plurality of query tokens based on the pattern action language.

25. The method of claim 24 wherein step (b) comprises parsing the query into a plurality of query tokens using a pattern action language, each of said plurality of query tokens being associated with a respective domain in a database, said parsing comprising the steps of:

(i) converting the query into a plurality of original tokens;

(ii) characterizing each original token; and

(iii) converting the plurality of characterized original tokens into said plurality of query tokens based on the pattern action language, said pattern action language being responsive to the respective domain with which each of said plurality of record tokens is associated.

26. The method of claim 24 wherein step (b) comprises parsing the query into a plurality of query tokens using a pattern action language, each of said plurality of query tokens being associated with a respective domain in a database, said parsing comprising the steps of:

(i) converting the query into a plurality of original tokens;

(ii) characterizing each original token; and

(iii) converting the plurality of characterized original tokens into said plurality of query tokens based on the pattern action language; and

wherein step (c) comprises generating at least one search token from a respective query token, each of said at least one search token being associated with the domain in the database

with which the respective query is associated.

27. A method of querying a database, the method comprising the steps of:

(a) inputting a query;

(b) parsing the query into a plurality of query tokens;

(c) generating at least one search token from a respective query token, said generating comprising the steps of:

(i) checking a list of unique record tokens within a database for at least one similar token, said at least one similar token qualifying as similar to the respective query token based on an information theoretic algorithm; and

(ii) translating each respective query token and any similar tokens into said at least one search token, said at least one search token comprising a phonetic equivalent for a respective query token or a similar token; and

(d) looking up at least one search token on an index table to access at least one record within the database.

28. The method of claim 27 wherein step (b) comprises parsing the query into a plurality of query tokens, each of said plurality of query tokens being associated with a respective domain in a database; and

wherein step (c) comprises generating at least one search token from a respective query token, each of said at least one search token being associated with the domain in the database with which the respective query token is associated, said generating comprising the steps of:

(i) checking a list of unique record tokens within a database for at least one similar token, said at least one similar token qualifying as similar to the respective query token based on an information theoretic algorithm; and

(ii) translating each respective query token and any similar tokens into said at least one search token, said at least one search token comprising a phonetic equivalent for a respective query token or a similar token.

29. The method of claim 28 wherein step (b) comprises parsing the query into a plurality of query tokens using a pattern action language, each of said plurality of query tokens being associated with a respective domain in a database.

30. A method of claim 29 wherein step (b) comprises parsing the query into a plurality of query tokens using a pattern action language, each of said plurality of query tokens being associated with a respective domain in a database, said parsing comprising the steps of:

(i) converting the query into a plurality of original tokens;

(ii) characterizing each original token; and

(iii) converting the plurality of characterized original tokens into said plurality of query tokens based on the pattern action language.

31. An apparatus for querying a database, the apparatus comprising:

a query input device;

a parser in signal communication with the query input device, said parser parsing the input to the query input device into a plurality of query tokens using a pattern action language;

a generator in signal communication with the parser, said generator generating at least one search token for a respective query token;

a database accessor in signal communication with the database and the generator, said database accessor accessing records in the database in response to at least one of the plurality of search tokens generated by the generator.

32. The apparatus of claim 31, the parser further comprising:

a tokenizer in signal communication with the query input device, said tokenizer creating a plurality of original tokens from the input to the query input device;

a token characterizer in signal communication with the tokenizer, said token characterizer characterizing each of the original tokens created by the tokenizer; and

a token converter in signal communication with the token characterizer, said token converter converting the plurality of characterized original tokens into said plurality of query tokens based on the pattern action language.

33. The apparatus of claim 32 wherein the tokenizer creates a plurality of original tokens from the input to the query input device, each of said plurality of original tokens being associated with a domain in a database; and

wherein the token converter converts the plurality of characterized original tokens into the plurality of query tokens based on the pattern action language, each of said plurality of query tokens being associated with the domain in the database with which a respective original token is associated, said pattern action language being responsive to the respective domain with which each of said plurality of query tokens is associated.

34. The apparatus of claim 32 wherein the tokenizer creates a plurality of original tokens from the input to the query input device, each of said plurality of original tokens being associated with a domain in a database;

wherein the token converter converts the plurality of characterized original tokens into the plurality of query tokens based on the pattern action language, each of said plurality of query tokens being associated with the domain in the database with which a respective original token is associated; and

wherein the generator generates at least one search token for a respective query token, said search token being associated with the domain in the database with which the respective query token is associated.

35. An apparatus for querying a database, the apparatus comprising:

a query input device;

a parser in signal communication with the query input device, said parser parsing the input to the query input device into a plurality of query tokens;

a generator in signal communication with the parser, said generator generating at least one search token for a respective query token, said generator further comprising:

a query expander in signal communication with the parser, said query expander adding similar tokens that are similar to at least one of the plurality of query tokens based on an information theoretic algorithm; and

a translator in signal communication with the query expander, said translator translating each query token and each similar token output by the query expander into a respective search token, each respective search token comprising a phonetic equivalent for a query token or a similar token; and

a database accessor in signal communication with the database and the generator, said database accessor accessing records in the database in response to at least one respective search token generated by the generator.

36. The apparatus of claim 35 wherein the parser parses the input into a plurality of query tokens, each of the plurality of query tokens being associated with a domain in the database;

wherein the query expander adds similar tokens that are similar to at least one of the plurality of query tokens based on an information theoretic algorithm, each of said similar tokens being associated with the domain in the database with which the at least one of the plurality of query tokens is associated; and

wherein the translator translates each of the plurality of query tokens and each of the similar tokens output by the query expander into a respective search token, each respective search token being associated with the domain in the database with which the respective query

token is associated.

