PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 15/40	A1	(11) International Publication Number: WO 93/07577
		(43) International Publication Date: 15 April 1993 (15.04.93)

(21) International Application Number: PCT/US92/08383

(22) International Filing Date: 2 October 1992 (02.10.92)

(30) Priority data:
773,101 8 October 1991 (08.10.91) US

(71) Applicant: WEST PUBLISHING COMPANY [US/US]; 610 Opperman Drive, Egan, MN 55123 (US).

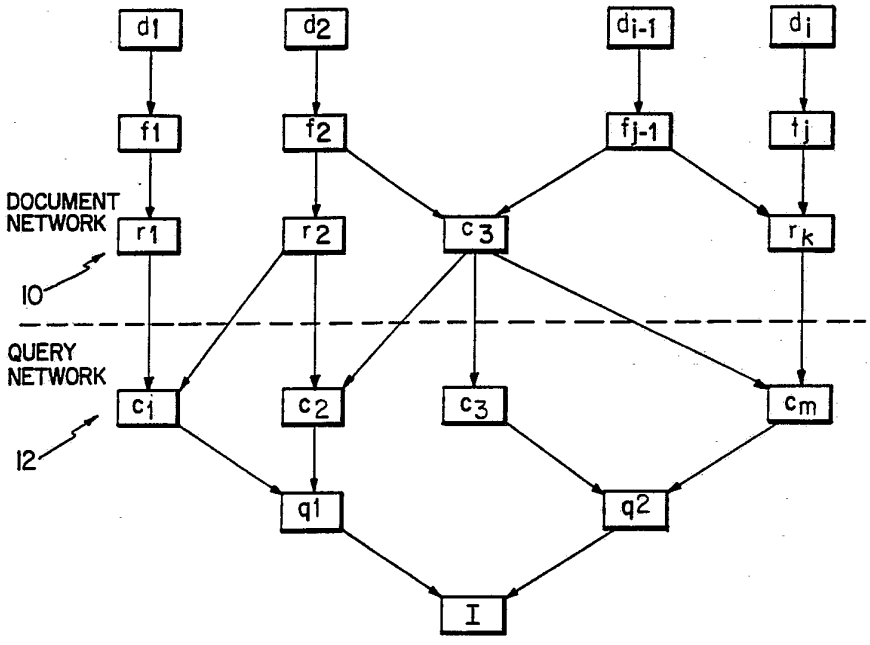
(72) Inventor: TURTLE, Howard, R. ; 2422 Sumac Way, Woodbury, MN 55125 (US).

(74) Agents: ANGUS, Robert, M. et al.; Kinney & Lange, Suite 1500, 625 Fourth Avenue South, Minneapolis, MN 55415 (US).

(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG).

Published
With international search report.

(54) Title: NATURAL LANGUAGE RETRIEVAL SEARCH QUERIES



(57) Abstract

A computer implemented process for creating a search query (50) for a document retrieval system (Fig. 3) in which each word of a natural language input query is compared to a database to remove stopwords therefrom (54). The input query words are stemmed (56) and the sequence of stemmed words is compared to phrases in the database to identify phrases in the search query. Identified phrases are substituted (58) for sequences of stemmed words so that the remaining phrases and stemmed words form the query nodes (q_1, q_2) of the query network for matching to representation nodes (r_1...r_k) of the document network of an inference network.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
MC	Monaco
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SK	Slovak Republic
SN	Senegal
SU	Soviet Union
TD	Chad
TG	Togo
UA	Ukraine
US	United States of America
VN	Viet Nam

NATURAL LANGUAGE RETRIEVAL SEARCH QUERIES

BACKGROUND OF THE INVENTION

This invention relates to information retrieval, and particularly to document retrieval from a computer database. More particularly, the invention concerns a method and apparatus for creating a search query in natural language for use in an inference network for document identification and retrieval purposes.

Presently, document retrieval is most commonly performed through use of Boolean search queries to search the texts of documents in the database. These retrieval systems specify strategies for evaluating documents with respect to a given query by logically comparing search queries to document texts. One of the problems associated with text searching is that for a single natural language description of an information need, different Boolean researchers will formulate different Boolean queries to represent that need. Because the queries are different, document ranking will be different for each search, thereby resulting in different documents being retrieved.

More recently, hypertext databases have been developed which emphasize flexible organizations of multimedia "nodes" through connections made with user-specified links and interfaces which facilitate browsing in the network. Early networks employed query-based retrieval strategies to form a ranked list of candidate "starting points" for hypertext browsing. Some systems employed feedback during browsing to modify the initial query and to locate additional starting points. Network structures employing hypertext databases have used automatically and manually generated links between documents and the concepts or terms that are used to

represent their content. For example, "document clustering" employs links between documents that are automatically generated by comparing similarities of content. Another technique is "citations" wherein documents are linked by comparing similar citations in them. "Term clustering" and "manually-generated thesauri" provide links between terms, but these have not been altogether suitable for document searching on a reliable basis.

Deductive databases have been developed employing facts about the nodes, and current links between the nodes. A simple query in a deductive database, where N is the only free variable in formula W, is of the form {N|W(N)}, which is read as "Retrieve all nodes N such that W(N) can be shown to be true in the current database." However, deductive databases have not been successful in information retrieval. Particularly, uncertainty associated with natural language affects the deductive database, including the facts, the rules, and the query. For example, a specific concept may not be an accurate description of a particular node; some rules may be more certain than others; and some parts of a query may be more important than others. For a more complete description of deductive databases, see Croft et al. "A Retrieval Model for Incorporating Hypertext Links", Hypertext '89 Proceedings, pp 213-224, November 1989 (Association for Computing Machinery), incorporated herein by reference.

A Bayesian network is one which employs nodes to represent the document and the query. If a proposition represented by a parent node directly implies the proposition represented by a child node, a implication line is drawn between the two nodes. If-then rules of Bayesian networks are interpreted as conditional probabilities. Thus, a rule A→B is interpreted as a

probability P(B|A), and the line connecting A with B is logically labeled with a matrix that specifies P(B|A) for all possible combinations of values of the two nodes. The set of matrices pointing to a node characterizes the dependence relationship between that node and the nodes representing propositions naming it as a consequence. For a given set of prior probabilities for roots of the network, the compiled network is used to compute the probability or degree of belief associated with the remaining nodes.

An inference network is one which is based on a plausible or non-deductive inference. One such network employs a Bayesian network, described by Turtle et al. in "Inference Networks for Document Retrieval", SIGIR 90, pp. 1-24 September 1990 (Association for Computing Machinery), incorporated herein by reference. The Bayesian inference network described in the Turtle et al. article comprises a document network and a query network. The document network represents the document collection and employs document nodes, text representation nodes and content representation nodes. A document node corresponds to abstract documents rather than their specific representations, whereas a text representation node corresponds to a specific text representation of the document. A set of content representation nodes corresponds to a single representation technique which has been applied to the documents of the database.

The query network of the Bayesian inference network described in the Turtle et al. article employs an information node identifying the information need, and a plurality of concept nodes corresponding to the concepts that express that information need. A plurality of intermediate query nodes may also be employed where multiple queries are used to express the information requirement.

The Bayesian inference network described in the Turtle et al. article has been quite successful for general purpose databases. However, it has been difficult to formulate the query network to develop nodes which conform to the document network nodes. More particularly, the inference network described in the Turtle et al. article did not use domain-specific knowledge bases to recognize phrases, such as specialized, professional terms, like jargon traditionally associated with specific professions, such as law or medicine.

For a more general discussion concerning inference networks, reference may be made to Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference by J. Pearl, published by Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988, and to Probabilistic Reasoning in Expert Systems by R. E. Neapolitan, John Wiley & Sons, New York, NY, 1990.

Prior techniques for recognizing phrases in an input query employed syntactic and statistical analysis and manual selection techniques. The present invention employs an automated domain-specific knowledge based system to recognize phrases.

SUMMARY OF THE INVENTION

The present invention provides a computer implemented process performing a search query in which a database is provided containing domain-knowledge specific phrases. A natural language input query is inputted to the computer system, the input query defining the composition of the text of documents sought to be identified. The natural language query is parsed and the stopwords are removed. The remaining words of the input query are stemmed to their basic roots, and the sequence of stemmed words in the list is compared to domain-specific phrases in the database to identify phrases in

the search query. The phrases from the database are substituted for the sequence of stemmed words from the list so that the remaining elements, namely, the substituted phrases and unsubstituted stemmed words, form the search query. More particularly, the individual terms of the completed search query form the query nodes of the query network.

An optional and desirable feature of the present invention resides in the provision of a technique for handling citations as a syntactic phrase, the citations being employed for a "weighting" of the statistical probability algorithms of the inference network.

Another optional and desirable feature of the present invention resides in the provision of determining a key number based on a topical database, the key number being added to the search query as a query node and affecting the statistical probability algorithms of the inference network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram representation of a Bayesian inference network with which the present invention is used.

Figure 2 is a block diagram representation of a simplified Bayesian inference network as in Figure 1.

Figure 3 is a block diagram of a computer system for carrying out the invention.

Figure 4 is a flowchart and example illustrating the steps of creating a search query in accordance with the preferred embodiment of the present invention.

Figure 5 is a flowchart and example of the steps for determining a key number for inclusion in the search query described in connection with Figure 4.

Figures 6A-6D are block diagram representations of a simplified Bayesian inference network illustrating different techniques for handling phrases.

Figure 7 is a flowchart illustrating the manner by which partial phrases are handled in a document retrieval system.

Figure 8 is a detailed flowchart of the query network in accordance with the presently preferred embodiment of the present invention.

Figure 9 is a detailed flowchart of a topic and key subroutine used in the query network illustrated in Figure 8.

Figure 10 is a detailed flowchart of a document network used with the query network shown in Figure 8.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The Inference Network

Inference networks employ a predictive probability scheme in which parent nodes provide support for their children. Thus, the degree to which belief exists in a proposition depends on the degree to which belief exists in the propositions which potentially caused it. This is distinct from a diagnostic probability scheme in which the children provide support for their parents, that is belief in the potential causes of a proposition increases with belief in the proposition. In either case, the propagation of probabilities through the network is done using information passed between adjacent nodes.

Figure 1 illustrates a Bayesian inference network as described in the aforementioned Turtle et al. article. The Bayesian network shown in Figure 1 is a directed, acyclic dependency graph in which nodes represent propositional variables or constraints and the arcs represent dependence relations between propositions. An arc between nodes represents that the parent node "causes" or implies the proposition represented by the child node. The child node contains a link matrix or tensor which specifies the probability that the child node is caused by any combination of the parent nodes.

Where a node has multiple parents, the link matrix specifies the dependence of that child node on the set of parents and characterizes the dependence relationship between the node and all nodes representing its potential causes. Thus, for all nodes there exists an estimate of the probability that the node takes on a value given any set of values for its parent nodes. If a node a has a set of parents $\pi_a=\{p_1,\ldots p_n\}$, the estimated probabilities $P(a|p_1,\ldots p_n)$ are determined.

