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

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2457693 A1 2003/02/27

(21) 2 457 693

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2002/08/14
(87) Date publication PCT/PCT Publication Date: 2003/02/27
(85) Entrée phase nationale/National Entry: 2004/02/13
(86) N° demande PCT/PCT Application No.: US 2002/025756
(87) N° publication PCT/PCT Publication No.: 2003/017143
(30) Priorités/Priorities: 2001/08/14 (60/312,385) US;
2001/11/08 (10/007,299) US

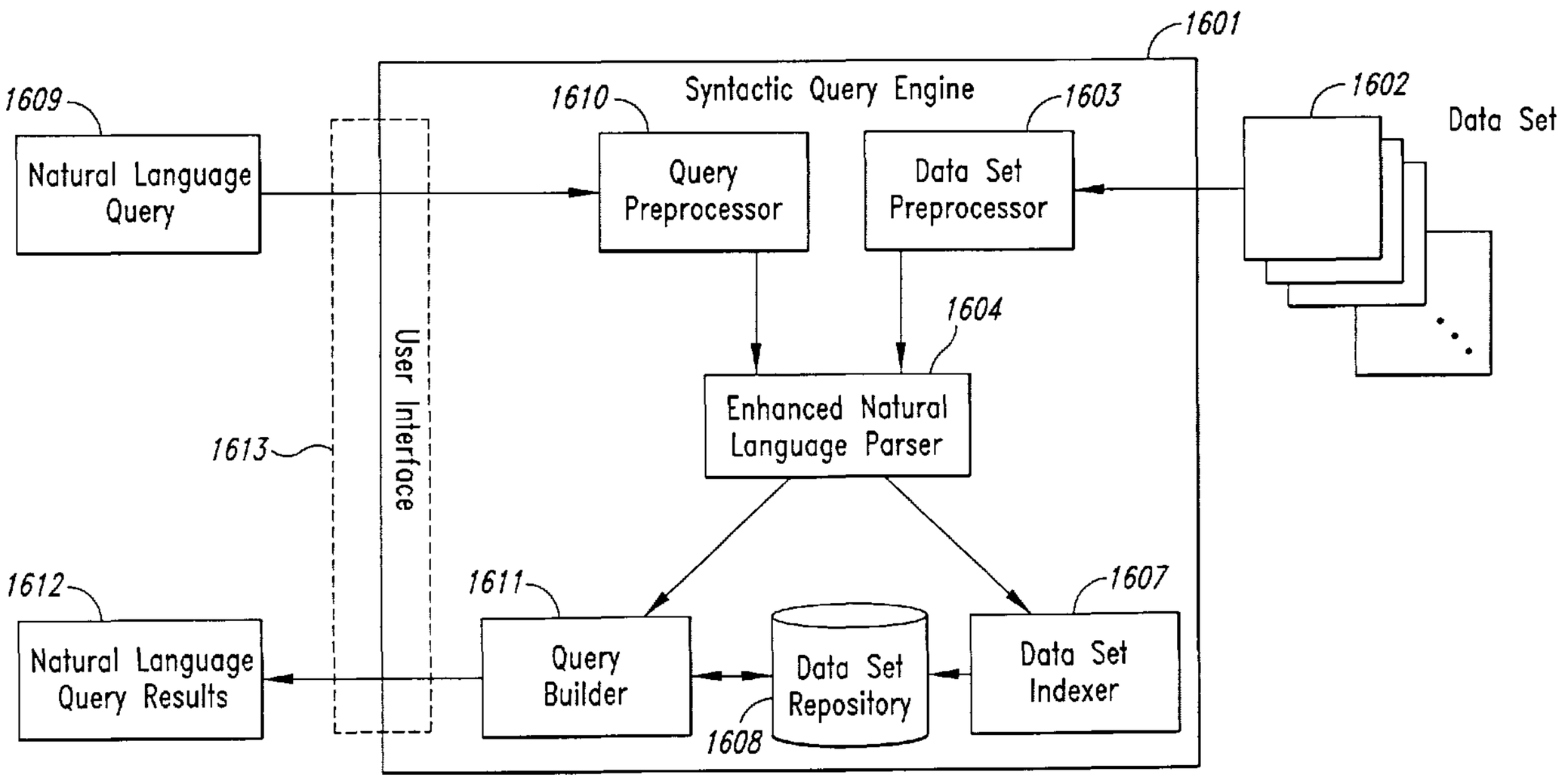
(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/30, G06F 17/27

(71) Demandeur/Applicant:
INSIGHTFUL CORPORATION, US

(72) Inventeurs/Inventors:
MARCHISIO, GIOVANNI B., US;
KOPERSKI, KRZYSZTOF, US;
LIANG, JISHENG, US;
MURUA, ALEJANDRO, US;
NGUYEN, THIEN, US

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : PROCEDE ET SYSTEME PERMETTANT D'EFFECTUER UNE RECHERCHE AMELIOREE DES DONNEES
(54) Title: METHOD AND SYSTEM FOR ENHANCED DATA SEARCHING