37. The apparatus of claim 36 wherein said parser parses the input to the query input device into a plurality of query tokens using a pattern action language.

38. The apparatus of claim 37 wherein the parser further comprises:

a tokenizer in signal communication with the query input device, said tokenizer converting each query into a plurality of original tokens, each of said plurality of original tokens being associated with a respective domain in a database;

a token characterizer in signal communication with the tokenizer, said token characterizer characterizing each original token; and

a token converter in signal communication with the token characterizer, said token converter converting a plurality of characterized original tokens into a plurality of query tokens based on the pattern action language, each of said plurality of query tokens being associated with the respective domain with which the original token is associated.

39. A method for accessing data within a database, the method comprising the steps of:

(a) selecting a token from a plurality of tokens as a first token on which to search;

(b) retrieving at least one record from the database in response to the selected token;

(c) determining a likelihood of relevance to the query for each of the at least one record;

(d) ordering each of the at least one record by likelihood of relevance to the query;

(e) comparing a continuation threshold to the highest likelihood of relevance to the query for the at least one record, and

(i) if the likelihood of relevance to the query for the at least one record exceeds the continuation threshold, terminating the search; and

(ii) if the continuation threshold exceeds the likelihood of relevance to the query for the at least one record, selecting a different token from the plurality of tokens as a next token on which to search, and repeating steps (b) through (e); and

(f) returning at least one retrieved record.

40. The method of claim 39 wherein step (c) comprises determining a likelihood of relevance to the query for each of the at least one record based on Record Linkage Theory.

41 The method of claim 40 wherein step (b) comprises retrieving a plurality of records from the database in response to the selected token;

wherein step (c) comprises determining a likelihood of relevance to the query for each of

the plurality of records based on Record Linkage Theory;

wherein step (d) comprises ordering each of the plurality of records by likelihood of relevance to the query;

wherein step (e) comprises comparing a continuation threshold to the highest likelihood of relevance to the query for the plurality of records, and

(i) if the likelihood of relevance to the query for the plurality of records exceeds the continuation threshold, terminating the search; and

(ii) if the continuation threshold exceeds the likelihood of relevance to the query for the plurality of records, selecting a different token from the plurality of tokens as the next token on which to search, and repeating steps (b) through (e); and

wherein step (f) comprises returning a plurality of retrieved records, said plurality of retrieved records ordered by likelihood of relevance to the query.

42. The method of claim 39, further comprising, prior to step (a), the steps of:

identifying a frequency of occurrence in a database for each of a plurality of tokens; and

ordering each token by the frequency of occurrence;

wherein step (a) comprises selecting a token from the plurality of tokens as a first token on which to search, said token having the lowest frequency of occurrence; and

wherein step (e) comprises comparing a continuation threshold to the highest likelihood of relevance to the query for the at least one record, and

(i) if the likelihood of relevance to the query for the at least one record exceeds the continuation threshold, terminating the search; or

(ii) if the continuation threshold exceeds the likelihood of relevance to the query for the at least one record, selecting a different token from the plurality of tokens as a next token on which to search, said different token having the next lowest frequency of occurrence, and repeating steps (b) through (e).

43. The method of claim 42, wherein a frequency of occurrence in a database is in a respective domain, further comprising, prior to step (a), the steps of:

identifying the frequency of occurrence in each domain of a database for each of a plurality of tokens, each of said plurality of tokens beings associated with a respective domain in the database; and

ordering each token by frequency of occurrence in each domain of the database;

wherein step (a) comprises selecting a token from the plurality of tokens as a first token

on which to search, said token having the lowest frequency of occurrence in the respective domain; and

wherein step (e) comprises comparing a continuation threshold to the highest likelihood of relevance to the query for the at least one record, and

(i) if the likelihood of relevance to the query for the at least one record exceeds the continuation threshold, terminating the search; and

(ii) if the continuation threshold exceeds the likelihood of relevance to the query for the at least one record, selecting a different token from the plurality of tokens as a next token on which to search, said different token having the next lowest frequency of occurrence in the respective domain, and repeating steps (b) through (e).

44. The method of claim 43 wherein step (c) comprises determining a likelihood of relevance to the query for each record based on Record Linkage Theory.

45. A method of claim 44, further comprising, prior to step (c), the step of checking a buffer of retrieved records for overflow and, if the buffer is overflowing, clearing the buffer and retrieving at least one record from the database, each of the at least one record containing all of the plurality of tokens.

46. An apparatus for accessing data within a database, the apparatus comprising:

a token selector, said token selector selecting a token from a plurality of tokens as a first token on which to search;

a database accessor in signal communication with the token selector and a database, said database accessor retrieving at least one record from the database in response to the selected token;

a relevance determiner in signal communication with the database accessor, said relevance determiner determining a likelihood of relevance to a query for each of the at least one record;

a relevance comparator in signal communication with the relevance determiner, said relevance comparator ordering each of the at least one record by likelihood of relevance to the query;

a threshold comparator in signal communication with the relevance comparator and the token selector, said threshold comparator comparing a continuation threshold to the highest likelihood of relevance to the query for the at least one record and terminating the search if the continuation threshold is exceeded or, if the continuation threshold is not exceeded, removing the

selected token from the plurality of search tokens and inputting the remaining search tokens to the token selector; and

an output device in signal communication with the threshold comparator, said output device returning the at least one retrieved record when the threhold comparator terminates the search.