The inference network is graphically illustrated in Figure 1 and consists of two component networks: a document network 10 and a query network 12. The document network consists of document nodes d_1, $d_2,\ldots d_{i-1}$, d_i, interior text representation nodes t_1, $t_2,\ldots t_{j-1}$, t_j, and leaf nodes r_1, r_2, $r_3,\ldots r_k$. The document nodes d correspond to abstract documents rather than their physical representations. The interior nodes t are text representation nodes which correspond to specific text representations within a document. The present invention will be described in connection with the text content of documents, but it is understood that the network can support document nodes with multiple children representing additional component types, such as audio, video, etc. Similarly, while a single text may be shared by more than one document, such as journal articles that appear in both serial issue and reprint collections, and parent/divisional patent specifications, the present invention shall be described in connection with a single text for each document. Therefore, for simplicity, the present invention shall assume a one-to-one correspondence between documents and texts.

The leaf nodes r are content representation nodes. There are several subsets of content representation nodes r_1, r_2, $r_3,\ldots r_k$, each corresponding to a single representation technique which has been applied to the

document texts. If a document collection has been indexed employing automatic phrase extraction and manually assigned index terms, then the set of representation nodes will consist of distinct subsets or content representation types with disjoint domains. For example, if the phrase "independent contractor" has been extracted and "independent contractor" has been manually assigned as an index term, then two content representation nodes with distinct meanings will be created, one corresponding to the event that "independent contractor" has been automatically extracted from the subset of the collection, and the other corresponding to the event that "independent contractor" has been manually assigned to a subset of the collection. As will become clear hereinafter, some concept representation nodes may be created based on the content of the query network.

Each document node has a prior probability associated with it that describes the probability of observing that document. The document node probability will be equal to 1/(collection size) and will be small for most document collections. Each text node contains a specification of its dependence upon its parent. By assumption, this dependence is complete (t_i is true) when its parent document is observed (d_i is true). Each representation node contains a specification of the conditional probability associated with the node given its set of parent text nodes. The representation node incorporates the effect of any indexing weights (for example, term frequency in each parent text) or term weights (inverse document frequency) associated with the concept.

The query network 12 is an "inverted" directed acyclic graph with a single node I which corresponds to an information need. The root nodes c_1, c_2, c_3, . . . c_m are the primitive concepts nodes used to express the

information requirement. A query concept node, c, contains the specification of the probabilistic dependence of the query concept on its set of parent representation content nodes, r. The query concept nodes $c_1 \ldots c_m$ define the mapping between the concepts used to represent the document collection and the concepts that make up the queries. A single concept node may have more than one parent representation node. For example, concept node c_2 may represent the query concept "independent contractor" and have as its parents representation nodes r_2 and r_3 which correspond to "independent contractor" as a phrase and as a manually assigned term.

Nodes q_1, q_2 are query nodes representing distinct query representations corresponding to the event that the individual query representation is satisfied. Each query node contains a specification of the query on the query concept it contains. The intermediate query nodes are used in those cases where multiple query representations express the information need I.

As shown in Figure 1, there is a one-to-one correspondence between document nodes, d, and text nodes, t. Consequently, the network representation of Figure 1 may be diagrammatically reduced so that the document nodes d_1, $d_2, \ldots d_{i-1}$, d_i are parents to the representation nodes r_1, r_2, $r_3, \ldots r_k$. In practice, it is possible to further reduce the network of Figure 1 due to an assumed one-to-one correspondence between the representation nodes r_1, r_2, $r_3, \ldots r_k$, and the concept nodes c_1, c_2, $c_3, \ldots c_m$. The simplified inference network is illustrated in Figure 2 and is more particularly described in the article by Turtle et al., "Efficient Probabilistic Inference for Text Retrieval," RIAO 91 Conference Proceedings, pp. 644-661, April, 1991 (Recherche d'Informaion Assistée par Ordinateur,

Universitat Autònoma de Barcelona, Spain), which article is herein incorporated by reference.

As described above, each child node carries a probability that the child node is caused by the parent node. The estimates of the dependence of a child node Q on its set of parents, P_1, P_2,...P_n, are encoded using the following expressions:

$$bel_{or}(Q) = 1 - (1-p_1)\cdot(1-p_2)\cdot \ldots \cdot (1-p_n) \quad \text{EQ 1}$$

$$bel_{and}(Q) = p_i(p_2P_2\cdot\ldots\cdot\ldots p_0n \quad \text{EQ 2}$$

$$bel_{not}(Q) = 1-p_1 \quad \text{EQ 3}$$

$$bel_{wtd\text{-}sum}(Q) = \frac{(w_1p_1 + w_2p_2 + \ldots + w_np_n)\,w_g}{w_1 + w_2 + w_2 + \ldots w_n} \quad \text{EQ 4}$$

where $P(P_1=true)=p_1$, $P(P_2=true)=p_2$,...$P(P_n=true)=p_n$, w_1, w_2,...w_n are the term weights for each term P_1, P_2,...P_n, and w_g is the maximum probability that the child node can achieve, $0 \leq w_g \leq 1$.

As described above, all child nodes carry a probability that the child was caused by the identified parent nodes. Document network 10 is created once and remains constant for each document in the network. The document network structure is not changed, except to add documents to the database. The document nodes d and text nodes t do not change for any given document once the document representation has been entered into document network 10. Most representation nodes are created with the database and are dependant on the document content. Some representation nodes (representing phrases and the like) are created for the particular search being

conducted and are dependant of the search query.

Query network 12, on the other hand, changes for each input query defining a document request. Therefore, the concept nodes c of the search network are created with each search query and provide support to the query nodes q and the information need, node I (Figure 1).

Document searching can be accomplished by a document-based scan or a concept-based scan. A document-based scan is one wherein the text of each document is scanned to determine the likelihood that the document meets the information need, I. More particularly, the representation nodes r_1, r_2, r_3,...r_k of a single document are evaluated with respect to the several query nodes q_1, q_2 to determine a probability that the document meets the information need. The top n-ranked documents are then selected as potential information need documents. The scan process reaches a point, for example after assigning a probability for more than n documents of a large document collection, that documents can be eliminated from the evaluation process after evaluating subsets of the representation nodes. More particularly, if a given document scores so low of a probability after only evaluating one or two representation nodes, determination can be made that even if the evaluation continued the document still would not score in the top n-ranked documents. Hence, most documents of a large collection are discarded from consideration without having all their representation nodes evaluated.

A concept-based scan is one wherein all documents containing a given representation node are evaluated. As the process continues through several representation nodes, a scorecard is maintained of the probabilities that each document meets the information need, I. More particularly, a single representation node r_1 is evaluated for each document in the collection to assign

an initial probability that the document meets the concept. The process continues through the several representation nodes with the probabilities being updated with each iteration. The top n-ranked documents are then selected as potential information need documents. If at some point in the process it can be determined that evaluation of additional representation nodes will not alter the ranking of the top n-ranked documents, the scan process can be terminated.

It can be appreciated that the representation nodes r_1, r_2, r_3,...r_k are nodes dependent on the content of the texts of the documents in the collection. Most representation nodes are created in the document database. Other representation nodes, namely those associated with phrases, synonyms and citations, are not manifest in any static physical embodiment and are created based on each search query. For example, a query manifesting the concept "employee" may be represented by one or more of "actor", "agent", "attendant", "craftsman", "doer", "laborer", "maid", "servant", "smith", "technician" and "worker", to name a few. These various representation nodes are created from the query node at the time of the search, such as through the use of thesauri and other tools to be described. A query node q_1, q_2, etc. can be manifest in one or more representations.

The Search Query

The present invention concerns development of the concept nodes c for use in the inference network illustrated in Figure 1. The invention will be described in connection with a specific search query as follows:

> "What is the liability of the United States under the Federal Tort Claims Act for injuries sustained by employees of an independent contractor working under contract with an agency of the United States government?"

Thus the present invention will be described in connection with a database for searching legal documents, but it is to be understood the concepts of the invention may be applied to other professional databases, such as medical, theological, financial and other types where specialized vocabularies, citations and digests are employed.

The present invention is carried out through use of a computer system, such as illustrated in Figure 3 comprising a computer 20 connected to an input/output terminal 22 and a read only memory (ROM) 24. ROM 24 may be any form of read only memory, such as a CD ROM, write protected magnetic disc or tape, or a ROM, PROM or EPROM chip encoded for the purposes described. Computer 20 may be a personal computer (PC) and may be optionally connected through modem 26, telephone communication network 28 and modem 30 to a central computer 32 having a memory 34. In one form of the invention, the document network 10 and the document database containing the texts of documents represented by the document network are contained in the central computer 32 and its associated memory 34. Alternatively, the entire network and database may be resident in the memory of personal computer 20 and ROM 24. In a legal database and document information retrieval network the documents may comprise, for example, decisions and orders of courts and government agencies, rules, statutes and other documents reflecting legal precedent. By maintaining the document database and document network at a central location, legal researchers may input documents into the document database in a uniform manner. Thus, there may be a plurality of computers 20, each having individual ROMs 24 and input/output devices 22, the computers 20 being linked to central computer 32 in a time-sharing mode. The search query is developed by each individual user or

researcher by input via the respective input/output terminal 22. For example, input/output terminal 22 may comprise the input keyboard and display unit of PC computer 20 and may include a printer for printing the display and/or document texts.

ROM 24 contains a database containing phrases unique to the specific profession to which the documents being searched are related. In a legal search and retrieval system as described herein, the database on ROM 24 contains stemmed phrases from common legal sources such as ***Black's*** or ***Statsky's Law Dictionary***, as well as common names for statutes, regulations and government agencies. ROM 24 may also contain a database of stopwords comprising words of indefinite direction which may be ignored for purposes of developing the concept nodes of the search query. For example, stopwords included in the database on ROM 24 may include prepositions, such as "of", "under", "above", "for", "with", etc., indefinite articles such as "a" and "the", indefinite verbs such as "is", "are", "be", etc. and indefinite adverbs such as "what", "why", "who", etc. The database on ROM 24 may also include a topic and key database such as the numerical keys associated with the well-known West Key Digest system.

Figure 4 is a flow diagram illustrating the process steps and the operation on the example given above in the development of the concept nodes c. The natural language query is provided by input through input terminal 22 to computer 20. In the example shown in Figure 4, the natural language input query is:

> "What is the liability of the United States under the Federal Tort Claims Act for injuries sustained by employees of an independent contractor working under contract with an agency of the United States government?"

By way of example, a corresponding WESTLAW Boolean

query might be:

"UNITED STATES" U.S. GOVERNMENT (FEDERAL /2 GOVERNMENT) /P TORT /2 CLAIM /P INJUR! /P EMPLOYEE WORKER CREWMAN CREWMEMBER /P INDEPENDENT /2 CONTRACTOR.

As shown in Figure 4, the natural language query shown in block 40 is inputted at step 50 to computer 20 via input/output terminal 22. The individual words of the natural language query are parsed into a list of words at step 50, and at step 54 each word is compared to the stopwords of the database in ROM 24. The stopwords "what", "is", "the", "of", "under", "for", "by", "an" and "with" are removed from the list. More particularly, the stopwords from the database are loaded into a hash table, and a hash function, h, is determined for each word of the natural language query. If element h of the table is null, the query word is not a stopword and the routine is exited. If the hth element is not a null, it points to a stored stopword. The query term is compared to the stopword, and if a match is determined, the query word is deleted from the query and the routine is exited. If no match is determined, a fixed (prime) value is added to h modulo to the table size and determine again whether the hth element of the table is a null, repeating the process until the query term matches a stopword or a null entry is found.