(57) Abrégé/Abstract:
Methods and systems for syntactically indexing and searching data sets to achieve more accurate search results are provided. Example embodiments provide a Syntactic Query Engine ("SQE") that parses, indexes, and stores a data set, as well as processes natural language queries subsequently submitted against the data set. The SQE comprises a Query Preprocessor, a Data Set Preprocessor, a Query Builder, a Data Set Indexer, an Enhanced Natural Language Parser ("ENLP"), a data set repository, and, in some embodiments, a user interface. After preprocessing the data set, the SQE parses the data set and determines the syntactic and grammatical roles of each term to generate enhanced data representations for each object in the data set. The SQE indexes and stores these enhanced data representations in the data set repository. Upon subsequently receiving a query, the SQE parses the query similarly and searches the indexed stored data set to locate data that contains similar terms used in similar grammatical roles. In this manner, the SQE is able to achieve more contextually accurate search results more frequently than using traditional search engines.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 February 2003 (27.02.2003)

PCT

(10) International Publication Number
WO 03/017143 A3

(51) International Patent Classification[7]: G06F 17/30, 17/27

(21) International Application Number: PCT/US02/25756

(22) International Filing Date: 14 August 2002 (14.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/312,385	14 August 2001 (14.08.2001)	US
10/007,299	8 November 2001 (08.11.2001)	US

(71) Applicant *(for all designated States except US)*: **INSIGHTFUL CORPORATION** [US/US]; Suite 500, 1700 Westlake Avenue North, Seattle, WA 98109-3044 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MARCHISIO, Giovanni, B.** [US/US]; Unit 303, 9815 NE 130th Place, Kirkland, WA 98034 (US). **KOPERSKI, Krzysztof** [CA/US]; Apt. D, 2311 Yale Avenue East, Seattle, WA 98102 (US). **LIANG, Jisheng** [CN/US]; 6343 114th Avenue Southeast, Bellevue, WA 98006 (US). **MURUA, Alejandro** [CL/US]; Apt. 302, 1310 East Thomas Street, Seattle, WA 98102 (US). **NGUYEN, Thien** [US/US]; 22220 98th Avenue West, Edmonds, WA 98020 (US).

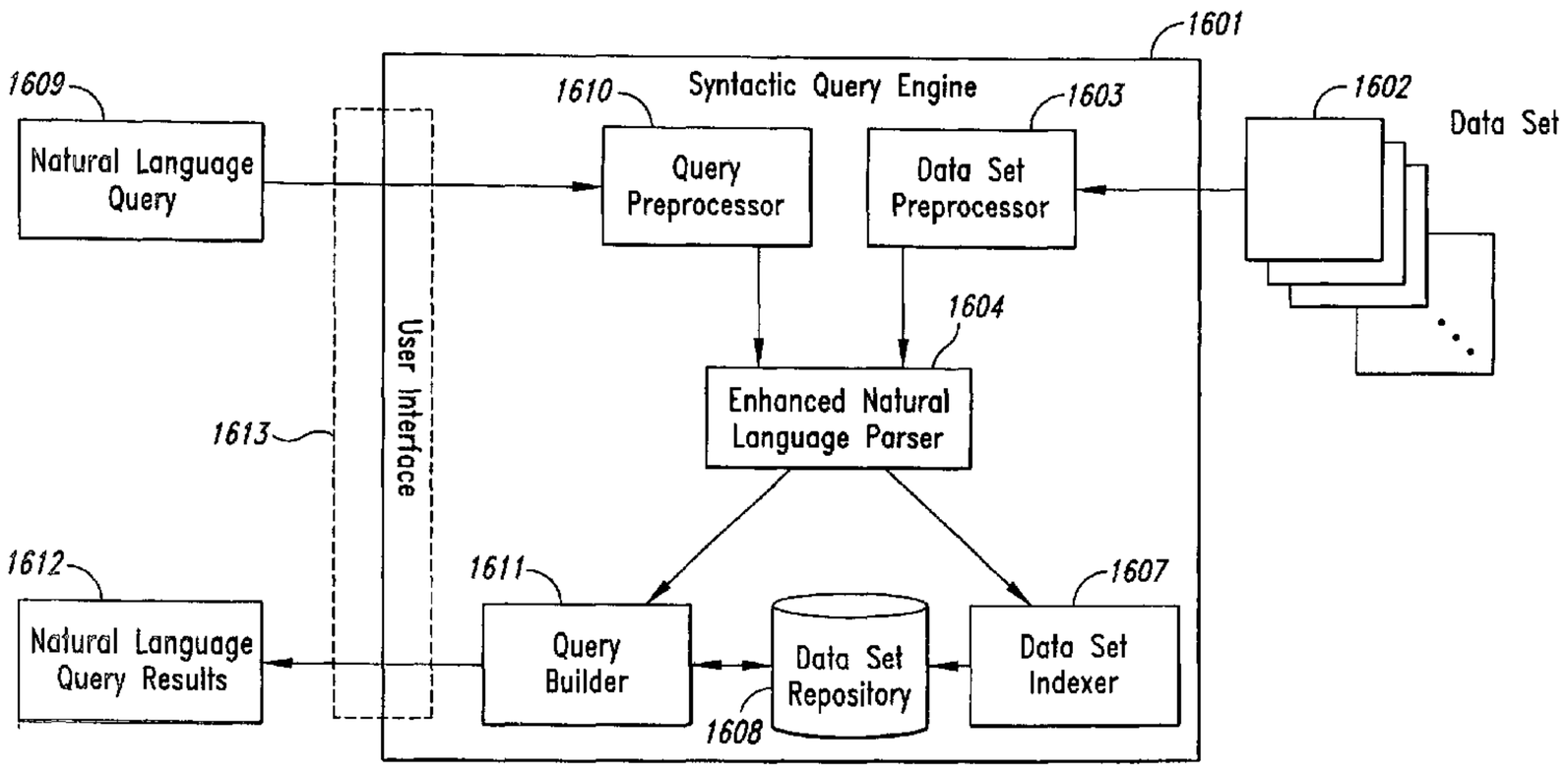
(74) Agents: **BIERMAN, Ellen, M.** et al.; Seed Intellectual Property Law Group PLLC, Suite 6300, 701 Fifth Avenue, Seattle, WA 98104-7092 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, 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, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