47. The apparatus of claim 46 wherein the relevance determiner determines a likelihood of relevance to a query for each of the at least one record based on Record Linkage Theory.

48. The apparatus of claim 47 wherein the database accessor retrieves a plurality of records from the database in response to the selected token;

a relevance determiner in signal communication with the database accessor, said relevance determiner determining a likelihood of relevance to a query for each of the plurality of records based on Record Linkage Theory;

a relevance comparator in signal communication with the relevance determiner, said relevance comparator ordering each of the plurality of records by likelihood of relevance to the query;

a threshold comparator in signal communication with the relevance comparator and the token selector, said threshold comparator comparing a continuation threshold to the highest likelihood of relevance to the query for the plurality of records and terminating the search if the continuation threshold is exceeded or, if the continuation threshold is not exceeded, removing the selected token from the plurality of search tokens and inputting the remaining search tokens to the token selector; and

an output device in signal communication with the threshold comparator, said output device returning the plurality of records, ordered by likelihood of relevance to the query, when the threhold comparator terminates the search.

49. The apparatus of claim 46, the apparatus further comprising:

a frequency comparator, said frequency comparator identifying a frequency of occurrence in a database for each of a plurality of tokens and ordering each of the plurality of tokens by the frequency of occurrence; and

wherein the token selector is in signal communication with the frequency comparator, said token selector selecting a token from the plurality of tokens as a first token on which to search, said token having the lowest frequency of occurrence.

50. The apparatus of claim 49 wherein the frequency comparator identifies a frequency of

occurrence in a domain in a database for each of a plurality of tokens and orders each of the plurality of tokens by the frequency of occurrence in a respective domain associated with the token; and

wherein the token selector selects a token from the plurality of tokens as a first token on which to search, said token having the lowest frequency of occurrence in the respective domain associated with the token.

51. The apparatus of claim 50 wherein the relevance determiner determines a likelihood of relevance to a query for each of the at least one record based on Record Linkage Theory.

52. The apparatus of claim 51, the apparatus further comprising:

a buffer overflow arrestor in signal communication with the database accessor, said buffer overflow arrestor checking for a buffer overflow and, if the buffer is exceeded, clearing the buffer and sending an overflow signal to the token selector;

wherein the token selector is in signal communication with the buffer overflow arrestor, said token selector selecting all tokens from the plurality of tokens as the tokens on which to search conjunctively in response to a signal from the buffer overflow arrestor; and

wherein the database accessor retrieves at least one record from the database, each of said at least one record containing all of the plurality of tokens.