At step 56, the remaining words are stemmed to reduce each word to its correct morphological root. One suitable software routine for stemming the words is described by Porter "An Algorithm for Suffix Stripping", Program, Vol. 14, pp 130-137 (1980). As a result of step 56 a list of words is developed as shown in block 42, the list comprising the stems of all words in the query, except the stopwords.

Phrases

Previous systems recognized linguistic structure

(for example, phrases) by statistical or syntactic techniques. Phrases are recognized using statistical techniques based on the occurrence of phrases in the document collection itself; thus, proximity, co-occurrence, etc. were used. Phrases are recognized using syntactic techniques based on the word/term structure and grammatical rules, rather than statistically. Thus, the phrase "independent contractor" could be recognized statistically by the proximity of the two words and the prior knowledge that the two words often appeared together in documents. The same term could be recognized syntactically by noting the adjective form "independent" and the noun form "contractor" and matching the words using noun phrase grammatical rules. (Manual selection systems have also been used wherein the researcher manually recognizes a phrase during input.)

Previous inference networks employed a two-term logical AND modeled as the product of the beliefs for the individual terms. Beliefs (probabilities) lie in the range between 0 and 1, with 0 representing certainty that the proposition is false and 1 representing certainty that the proposition is true. The belief assigned to a phrase is ordinarily lower than that assigned to either component term. However, experiments reveal that the presence of phrases represents a belief higher than the belief associated with either component term. Consequently, separately identifying phrases as independent representation nodes significantly increases the performance of the information retrieval system. However, single terms of an original query are retained because many of the concepts contained in the original query are not described by phrases. Experimentation has suggested that eliminating single terms significantly degrades retrieval performance even though not all single terms from an original query are required for effective retrieval.

As previously described, the phrase relationships in the search query are recognized by domain-knowledge based techniques (e.g., the phrase database), and by syntactic relationships. The primary reason to solely select syntactical and domain-based phrases for purposes of the query network is to reduce user involvement in identifying phrases for purposes of creating a query.

The present invention employs a domain-knowledge based system wherein the candidate phrases are stored in a database and the individual terms of the query are compared to the database of phrases to locate phrases in the query. An example of a domain-knowledge database is a database containing phrases from a professional dictionary. This type of phrase handling is particularly suitable for professional information retrieval where specialized phrases are often employed.

At step 58 in Figure 4, computer 20 returns to the database in ROM 24 to determine the presence of phrases within the parsed and stemmed list 42. The phrase database in ROM 24 comprises professional, domain-specific phrases (such as from *Black's Law Dictionary*) which have had stopwords removed therefrom and which have been stemmed in accordance with the same procedure for stemming the words of a search query. Computer 20 compares the first and second words of list 42 to the database of phrases in ROM 24 to find any phrase having at least those two words as the first words of a phrase. Thus, comparing the first two terms "LIABL" and "UNIT" to the database of phrases (such as *Black's Law Dictionary*), no match is found. Thus, as shown in block 44, "LIABIL" is retained for the search query. The next two words "UNIT" and "STATE" are compared to the database of phrases and found to correspond to a phrase therein. The next word "FEDER" is then compared to the database to determine if it corresponds to the third word of any phrase commencing with "UNIT STATE". In this case no

phrase is found, so both "UNIT" and "STATE" are removed from the list 44 and substituted with a phrase representing the term "UNIT STATE". The terms "FEDER" and "TORT" are compared to the database and found to compare to phrases in the database. The third and fourth words "CLAIM" and "ACT" also compare to at least one phrase commencing with "FED" and "TORT". Consequently, each of the terms "FEDER", "TORT", "CLAIM" and "ACT" are substituted with the phrase "FEDER TORT CLAIM ACT". The process continues to substitute phrases from the database for sequences of stemmed words from the parsed list 42, thereby deriving the list 44.

As shown at reference numeral 60, the original natural language query contained the phrase UNITED STATES after "AGENCY". However, as described below, duplicate words and phrases are eliminated. Consequently, the stem words "UNIT" and "STATE" do not appear after "AGENCY" in the parsed list of stem words shown in block 42. Consequently, the natural language phrase "AGENCY OF THE UNITED STATES" became a phrase in search query 44 comprising the words "AGENC UNIT STATE".

The phrase lookup is accomplished one word at a time. The current word and next word are concatenated and used as a key for the phrase database query. If a record with the key is found, the possible phrases stored under this key are compared to the next word(s) of the query. As each phrase is found, a record of the displacement and length of each found phrase is recorded.

As indicated above, once successive terms have been identified as a phrase, the individual terms do not appear in the completed query shown at block 44 in Figure 4. In rare cases two phrases might seemingly overlap (i.e., share one or more of the same words). In such a case, the common word is not repeated for each phrase, but instead preference in the overlap is accorded to the

longer phrase. For example, if a natural language search query contained "vertical restraints of trade is the dispute", the parsed and stemmed list (with stopwords removed) would appear as: "vertic", "restrain", "trad", "disput". The database would identify two possible phrases: "vertic restrain trad" and "trad disput", with "trad" overlapping in both phrases. With preference accorded to the longest possible phrase, the query nodes would represent (1) "vertic restrain trad" and (2) the single word "disput".

Topic and Key Database

One optional and desirable feature of the present invention resides in the inclusion of a topic and key database, stored in ROM 24 (Figure 3) to enhance the search query. The topic and key database includes a plurality of topical definitions together with unique key numbers associated therewith. As one example, the topic and key database includes the definitions and key digest numbering system of the well-known West Key Digest System from West Publishing Company of St. Paul, Minnesota. In the West system, headnotes represent digests of the texts and reasoning of judicial decisions. The headnotes are keyed to a numbering system so that like numbers concern like matters.

In accordance with the present invention, as judicial decisions are entered into the document network, the West key numbers associated with the headnotes are included in the document as identifiers of the contents of the decisions. The key numbers thus become document nodes for information retrieval purposes. As a search query is developed, the user may include the key numbers of the topics to be searched, or the key numbers may be generated from the database in ROM 24 or memory 34 for inclusion in the search query.

Figure 5 illustrates the process for generating the

key numbers for inclusion in the search query. At step 62, the key is located in the topic and key database. More particularly, with reference to Figures 1 and 2, the original search query

> "What is the liability of the United States under the Federal Tort Claims Act for injuries sustained by employees of an independent contractor working under contract with an agency of the United States government?"

is entered into the retrieval system as described in connection with Figure 4. The words of the query may be parsed, stopwords removed, and the remaining words stemmed, as indicated at steps 52, 54 and 56 in Figure 4, and the remaining stemmed words in the list of block 42 processed as query nodes q in Figure 2 to be compared to the texts of the definitions of the topical digests which act as document nodes d. Since the size of the definition portion of the topic and key database is relatively small (compared to the size of the document database), it is preferred that each word of the entire search query (including rooted words) perform the function of query nodes for comparison to the definition texts of the topics. In this preferred manner, the topical database is searched in parallel with the performance of steps 52-56 in Figure 4. In either case, a probability is determined for each topical definition that the definition matches the search query, and the key numbers associated with those definitions are identified.

At step 64, the key numbers are ranked in accordance with their probabilities, and the top n-ranked key numbers are selected for inclusion in the search query. More particularly, the probabilities are determined in accordance with the relationship expressed in equation 4, with $bel_{wtd\text{-}sum}(Q)$ being the probability that the key number matches the search query. In selecting one or more key numbers for inclusion in the search query, it is

preferred that n be no greater than 4. It may be preferable in some cases that the probability, $bel_{wtd-sum}(Q)$, for the selected key numbers be greater than some threshold value, such as 0.45, thereby assuring that all selected key numbers have a threshold relationship to the query. At step 66, the selected key numbers are added to the search query 44 (Figure 4) and serve as additional query nodes q.

In carrying out the invention, the example given above generates the following West Key Numbers (the titles to the topics associated with the Key Numbers being set forth for reference:

393K78(9)	United States. Property, Contracts and Liabilities. Torts. Personal Injuries in General.
170AK2515	Federal Civil Procedure. Tort Cases in General.
393K50	United States. Government and Officers. Liabilities of Officers or Agents for Negligence or Misconduct.
413K2085	Workers Compensation. Effect of Act on other Statutory or Common Law Rights of Action and Defense.

Citations

Case and statute citations are handled syntactically using word-level proximity. More particularly, citations in the original search query 40 (Figure 4) are identified and removed from the query and encoded into list 44 as single terms or query nodes comprising numeric tokens. For example, the citation 46 U.S.C. 688 may be encoded as 46 +2 688, (meaning 46 within two words of 688) and the citation 10 USPQ 2d 1985 may be encoded 10 +3 1985. To encompass most citations, particularly of State reporter systems, it is preferred to encode all citations as within five words. Hence, the above two citations will be encoded as 46 +5 688 and 10 +5 1985.

Hyphenations

Hyphenated terms in search queries are handled in much the same manner as citations. The hyphen is removed and the component words are searched using an adjacency operation which finds all adjacent occurrences of the component words.

Synonyms

Synonyms are created from a predefined database stored in ROM 24 (Figure 3). Examples of synonyms include 2d/2nd/second and habeas/habeus. Where a search query includes a word having a synonym, a new representation node r (Figure 2) is created for each synonym. However, the weight associated with the node is based on the frequency of the entire class of nodes comprising all synonyms, rather than any one term of the class.

Duplicate terms

Where a single word, term or phrase occurs more than once in a query, the word, term or phrase is evaluated only once. After the word, term or phrase has been processed for phrase identification as heretofore described, the duplicate word, term or phrase is simply

dropped from the search query, as was the case of the second occurrence of "United States" in the natural language query shown at reference numeral 60 in Figure 4. As will be explained hereinafter, the component probability score for each document containing a term duplicated in the query is multiplied by the query frequency, and the query normalization factor is increased by that frequency. Thus, the effect is that the duplicated term is evaluated multiple times as dictated by the query, but in a computationally simpler manner.

Thesauri

Thesauri are employed to identify words of similar or related meaning, as opposed to synonyms having identical meaning. In the present invention, the thesauri are used to suggest broader, narrower and related terms to the researcher for inclusion in the search query. The relationships are determined from the phrase database (such as ***Black's Law Dictionary***), from the topic and key database, and from an analysis of the statistical properties of the concept in texts (e.g., terms that frequently co-occur are related).

Document Retrieval

Phrases are not separately permanently identified in the document network. Instead, the representation nodes in the document network are created for the phrase by those concept nodes (Figure 1) which themselves are a function of the phrase in the query.

Figures 6A-6D illustrate different treatments of phrases in the document network of an inference network. Representation contents r_1 and r_2 shown in Figures 6A-6D correspond to two words in the text of document d_m. Representation content r_3 corresponds to the phrase in the text consisting of the two words. Q represents the query. For example, r_1 and r_2 may correspond to the

occurrence of the terms "independent" and "contractor", respectively, while r_3 corresponds to the occurrence of the phrase "independent contractor". In the model illustrated in Figure 6A, the phrase is treated as a separate representation content, independent of the contents corresponding to the component words. The belief in the phrase content can be estimated using evidence about component words and the relationship between them, including linguistic relationships. The presence of the query phrase concept in the document increases the probability that the document satisfies the query (or information need). The model of Figure 6B illustrates the case where the belief in the phrase concept depends on the belief in the concepts corresponding to the two component words. Figure 6C illustrates a term dependence model where the phrase is not represented as a separate concept, but as a dependence between the concepts corresponding to the component words. A document that contains both words will more likely satisfy the query associated with the phrase due to the increase belief coming from the component words themselves. However, experimentation has revealed that the model of Figure 6C is less appropriate for phrases and more appropriate for thesauri and synonyms. In Figure 6D belief in the phrase concept is established from evidence from the document text itself, whereas belief in the concepts representing component words are derived from belief in the phrase itself. The model of Figure 6D makes explicit the conditional dependence between the component concepts and addresses the practice of some authors that all component words of a phrase might not always be used in the text representation of a document. For the purposes of the present invention it is preferred that document network 10 employ the phrase model of Figure 6A so that the representation contents for the phrases are independent

of the corresponding words. Hence, a match between the concept node of a search query and the content node of a documentation representation is more likely to occur where the search query contains only the phrase, and not the component words. It is understood that the other models (Figures 6B-6D) could be employed with varying results.

Thus far, there has been described techniques for obtaining lists containing single words, phrases, proximity terms (hyphenations and citations) and key numbers. These elements represent the basic concept nodes contained in the query. The phrases, hyphenations and citations create representation nodes of the document network. Computer 20 (Figure 3) forwards the search query to computer 32, which determines the probability that a document containing some subset of these concepts matches the original query. For each single document, the individual concepts represented by each single word, phrase, proximity term, and key number of the query are treated as independent evidence of the probability that the document meets the information need, I. The probability for each concept is determined separately and combined with the other probabilities to form an overall probability estimate.

The probabilities for individual concepts are based on the frequency in which the concept occurs in the document (tf) and the frequency (f_i) with which the concept (i) occurs in the entire collection. The collection frequency may also be expressed as an inverse document frequency (idf_i). The inference network operates on two basic premises:

- A concept that occurs frequently in a document (a large tf) is more likely to be a good descriptor of that document's content, and
- A concept that occurs infrequently in the

collection (a large idf_i) is more likely to be a good discriminator than a concept that occurs in many documents.

It can be shown that the probability, $P(c_i|d_j)$ that concept c_i is a "correct" descriptor for document d_j may be represented as

$$P(c_i|d_j) = 0.4 + 0.6 \cdot idf_i \cdot tf_{ij}, \qquad \text{EQ 5}$$

where

$$tf_{ij} = 0.5 + 0.5 \cdot \frac{\log f_{ij}}{\log \max f_j} \qquad \text{EQ 6}$$

and

$$idf_i = \frac{\log \frac{d}{f_i}}{\log d} \qquad \text{EQ 7}$$

and where d is the number of documents in the collection, f_{ij} is the frequency of concept i in document j, f_i is the frequency of documents in the collection containing term i (i.e., the number of documents in which term i occurs), and max f_j is the maximum frequency for any term occurring in document j.

As shown by equation 4, the probability is computed for each concept/document pair, and the probabilities are summed. The result is normalized by the number of concepts in the query to determine the overall probability estimate that the document satisfies the information requirement set forth in the query.

Each document which does not contain the concept is assigned a default probability (0.4) which is essentially the probability that the concept accurately describes the content of the document even though its corresponding text does not occur in the document. For documents that do contain the concept, the application of Equations 5-7 is straightforward for single terms. For

documents that do contain the concept, the application of Equations 5-7 is straightforward for single terms. For proximity terms such as citations and hyphenations, the frequency of the satisfaction of the proximity constraints must be determined for the document and the collection as a whole with the new values are used f_{ij} and f_i. For example, in the case of a citation (such as 46 U.S.C. 688) the proximity constraint would be "46 +6 688" as heretofore explained. The frequencies f_{ij} and f_i become the frequencies that the proximity constraint is satisfied (that is, the number of times 46 occurs within six words of "688" for each document and for the number of documents in the collection as a whole).

Phrases are treated in a manner similar to proximity terms, except that a document which does not contain the full phrase receives a partial score for a partial phrase. For example, if a query contains the phrase "FEDERAL TORT CLAIMS ACT" and a document contains the phrase "tort claims" but not "Federal Tort Claims Act", the document will receive a score based on the frequency distribution associated with "TORT CLAIMS". Figure 7 is a flow diagram illustrating the process of handling partial matches. As shown at step 68, the full phrase is evaluated against the collection as heretofore described. The inverse document frequency (idf_i) is determined for the full phrase (step 70), and if idf_i is greater than a predetermined threshold (e.g., 0.3) the maximum belief achieved for any single term is selected as the belief for the partial phrase (step 72). If idf_i is smaller or equal to the threshold value (0.3), the preselected default belief (0.4) is assigned to the documents containing the partial phrase (step 74).

Since the frequency of "TORT CLAIMS" must equal or exceed that of the longer phrase, the probability estimate for the partial phrase would generally be lower

than that assigned to documents containing the complete phrase. For phrases which occur extremely often (for example, where idf_i is less than 0.3) it is preferred to dispense with the partial matching strategy, and treat the phrase as a pure proximity term by assigning the default belief (0.4) to all documents containing the partial phrase but not the full phrase (step 74). For phrases which appear less often (where idf_i is greater than 0.3), the maximum belief achieved by any single word of the partial phrase is assigned to the belief for the partial phrase.

As previously explained, duplicate terms are purged from the search query. However, where duplicate terms appear in the search query, the component probability score for each document containing the term is multiplied by the query frequency. For example, if a document contains a term which appears twice in a natural language query receives a component probability of 0.425, the probability score is multiplied by 2 (to 0.850) for that term. When the probabilities are summed and normalized as described above, the normalization factor is increased to reflect the frequency of the duplicated term (increased by 1 in this example). Thus, the duplicated term is treated as if it had been evaluated multiple times as dictated by the query, but in a computationally simpler manner.

As described above, the probability estimates for each document/concept pair are summed and the result is normalized by the number of concepts in the query. For the example given in Figure 4 the search query shown in block 44 employs eleven concepts, so the total probability for each document will be divided by 11 to determine the overall probability that the given document meets the overall query. For example, assume for a given document that the eleven probabilities are:

0.400	0.430	0.466
0.543	0.436	0.433
0.512	0.400	0.481
0.460	0.472	

The overall probability is the sum of the individual probabilities (5.033) divided by the number of concepts (11) for a total probability of 0.458. This indicates a probability of 0.458 that the document meets the full query shown in block 40 in Figure 4. The probability estimate is determined for each document represented in the database, whereupon they are ranked in accordance with the value of the probability estimate to identify the top n documents. The ranking or identification is provided by computer 32 (Figure 3) to computer 20 for display and/or printout at output terminal 22. Additionally, the document texts may be downloaded from computer 32 to computer 20 for display and/or printout at output terminal 22.

Figures 8 - 10 are detailed flowcharts of the inference network, Figure 8 being a detailed flowchart of the query network 10, Figure 9 being a detailed flowchart of the topic and key subroutine, and Figure 10 being a detailed flowchart of the document network 12. As heretofore described, an input query written in natural language is loaded into the computer, such as into a register therein, and is parsed (step 80) compared to the stopwords in database 82 (step 84) and stemmed at step 86. At step 88, all duplicate terms are located, mapped, counted and removed, with a count x representing the number of duplicate terms removed. The result is the list 42 illustrated in figure 4. Using synonym database 90, the list is compared at step 92 to the synonym database and synonyms are added to the list. As will be explained hereinafter, the handling of synonyms may actually occur after handling of the phrases. Citations

are located at step 94 and are related by word-level proximity numbers as heretofore described. More particularly, a proximity relationship is established showing the page number within five words of the volume number, without regard to the reporter system employed. The handling of citations, like the handling of synonyms, may be accomplished after phrase resolution, if desired.

Employing phrase database 96, a decision is made step 98 as to whether or not phrases are present in the query. If phrases are present, a comparison is made as step 100 to identify phrases. At step 102 a determination is made as to whether successive phrases share any common term(s) (an overlap condition). More particularly, and as heretofore described, terms which are apparently shared between successive phrases are detected at step 102. At step 104 a determination is made as to which phrase is the longer of the two phrases, and the shared term is included in the longer phrase and excluded from the shorter phrase. As a result of deleting the shared term from the shorter phrase, the resulting shorter phrase may not be a phrase at all, in which case the remaining term(s) are simply handled as stemmed words. On the other hand, if the two phrases are of equal length, then the shared term is accorded to the first phrase and denied to the second phrase.

After overlap conflict is resolved at step 104, the resulting phrase substitution occurs at step 106. The process loops back to step 98 to determine if phrases are still present, and if they are the process repeats until no further phrases are present, and the search query illustrated at block 44 in figure 4 is developed.

As heretofore described, the handling of synonyms and citations may occur after resolution of the phrases, rather than before.

Topic and key subroutine 108 receives input from

the parsing step 80 and returns key numbers for inclusion in the list 44 illustrated in figure 4. The key numbers may be inserted into the search query before or after the handling of phrases, as desired. Topic and key subroutine 108 is illustrated in greater detail in Figure 9 and includes an input from the parsing step 80 to count the number of terms in the input query and set the number to the number z, at step 110. Step 112, i is set to 0, and at step 114 1 is added to i. Using topic and key database 116 as previously described, the terms of the input query are compared to the terms of the topic and key database to determine the frequency that each term in the input query appears in the database 116 (idf_i) and to determine the frequency that the term appears in the respective text of the respective topic (tf_{ij}). Thereupon the probability is determined that the topic text meets the individual term of the information need of the input query by determining $0.4 + 0.6\ idf_i \cdot tf_{ij}$. The results for all terms are accumulated at step 122, and at step 124 a determination is made as to whether all of the terms of the input query have been processed. More particularly, a determination is made as to whether i equals z, and if not, the process loops back to step 114 to add 1 to i and continue the process using the next term. When the process has looped through each of the terms of the input query so that i equals z, the amount accumulated at step 120 through the several loops is normalized by dividing by z at step 126 and storing the result at step 128. As a result of the steps through step 126 is to determine the probability that the input query is satisfied by the topical text. The entire process is repeated for the other terms of the input query (step 130), and the topical texts are ranked to determine the top n texts (step 132), it being preferred that n is no greater than 4. If desired, those texts

having a probability less than a predetermined threshold may be eliminated at step 134. The result is fed back to ROM 24 containing database 116 to download the key numbers associated with the selected topical texts to list 44 being developed in Figure 8. It may be possible to eliminate topical texts from the compare process after comparing less than all of the terms in the text to terms in the input query. More particularly, if a text scores so low a probability after comparing a few terms that it becomes evident it cannot score in the top four topics, the text can be discarded from further consideration. Further although the process has been described in connection with a text-based scan system, a concept-based scan system may be employed instead. Thus, the text-based scan and concept-based scan are similar to the document-based scan and concept-based scan described above in connection with the inference network.