Prior Art 1/7

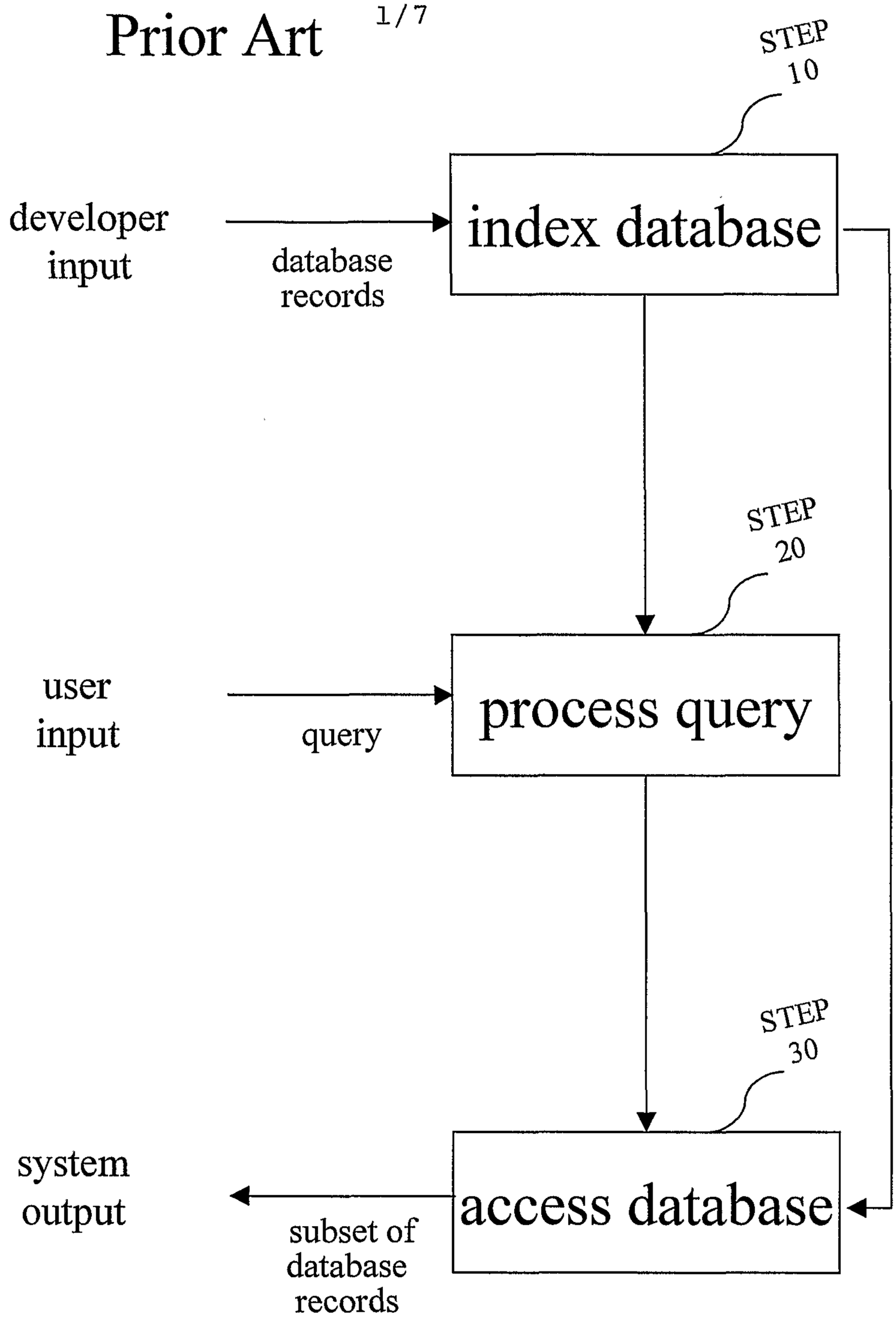

FIG. 1

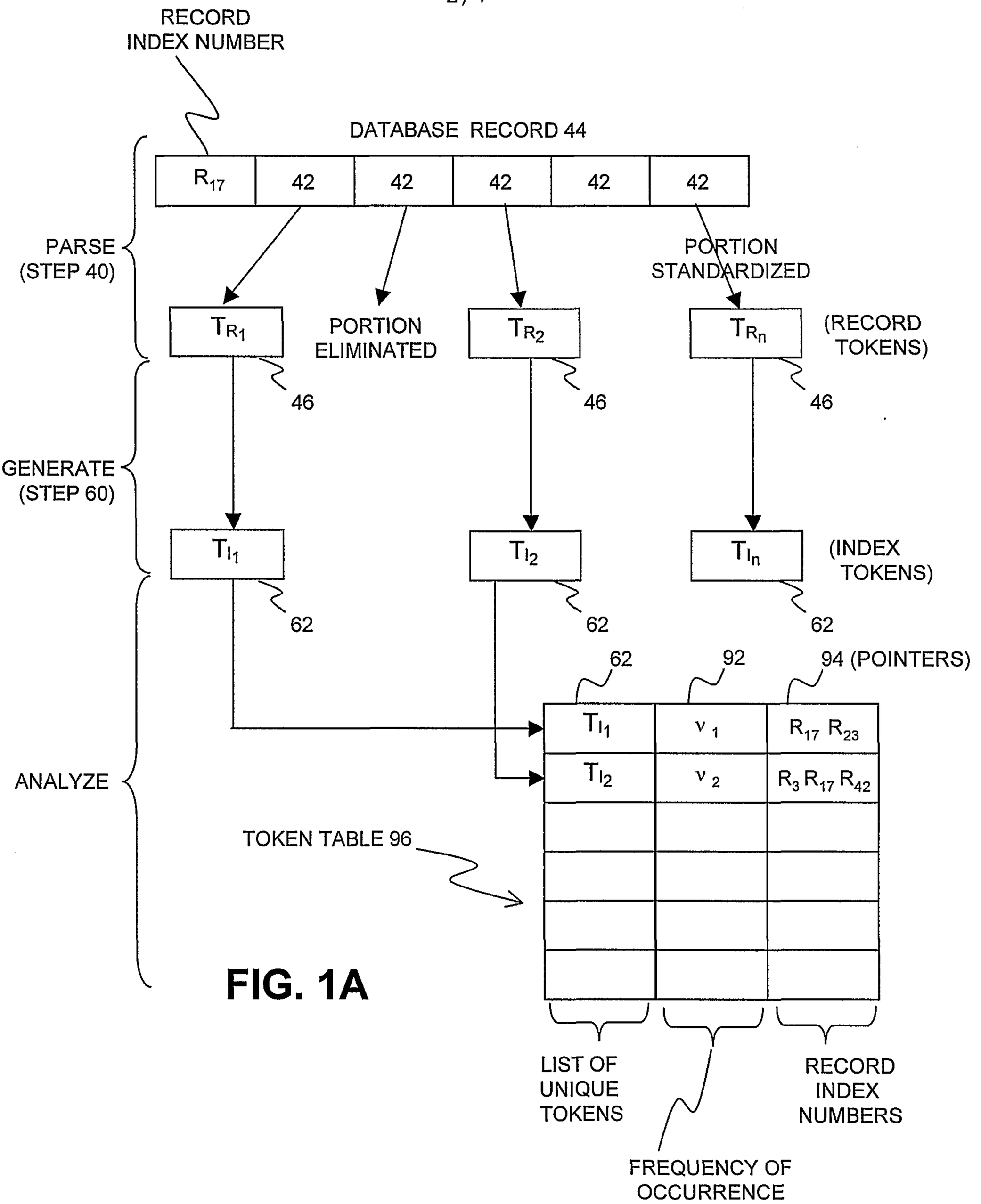
RECORD INDEX NUMBER
DATABASE RECORD 44
R_{17} 42 42 42 42 42
PARSE (STEP 40)
PORTION STANDARDIZED
T_{R_1}
PORTION ELIMINATED
T_{R_2}
T_{R_n}
(RECORD TOKENS)
46
46
46
GENERATE (STEP 60)
T_{I_1}
T_{I_2}
T_{I_n}
(INDEX TOKENS)
62
62
62
ANALYZE
62
92
94 (POINTERS)
T_{I_1} ν_1 R_{17} R_{23}
T_{I_2} ν_2 R_3 R_{17} R_{42}
TOKEN TABLE 96
LIST OF UNIQUE TOKENS
FREQUENCY OF OCCURRENCE
RECORD INDEX NUMBERS