As illustrated in Figure 10, the resulting search query is provided to the document network where, at step 140 the number of terms z is counted, at step 142 i is set to 0 and at step 144 1 is added to i. Using document database 146 which also contains the text of the documents, the frequency that each term appears in database 146 (idf_i) is determined and the frequency that the term appears in the respective text (tf_{ij}) are determined at step 148. The component probability is determined at step 150 as heretofore described and is accumulated with other component probabilities at step 152. At step 154 a determination is made as to whether or not i equals z (where z is the number of terms in the search query). If all of the terms have not been compared to the database, the process is looped, adding 1 to i and repeated for each term until i equals z at step 154. As heretofore described, when terms having duplicates deleted from the input query are processed at

step 148, the probability for such terms is multiplied by the number of duplicates deleted, thereby weighing the probability in accordance with the frequency of the term in the original input query. Consequently, at step 156, it is necessary to divide the accumulated component probability for the document by x + z (where x is the number of duplicate terms deleted from the input query) to thereby normalize the probability. The probability for each document is stored at step 158 and the process repeated at step 160 for the other documents. At step 162 the documents are ranked in accordance with the determined probabilities, and the top ranked documents are printed out or displayed at step 164.

As previously described, the scan technique may be a concept-based scan, rather than the document-based scan described. Further, as previously described, the scan may be aborted after less than complete scan of any given document if the probabilities result in a determination that the document will not reach the cutoff for the n top-ranked documents to be displayed or printed.

While the present invention has been described in connection with a time-shared computer system shown in Figure 3 wherein search queries are generated by PC computers or dumb terminals for transmission to and time-shared processing by a central computer containing the document network, it may be desirable in some cases to provide the document network (with or without the document text database) to the user for direct use at the PC. In such a case, the document database would be supplied on the same ROM 24 as the databases used with the search query, or on a separately supplied ROM for use with computer 20. For example, in the case of a legal database, updated ROMs containing the document database could be supplied periodically on a subscription basis to

the user. In any case, the stopwords, phrases and key numbers would not be changed often, so it would not be necessary to change the ROM containing the databases of stopwords, phrases and key numbers.

Although the present invention has been described with reference to preferred embodiments, workers skilled in the art will recognize that changes may be made in form and detail without departing from the spirit and scope of the invention.

WHAT IS CLAIMED IS:

1. A computer-implemented process for forming a search query for use in a computer-implemented process to search and identify documents likely to match an input query defining the composition of the text of documents sought to be identified, the input query comprising a plurality of terms composed in natural language, the process comprising:

a) providing a database containing a plurality of stopwords and a plurality of phrases, each of the phrases consisting of a plurality of stemmed terms;

b) inputting the input query to a computer;

c) parsing the input query into separate terms;

d) comparing each term of the input query to the stopwords in the database and deleting each term from the input query matching a stopword;

e) stemming the terms of the input query to form stemmed terms;

f) comparing sequences of stemmed terms to each phrase in the database to identify those sequences of stemmed terms corresponding to phrases in the database; and

g) substituting identified sequence of stemmed terms with the corresponding phrase from the database to form the search query.

2. A computer-implemented process for forming a search query according to claim 1 further including, for each sequence of terms forming a phrase, identifying those terms which are shared by successive phrases, and assigning the shared term to the phrase containing the most terms or to the first phrase of the sequence.

3. A computer-implemented process for forming a

search query according to claim 1 further including providing a second database containing a plurality of topics each having a descriptive text and an associated unique numerical key, the text being composed of a plurality of terms, comparing the terms of the input query or the search query to the terms of the texts of the topics in the second database, assigning a statistical weight to each topical text reflecting the probability that the text matches the query, ranking the topical texts based on the statistical weight, and inserting into the search query the numerical keys associated with up to n highest ranked topical texts, where n is a predetermined integer.

4. A computer-implemented process for forming a search query according to claim 3 wherein the statistical weight of the topical texts associated with numerical keys inserted into the search query exceed a predetermined threshold.

5. A computer-implemented process for forming a search query according to claim 3 wherein the statistical weight for each topical text is determined by comparing each term of the query to the each term of the text, determining the probability that the query term is a correct descriptor of the text in accordance with the relationship

$$P(c_i|d_j) = 0.4 + 0.6 \cdot idf_i \cdot tf_{ij},$$

where idf_i is the frequency that the term appears in the second database and tf_{ij} is the frequency that the term appears in the respective text, for each text adding tne probabilities for all terms of the query and normalizing the result by the number of terms in the query.

6. A computer-implemented process for forming a search query according to claim 1 wherein the input query may include one or more sequences of terms forming citations having numerical portions, the process further

includes:

h) identifying each sequence of terms in the input query comprising a citation, and

i) substituting each identified sequence of terms with a citation term comprising the numerical portions of the citation linked with a predetermined word-level proximity number.

7. A computer-implemented process according to claim 1 further including identifying documents likely to match the input query by:

h) providing a second database containing representations of the contents of the texts of a plurality of documents to be searched the texts of each document containing a plurality of terms;

i) after step (g), comparing the search query to the representations for each document in the second database to identify the frequency of occurrences of the stemmed query terms and identified phrases in the representations for each document;

j) assigning a statistical weight to the document representing the probability that the document matches the search query based on the number of occurrences of the stemmed query terms and identified phrases in the representations for each document; and

k) ranking the documents based on the statistical weight found in step (j).

8. A computer-implemented process according to claim 7 wherein the input query may include one or more sequences of terms forming citations having numerical portions, the process further includes:

l) identifying each sequence of terms in the input query comprising a citation, and

m) substituting each identified sequence of terms with a citation term comprising the numerical portions of the citation linked with a predetermined word-level proximity number,

step (i) includes comparing the identified citations to terms and sequences of terms in the representations for each document, and step (j) includes assigning a statistical weight to each document concerning the probability that the document matches the search query on the frequency of occurrences of the identified citations in the representations for each document.

9. A computer-implemented process according to claim 7 further including displaying the texts of selected ones of the documents.

10. A computer-implemented process according to claim 1 employing an inference network for identifying documents likely to match the input query, the input query comprising a plurality of terms composed in natural language defining the composition of the text of documents sought to be identified, the inference network being implemented in computer means having a query network and a document network, the document network providing a second database representing a plurality of documents to be searched, each document containing a plurality of terms each represented by a node, the computer means comparing each term, i, of a search query, c, to each of the nodes of each document, j, to determine the probability that the individual term of the search query, c_i, is a correct descriptor of the document in accordance with the relationship

$$P(c_i|d_j) = 0.4 + 0.6 \cdot idf_i \cdot tf_{ij},$$

where idf_i is the frequency that term i appears in the entire collection of documents in the second database, and tf_{ij} is the frequency that the term, i, appears in the respective document, j, the computer means adding,

for each document in the second database, the probabilities for each term of the search query and normalizing the result by the number of terms in the search query, the computer means ranking the documents in accordance with the results of adding the probabilities for each document.

11. A computer-implemented process according to claim 10 wherein the input query may include one or more sequences of terms forming citations having numerical portions, and the establishing a query network further includes:

identifying each sequence of terms in the input query comprising a citation, and

substituting each identified sequence of terms with a citation term comprising the numerical portions of the citation linked with a predetermined word-level proximity number so that each citation term becomes a term, i, and

step (d) includes comparing the identified citation terms to the nodes of each document and determining the probability that the term is the correct descriptor of the document.

12. A computer-implemented process according to claim 11 further including displaying the texts of selected ones of the documents.

13. A computer system for forming a search query to be used in a computer-implemented process to search and identify documents likely to match an input query defining the composition of the text of documents sought to be identified, the input query comprising a plurality of terms composed in natural language, said system comprising:

a read only memory containing a database consisting of a plurality of stopwords and a plurality of phrases, each of the phrases each

consisting of a plurality of stemmed terms;

register means for storing an input query;

parsing means responsive to the register means for parsing the input query;

first comparing means for comparing each term in the register means to the stopwords in the database;

first deleting means responsive to the first comparing means for deleting each term from the register means matching a stopword;

first processing means for stemming each term in the register means to a stemmed term;

second comparing means for comparing sequences of stemmed terms in the register means to each phrase in the database to identify those sequences of stemmed terms which correspond to phrases in the database; and

second processing means for substituting identified sequence of stemmed terms in the register means with the corresponding phrase from the database.

14. A computer system for forming a search query according to claim 13 further including third processing means for identifying those terms which are shared by successive phrases, and fourth processing means for assigning the shared term to the phrase containing the most terms or to the first phrase of the sequence.

15. A computer system for forming a search query according to claim 13 wherein the read only memory further contains a second database consisting of a plurality of topics each having a descriptive text and an associated unique numerical key, the text being composed of a plurality of terms, third comparing means for comparing the terms of the input query or the search query to the terms of the texts of the topics in the

second database, fifth processing means for assigning a statistical weight to each topical text reflecting the probability that the text matches the query, ranking means for ranking the topical texts based on the statistical weight, the register means being responsive to the ranking means to store the numerical keys associated with up to n highest ranked topical texts, where n is a predetermined integer.

16. A computer system for forming a search query according to claim 15 wherein the statistical weight of the topical texts associated with numerical keys stored into the register means exceed a predetermined threshold.

17. A computer system for forming a search query according to claim 15 further including fourth comparing means for comparing each term of the query to each term of the text, sixth processing means for determining the probability that the query term is a correct descriptor of the text in accordance with the relationship

$$P(c_i|d_j) = 0.4 + 0.6 \cdot idf_i \cdot tf_{ij},$$

where idf_i is the frequency that the term appears in the second database and tf_{ij} is the frequency that the term appears in the respective text, adding means for adding for each text the probabilities for all terms of the query, and normalizing means responsive to the adding means for normalizing the result by the number of terms in the query.

18. A computer system for forming a search query according to claim 13 wherein the input query may include one or more sequences of terms forming citations having numerical portions, said system further including:

seventh processing means for identifying each sequence of terms in the input query comprising a citation, and

eighth processing means for substituting each identified sequence of terms with a

citation term comprising the numerical portions of the citation linked with a predetermined word-level proximity number.

19. A computer system according to claim 13 wherein the system is arranged to identify documents likely to match the input query, the system further comprising:

a second memory containing a second database containing representations of the contents of the texts of a plurality of documents to be searched, each document text containing a plurality of terms;

third comparing means for comparing the search query in the register means to the representations for the terms of each document in the second memory to identify the frequency of occurrences of the stemmed terms and identified phrases in the representations for each document;

third processing means responsive to the third comparing means for assigning a statistical weight to each document represented in the second database concerning the probability that the document matches the search query; and

fourth processing means responsive to the third processing means for ranking the documents according to statistical weight.

20. A computer system according to claim 19 wherein the input query may include one or more sequences of terms forming citations having numerical portions, said system further including fifth processing means for identifying each sequence of terms in the input query comprising a citation, sixth processing means for substituting each identified sequence of terms in the register with a citation term comprising the numerical

portions of the citation linked with a predetermined word-level proximity number, fourth comparing means for comparing the citations in the register means to the representations of the terms for each document in the second memory to identify the frequency of occurrences of the citations in the representations for each document; the third processing means is further responsive to the fourth comparing means for assigning a statistical weight to each document concerning the probability that the document matches the search query.

21. A computer system according to claim 19 further including display means for displaying the texts of selected ones of the documents.