FIG. 1A

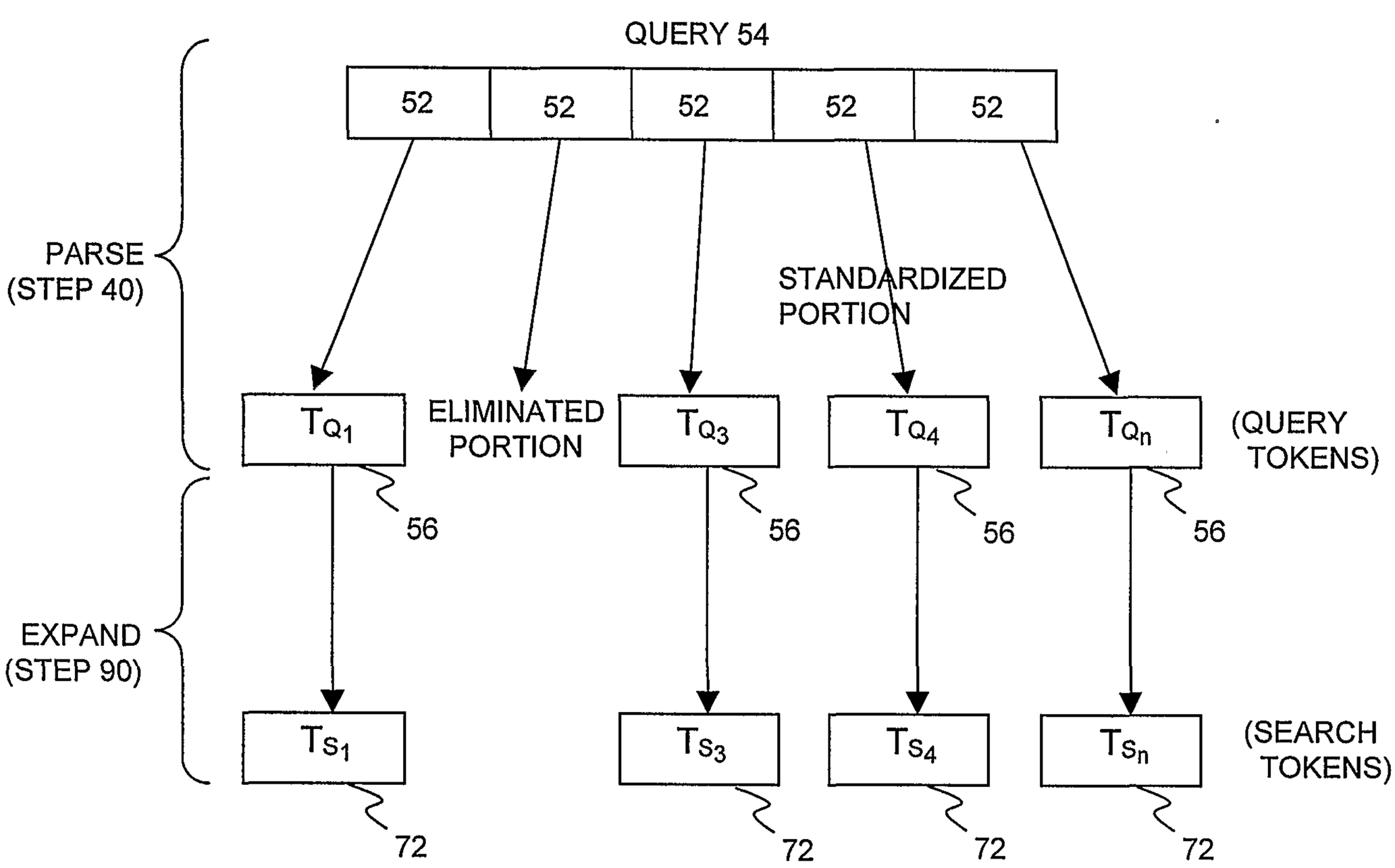
QUERY 54
52
52
52
52
52
PARSE
(STEP 40)
STANDARDIZED
PORTION
T_{Q_1}
ELIMINATED
PORTION
T_{Q_3}
T_{Q_4}
T_{Q_n}
(QUERY
TOKENS)
56
56
56
56
EXPAND
(STEP 90)
T_{S_1}
T_{S_3}
T_{S_4}
T_{S_n}
(SEARCH
TOKENS)
72
72
72
72