22. The computer system according to claim 13 employing an inference network for identifying documents likely to match the input query, the input query comprising a plurality of terms composed in natural language defining the composition of the text of documents sought to be identified, the inference network including computer means and a read only memory arranged in a query network and a document network, the document network comprising a second database containing a plurality of nodes representing the text of each of a plurality of documents, the computer means having third compare means for comparing each term, i, of the search query, c, to each of the nodes of each document, j, in the second database, third processing means for determining the probability that the individual term of the search query, c_i, is a correct descriptor of the document, j, in accordance with the following relationship

$$P(c_i|d_j) = 0.4 + 0.6 \cdot idf_i \cdot tf_{ij},$$

where idf_i is the frequency that documents containing term i appear in the entire collection of documents in the second database, and tf_{ij} is the frequency that the

term, i, appears in the respective document, j, adding means for adding the probabilities determined by the processing means for each term of the search query for each document in the second database, normalizing means responsive to the adding means for normalizing the sums of probabilities by the number of terms in the search query, and ranking means responsive to the normalizing means for ranking the documents in the second database in accordance with the values of the normalized sums of probabilities.

23. A computer system according to claim 22 wherein the input query may include one or more sequences of terms forming citations having numerical portions, and the computer means of the query network further includes:

fourth processing means for identifying each sequence of terms in the input query comprising a citation, and

substitution means for substituting each identified sequence of terms with a citation term comprising the numerical portions of the citation linked with a predetermined word-level proximity number so that each citation term becomes a term, i, of search query, c, and

the third compare means being further responsive to the substitution means for comparing each term, i, of the search query, c, to each of the nodes, j, in the second database.

24. A computer system according to claim 22 further including display means for displaying the texts of selected ones of the documents.

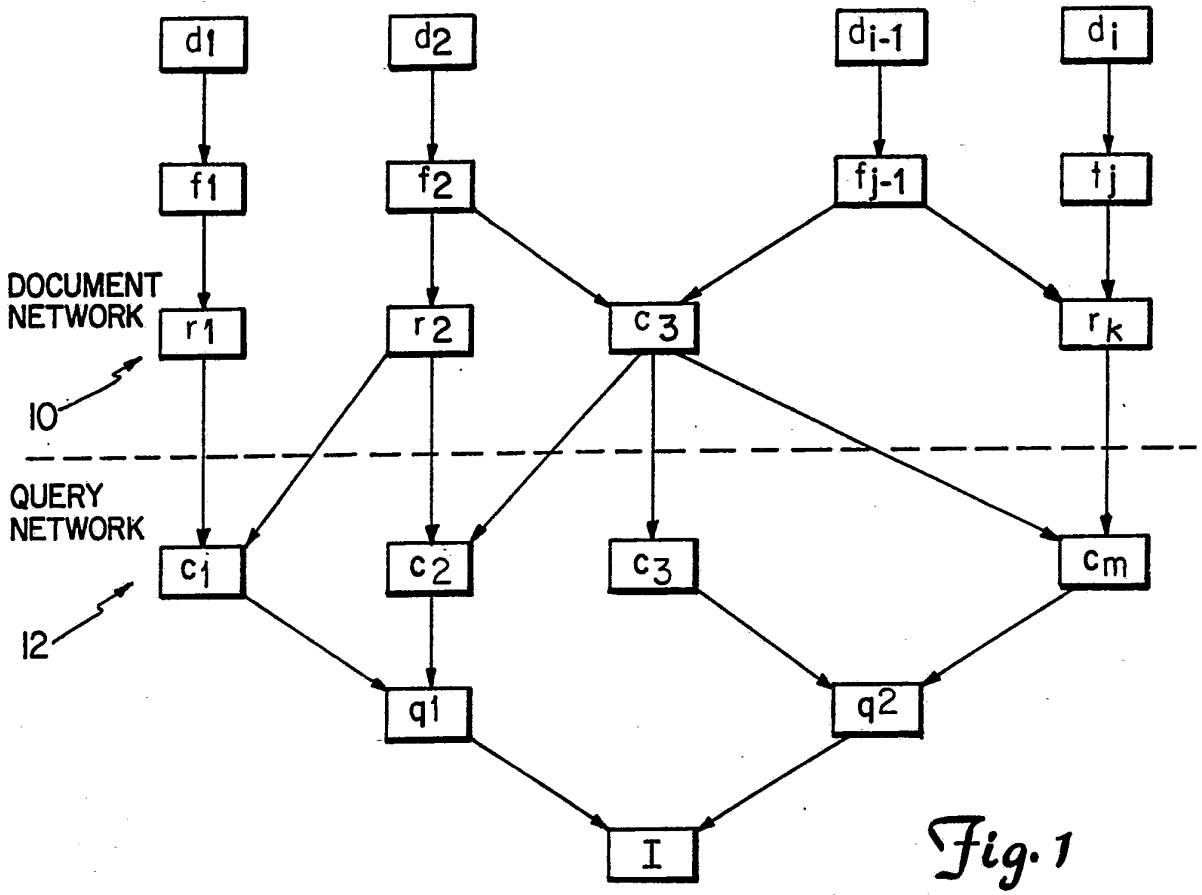

d1
d2
di-1
di
f1
f2
fj-1
tj
DOCUMENT NETWORK
r1
r2
c3
rk
10
QUERY NETWORK
ci
c2
c3
cm
12
q1
q2
I

Fig. 1

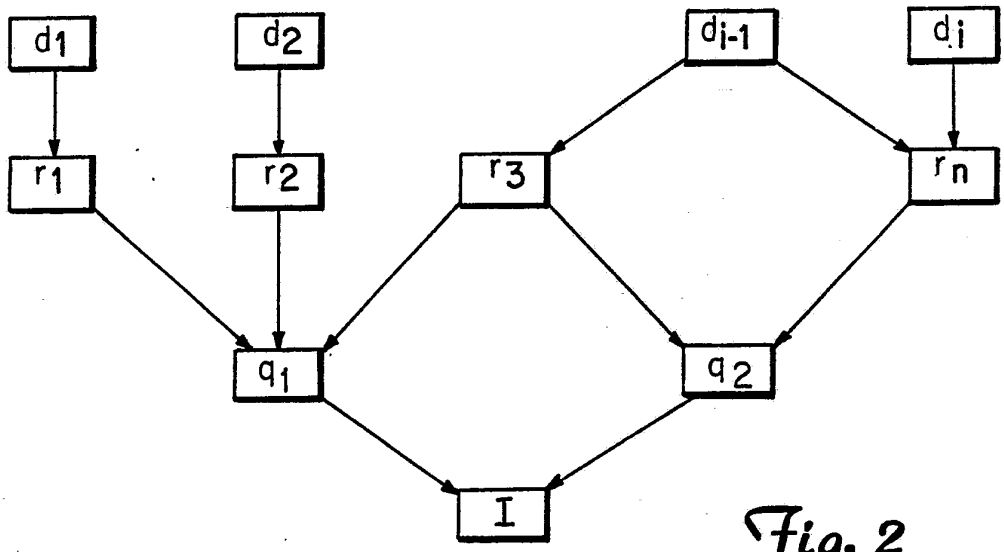

d1
d2
di-1
di
r1
r2
r3
rn
q1
q2
I

Fig. 2

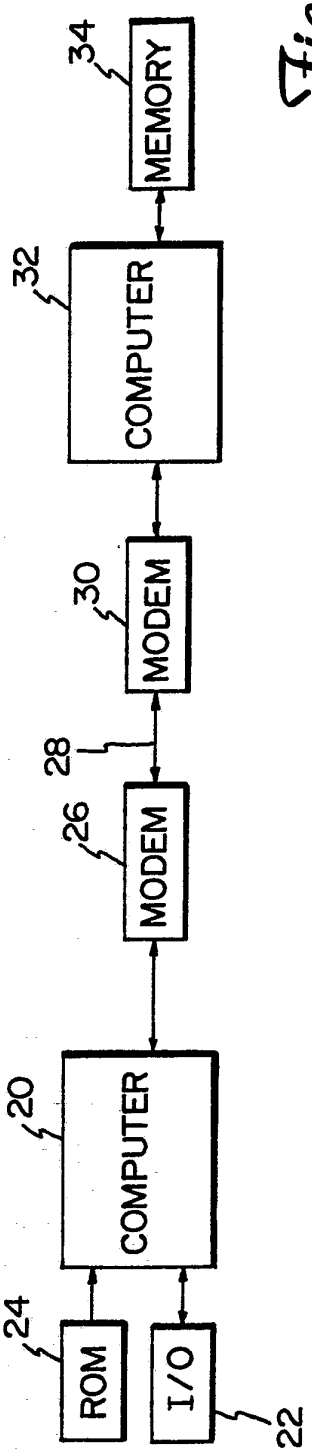

ROM
I/O
COMPUTER
MODEM
MODEM
COMPUTER
MEMORY
24
22
20
26
28
30
32
34

Fig. 3

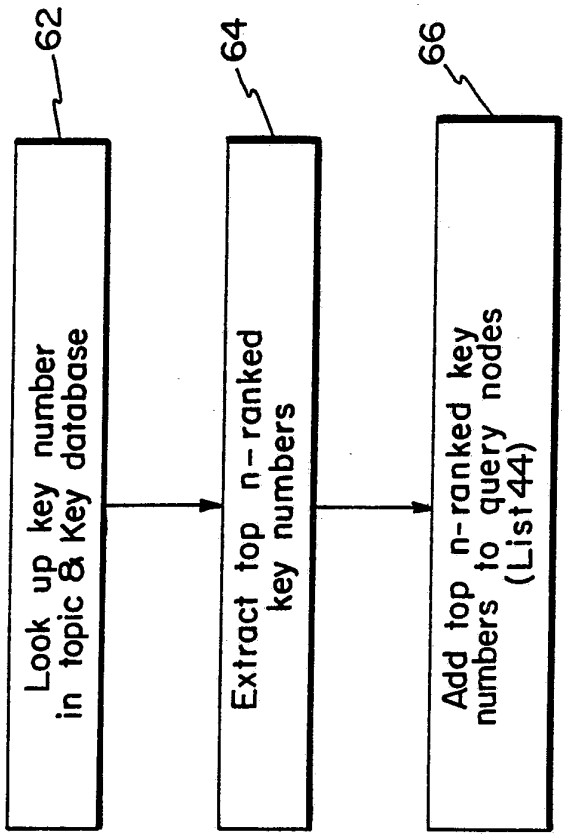

Look up key number in topic & Key database
Extract top n-ranked key numbers
Add top n-ranked key numbers to query nodes (List 44)
62
64
66

Fig. 5

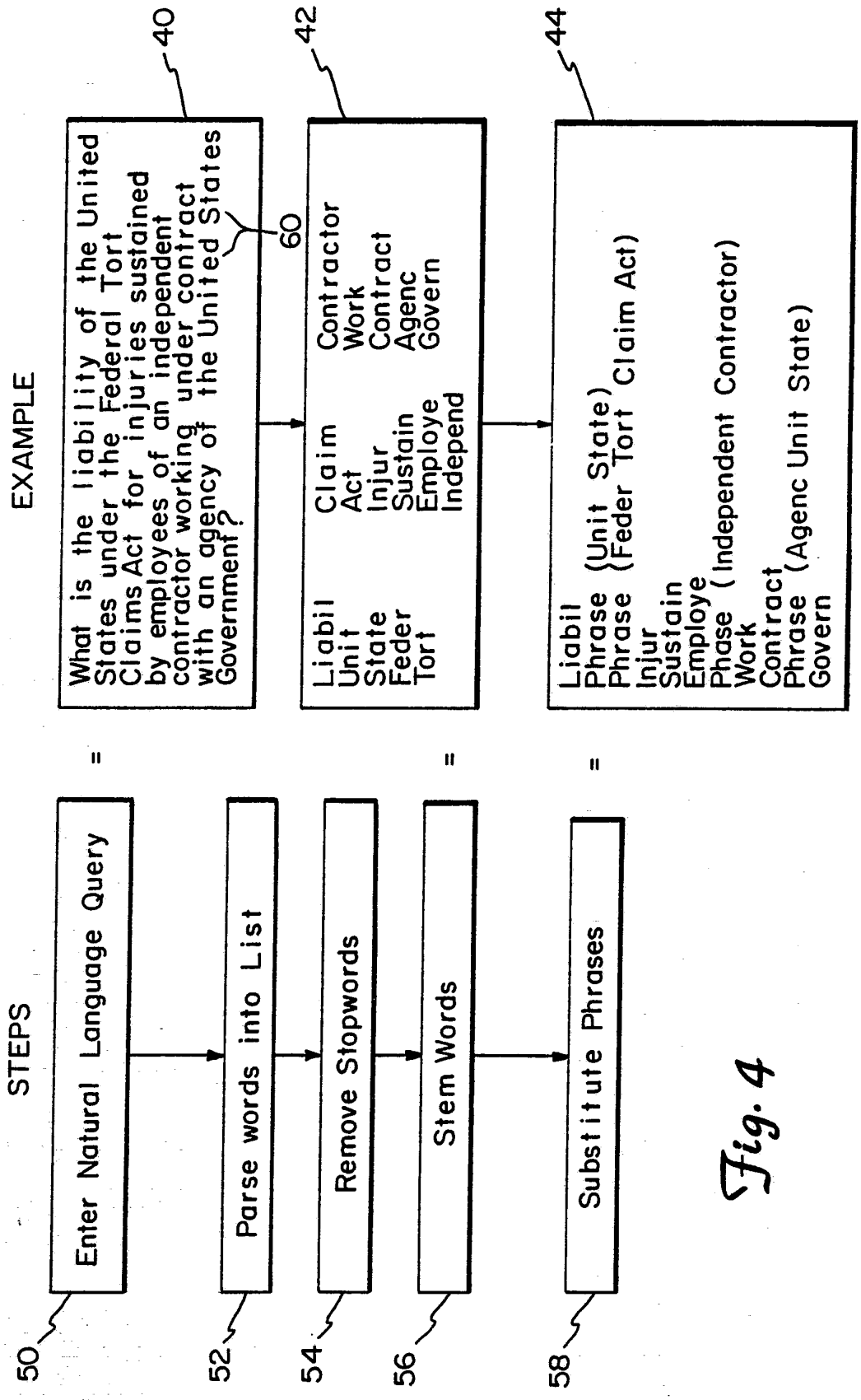

STEPS
EXAMPLE
Enter Natural Language Query
Parse words into List
Remove Stopwords
Stem Words
Substitute Phrases
50
52
54
56
58
=
=
=
40
42
44
60
What is the liability of the United States under the Federal Tort Claims Act for injuries sustained by employees of an independent contractor working under contract with an agency of the United States Government?
Liabil Unit State Feder Tort
Claim Act Injur Sustain Employe Independ
Contractor Work Contract Agenc Govern
Liabil
Phrase (Unit State)
Phrase (Feder Tort Claim Act)
Injur
Sustain
Employe
Phase (Independent Contractor)
Work
Contract
Phrase (Agenc Unit State)
Govern

Fig. 4

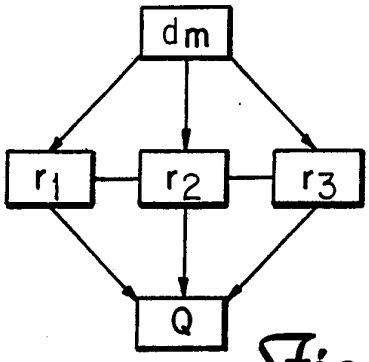


Fig 6A

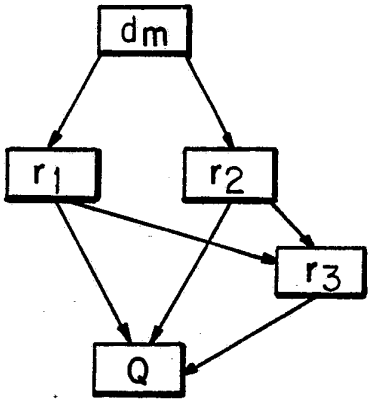


Fig. 6B

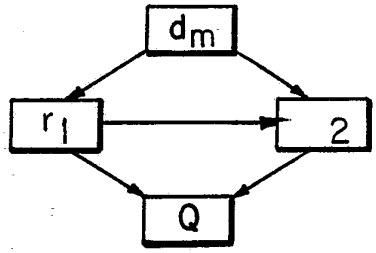


Fig. 6C

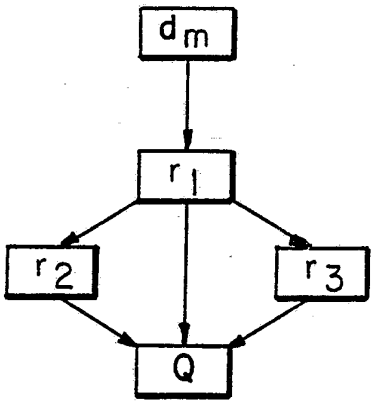
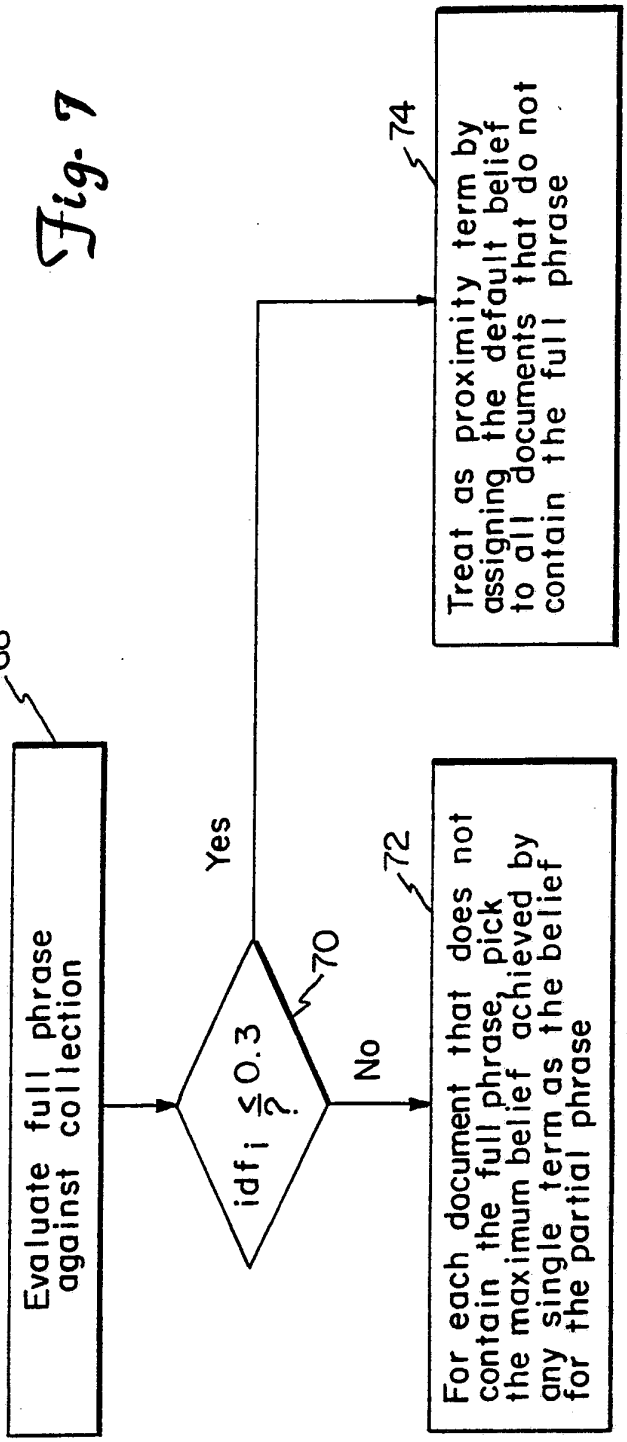


Fig. 6D

Fig. 7


68
Evaluate full phrase against collection
idf i ≤ 0.3 ?
70
Yes
No
72
For each document that does not contain the full phrase, pick the maximum belief achieved by any single term as the belief for the partial phrase
74
Treat as proximity term by assigning the default belief to all documents that do not contain the full phrase

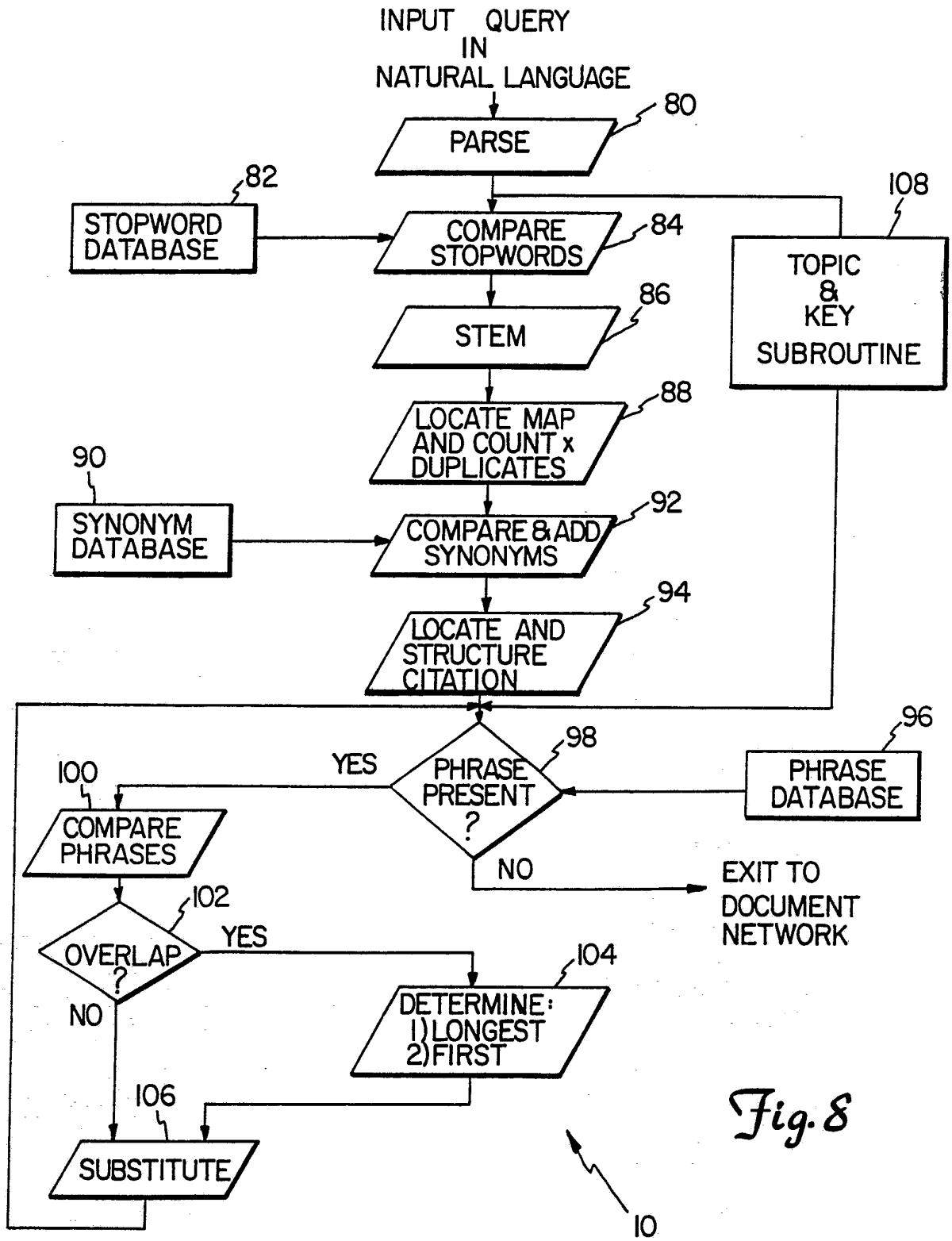

INPUT QUERY IN NATURAL LANGUAGE
PARSE
80
STOPWORD DATABASE
82
COMPARE STOPWORDS
84
TOPIC & KEY SUBROUTINE
108
STEM
86
LOCATE MAP AND COUNT x DUPLICATES
88
SYNONYM DATABASE
90
COMPARE & ADD SYNONYMS
92
LOCATE AND STRUCTURE CITATION
94
PHRASE PRESENT ?
98
YES
PHRASE DATABASE
96
COMPARE PHRASES
100
NO
EXIT TO DOCUMENT NETWORK
OVERLAP ?
102
YES
NO
DETERMINE: 1) LONGEST 2) FIRST
104
SUBSTITUTE
106
10