FIG. 1B

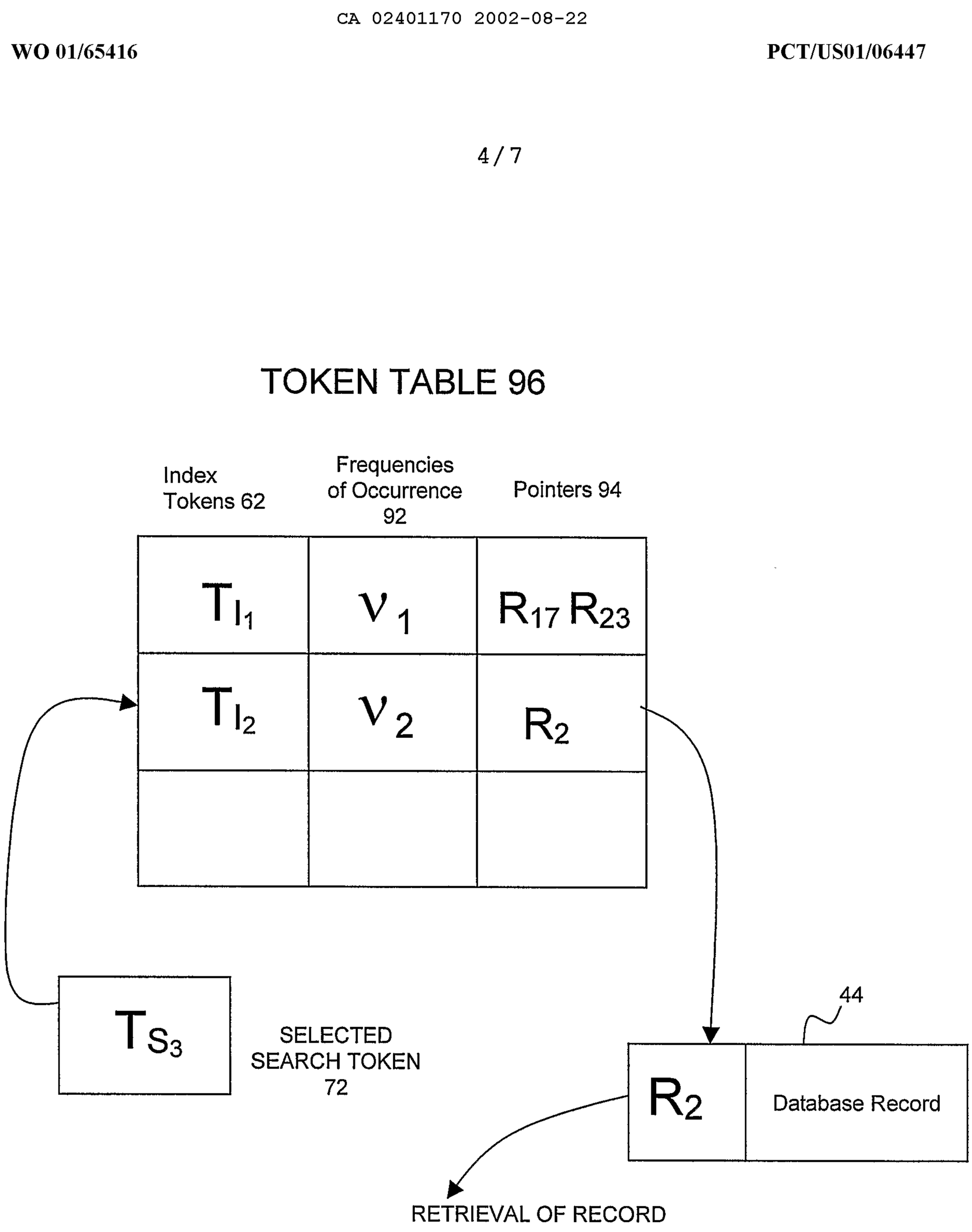
TOKEN TABLE 96
Index Tokens 62
Frequencies of Occurrence 92
Pointers 94
T_{I_1}
ν_1
R_{17} R_{23}
T_{I_2}
ν_2
R_2
T_{S_3}
SELECTED SEARCH TOKEN 72
44
R_2
Database Record
RETRIEVAL OF RECORD

FIG. 1C

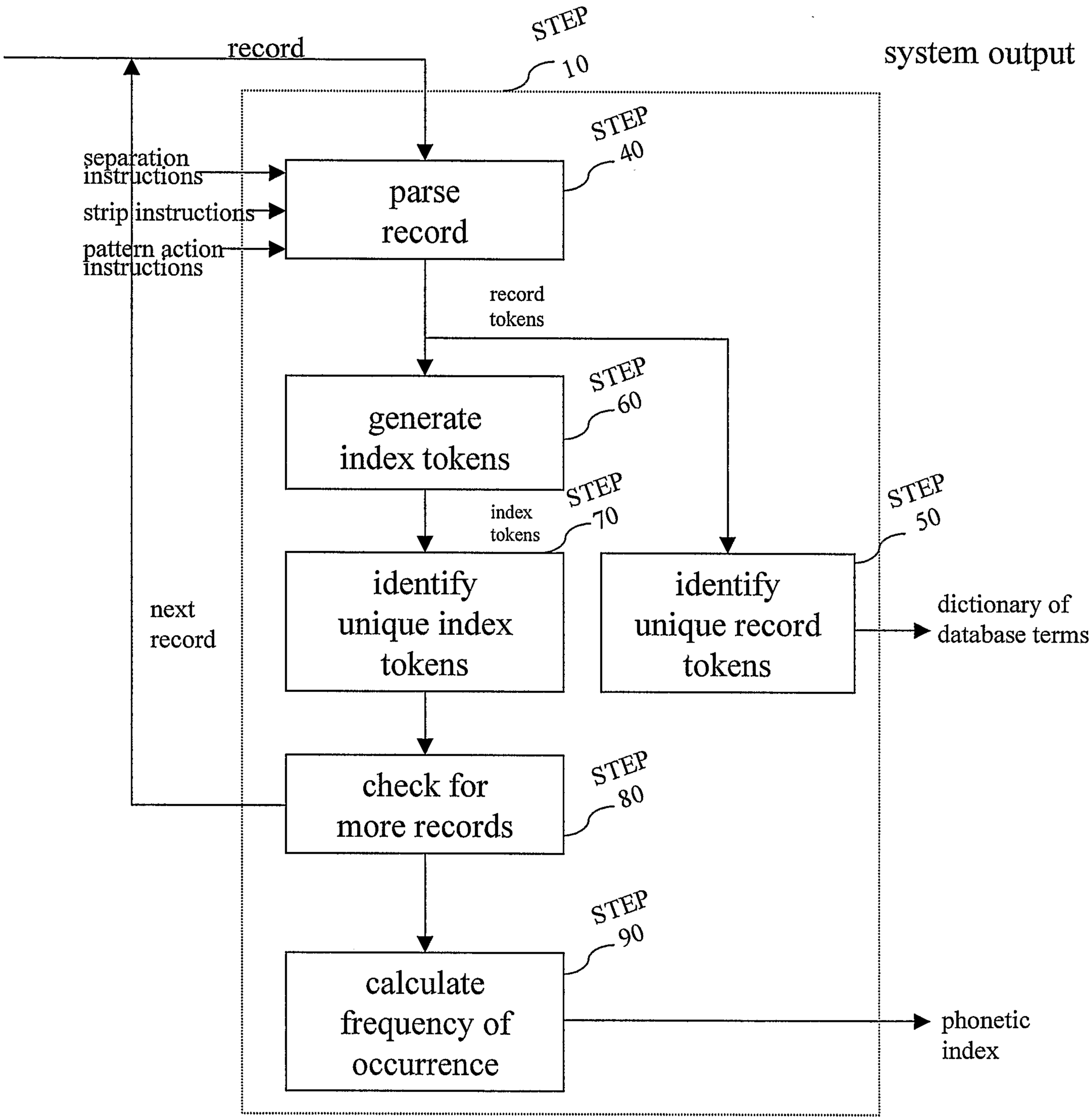
record
STEP 10
system output
separation instructions
strip instructions
pattern action instructions
parse record
STEP 40
record tokens
generate index tokens
STEP 60
index tokens
STEP 70
identify unique index tokens
identify unique record tokens
STEP 50
dictionary of database terms
next record
check for more records
STEP 80
calculate frequency of occurrence
STEP 90
phonetic index

FIG. 2

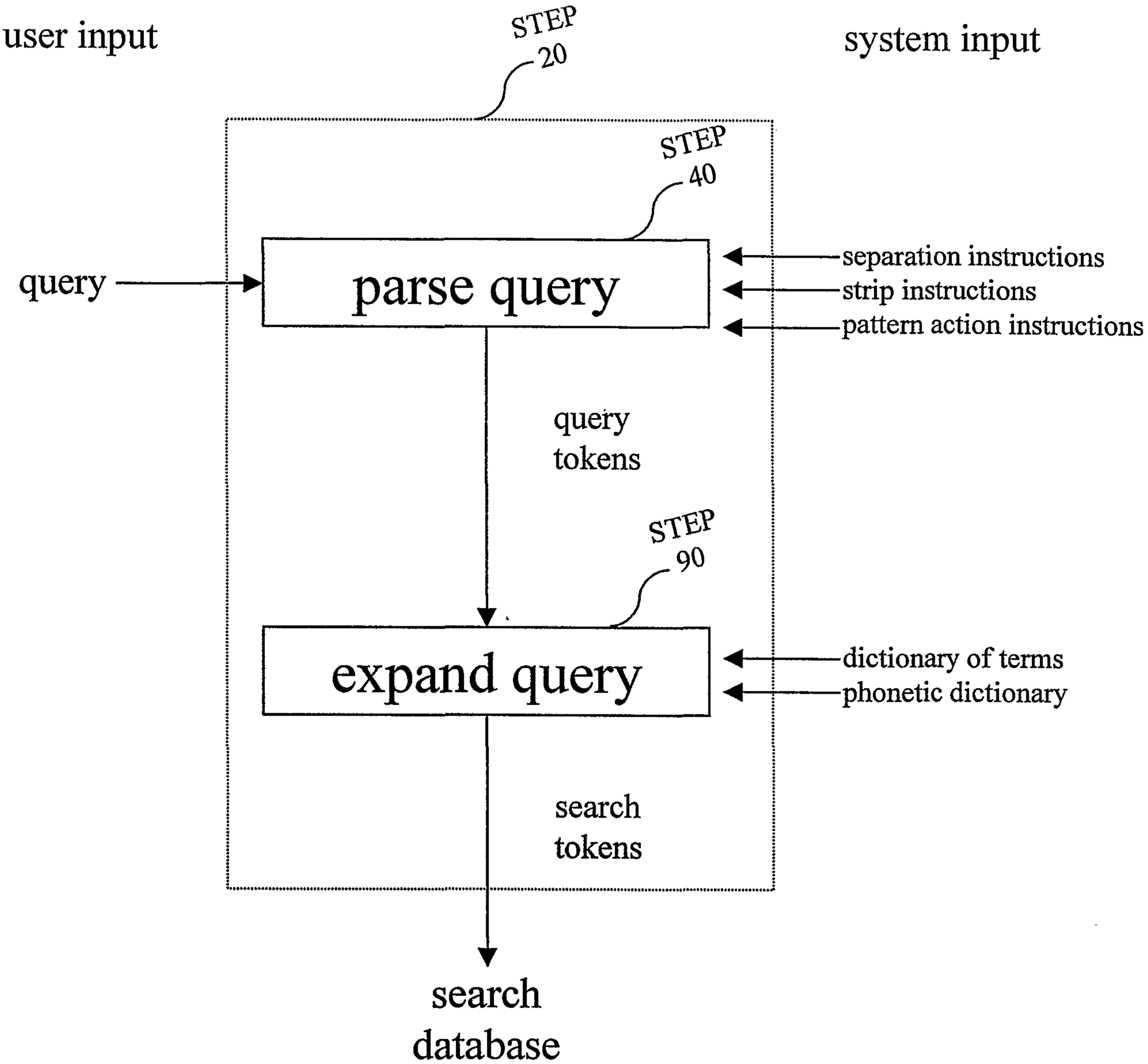
user input
system input
STEP 20
STEP 40
query
parse query
separation instructions
strip instructions
pattern action instructions
query tokens
STEP 90
expand query
dictionary of terms
phonetic dictionary
search tokens
search database