Fig. 8

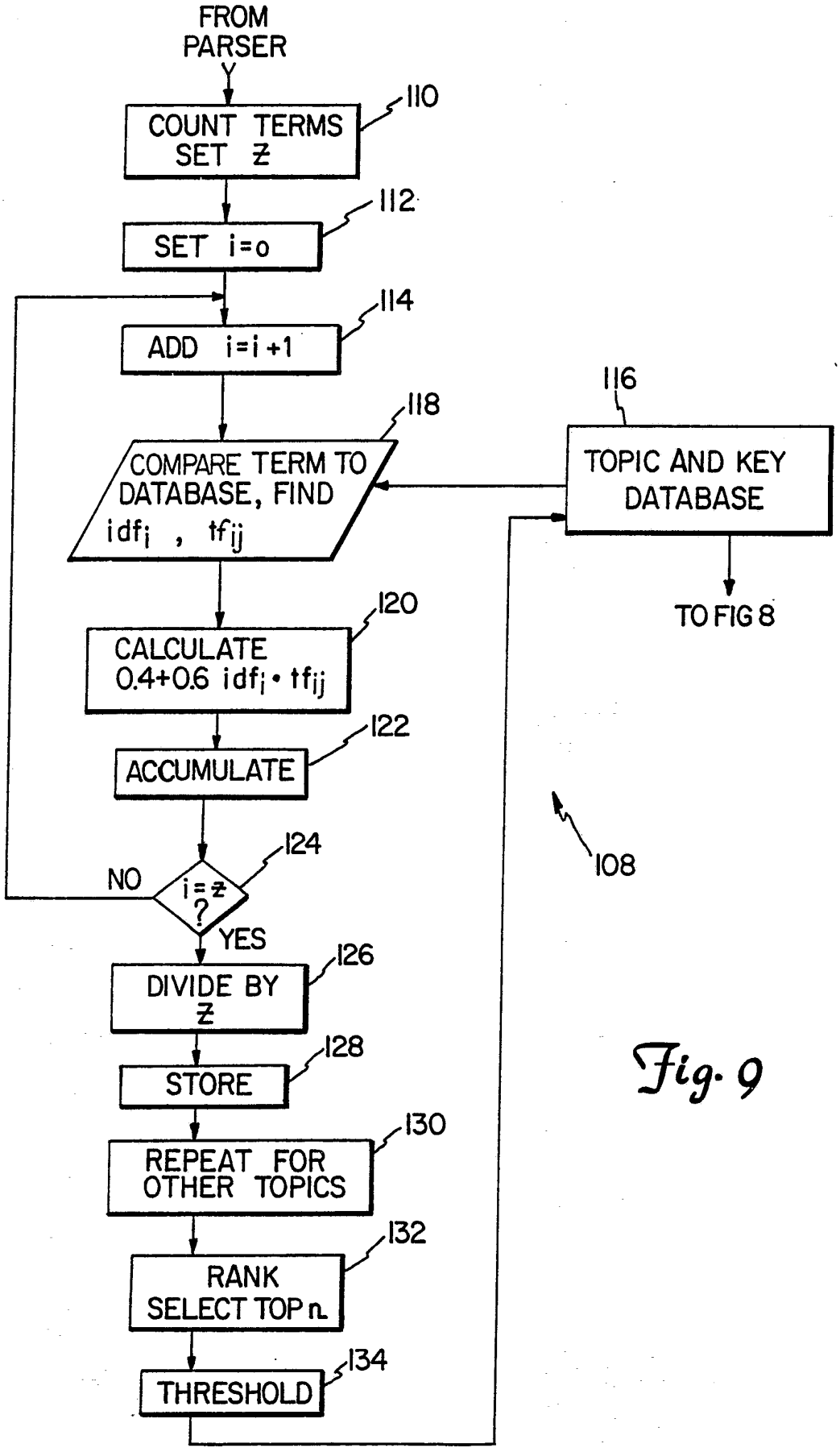

FROM PARSER
COUNT TERMS SET z
110
SET i=o
112
ADD i=i+1
114
COMPARE TERM TO DATABASE, FIND idf_i , tf_ij
118
116
TOPIC AND KEY DATABASE
TO FIG 8
CALCULATE 0.4+0.6 idf_i • tf_ij
120
ACCUMULATE
122
i=z ?
124
NO
YES
DIVIDE BY z
126
STORE
128
REPEAT FOR OTHER TOPICS
130
RANK SELECT TOP n
132
THRESHOLD
134
108

Fig. 9

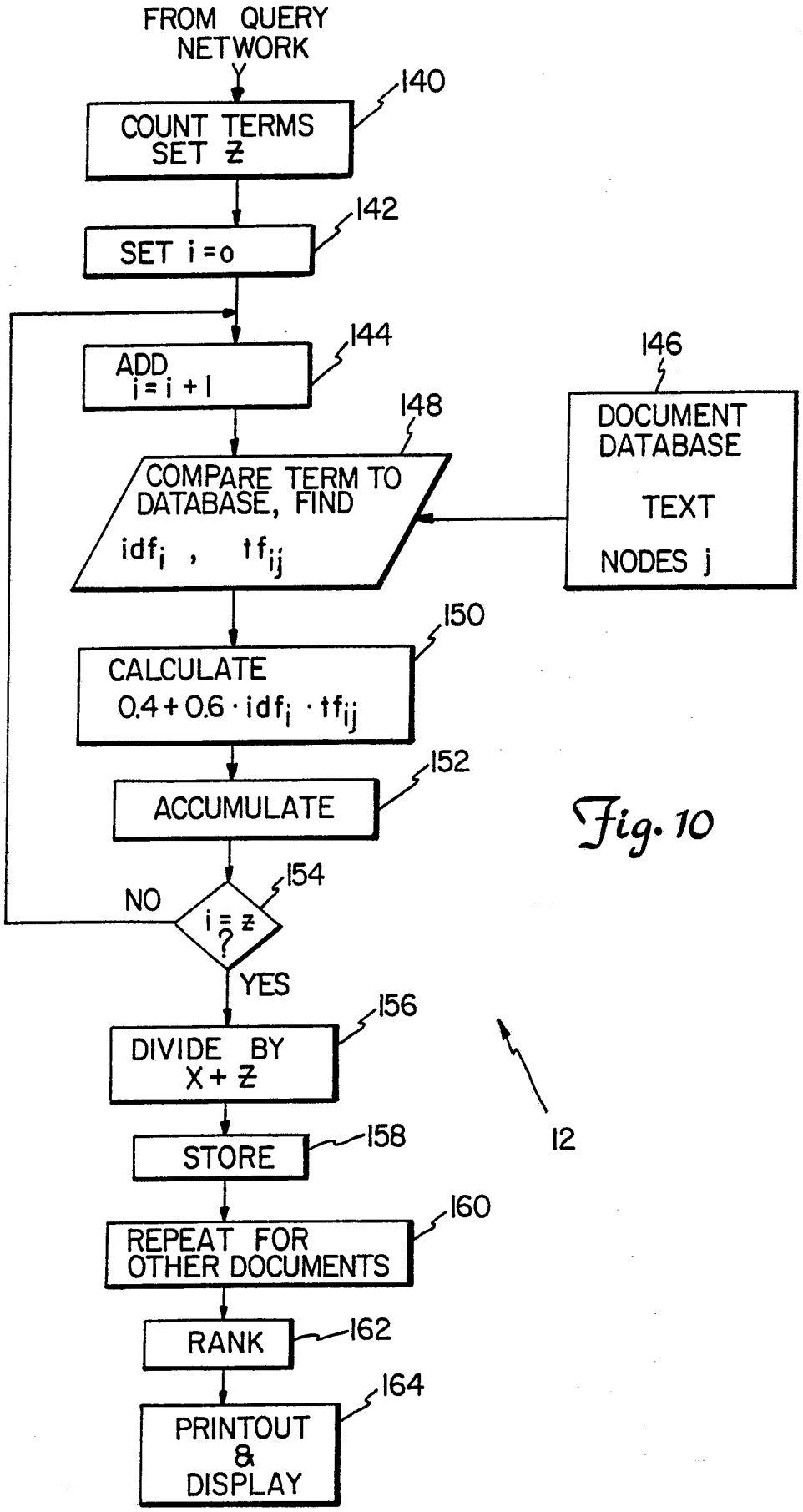

FROM QUERY NETWORK
Y
140
COUNT TERMS SET Z
142
SET i=o
144
ADD i=i+1
146
DOCUMENT DATABASE
TEXT
NODES j
148
COMPARE TERM TO DATABASE, FIND
idf_i , tf_ij
150
CALCULATE
0.4+0.6·idf_i·tf_ij
152
ACCUMULATE
154
NO
i=z ?
YES
156
DIVIDE BY X+Z
158
STORE
160
REPEAT FOR OTHER DOCUMENTS
162
RANK
164
PRINTOUT & DISPLAY
12

Fig. 10

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US92/08383

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 15/40
US CL :395/600; 364/419; 364/513
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/200; 364/300

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

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

APS: PARS ### Query Inference Stem ### (w) word

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US,P, 4,832,308 (BARBIC ET AL) 18 APRIL 1989 Abstract; column 2 line 63 et seq.	1-2 3-24
Y	US,P, 4,839,853 (DEERWESTER ET AI) 13 JUNE 1989 Abstract; column 9 & 10	3-24
Y	SIGIR 90, SEPTEMBER 1990, (TURTLE ET AL) Inference Networks for document Retrieval fig 1	10-24

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

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

Date of the actual completion of the international search 23 NOVEMBER 1992	Date of mailing of the international search report 15 JAN 1993
Name and mailing address of the ISA/ Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE	Authorized officer WAYNE AMSBURY Telephone No. (703) 308-4560

Form PCT/ISA/210 (second sheet)(July 1992)★