FIG. 3

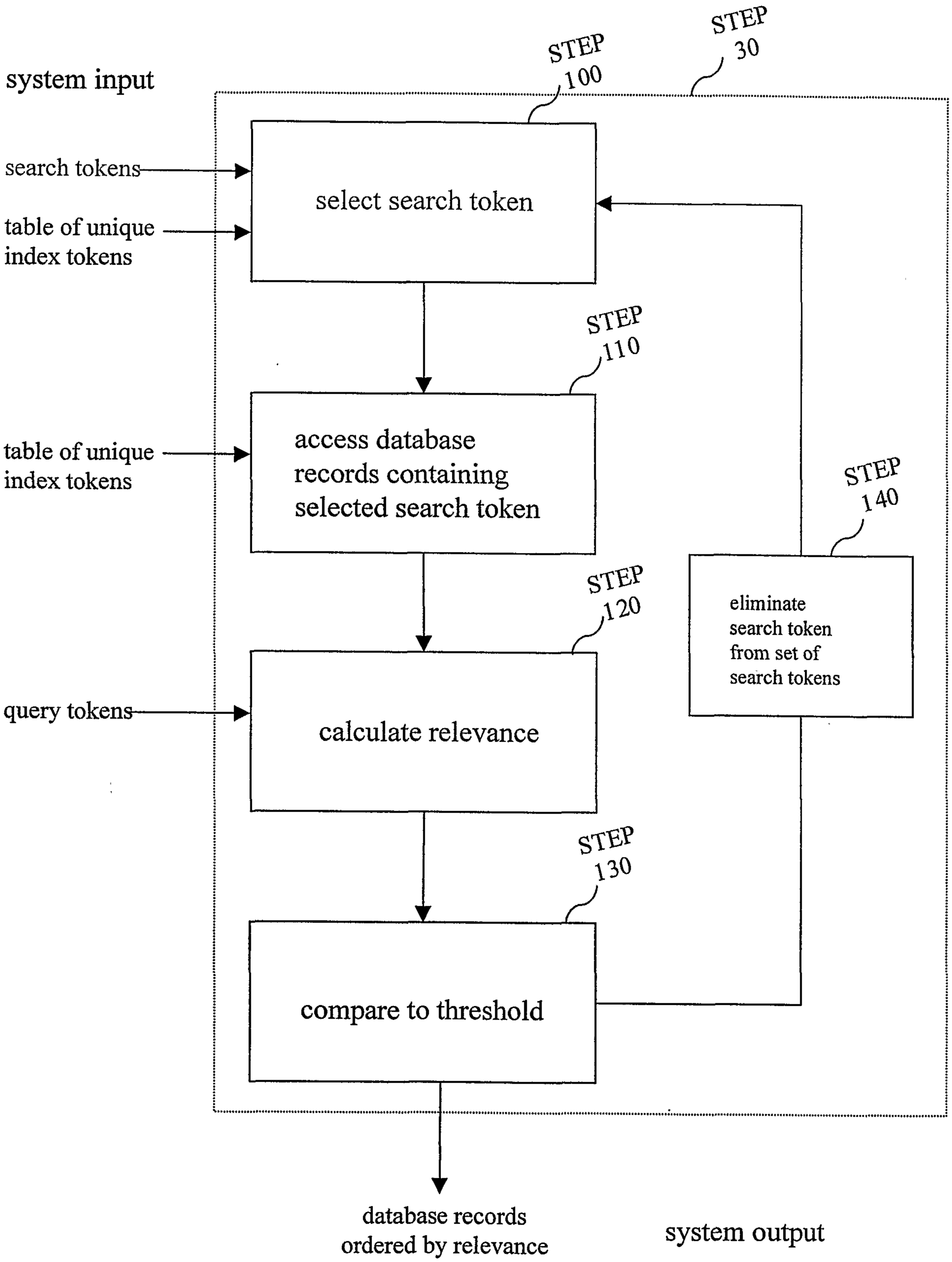
system input
STEP 100
STEP 30
search tokens
table of unique index tokens
select search token
STEP 110
table of unique index tokens
access database records containing selected search token
STEP 120
query tokens
calculate relevance
STEP 130
compare to threshold
STEP 140
eliminate search token from set of search tokens
database records ordered by relevance
system output

FIG. 4