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

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR ENHANCED DATA SEARCHING



(57) Abstract: Methods and systems for syntactically indexing and searching data sets to achieve more accurate search results are provided. Example embodiments provide a Syntactic Query Engine ("SQE") that parses, indexes, and stores a data set, as well as processes natural language queries subsequently submitted against the data set. The SQE comprises a Query Preprocessor, a Data Set Preprocessor, a Query Builder, a Data Set Indexer, an Enhanced Natural Language Parser ("ENLP"), a data set repository, and, in some embodiments, a user interface. After preprocessing the data set, the SQE parses the data set and determines the syntactic and grammatical roles of each term to generate enhanced data representations for each object in the data set. The SQE indexes and stores these enhanced data representations in the data set repository. Upon subsequently receiving a query, the SQE parses the query similarly and searches the indexed stored data set to locate data that contains similar terms used in similar grammatical roles. In this manner, the SQE is able to achieve more contextually accurate search results more frequently than using traditional search engines.

WO 03/017143 A3

WO 03/017143 A3

Published:
— *with international search report*

(88) Date of publication of the international search report:
30 October 2003

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.

METHOD AND SYSTEM FOR ENHANCED DATA SEARCHING

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a method and system for searching for information in a data set, and, in particular, to methods and systems for syntactically indexing and searching data sets to achieve greater search result accuracy.

Description of the Related Art

Often times it is desirable to search large sets of data, such as collections of millions of documents, only some of which may pertain to the information being sought. In such instances it is difficult to either identify a subset of data to search or to search all data yet return only meaningful results. Several search techniques have been used to support searching large sets of data, none of which have been able to attain a high degree of accuracy of search results due to their inherent limitations.

One common technique is that implemented by traditional keyword search engines. Data is searched and results are generated based on matching one or more words or terms designated as a query. The results are returned because they contain a word or term that matches all or a portion of one or more keywords that were submitted to the search engine as the query. Some keyword search engines additionally support the use of modifiers, operators, or a control language that specifies how the keywords should be combined in a search. For example, a query might specify a date filter to be used to filter the returned results. In many traditional keyword search engines, the results are returned ordered, based on the number of matches found within the data. For example, a keyword search against Internet websites typically returns a list of sites that contain one or more of the submitted keywords, with the sites with the most matches appearing at the top of the list. Accuracy of search results in these systems is thus presumed to be associated with frequency of occurrence.

One drawback to traditional search engines is that they don't return data that doesn't match the submitted keywords, even though it may be relevant. For example, if a user is searching for information on what products a

particular country imports, data that refers to the country as a "customer" instead of using the term "import" would be missed if the submitted query specifies "import" as one of the keywords, but doesn't specify the term "customer." (*E.g.*, The sentence "Argentina is a customer of the Acme Company" would be missed.) Ideally, a user would be able to submit a query in the form of a question and receive back a set of results that were accurate based on the meaning of the query – not just on the specific terms used to phrase the question.

Natural language parsing provides technology that attempts to understand and identify the syntactical structure of a language. Natural language parsers have been used to identify the parts of speech of each term in a submitted sentence to support the use of sentences as natural language queries. They have been used also to identify text sentences in a document that follow a particular part of speech pattern; however, these techniques fall short of being able to produce meaningful results when the documents do not follow such patterns. The probability of a sentence falling into a class of predefined sentence templates or the probability of a phrase occurring literally is too low to provide meaningful results. Failure to account for semantic and syntactic variations across a data set, especially heterogeneous data sets, has led to disappointing results.

BRIEF SUMMARY OF THE INVENTION

Embodiments of the present invention provide methods and systems for syntactically indexing and searching data sets to achieve more accurate search results. Example embodiments provide a Syntactic Query Engine ("SQE") that parses, indexes, and stores a data set, as well as processes queries subsequently submitted against the data set. The SQE parses each object in the data set and transforms it into a canonical form that can be searched efficiently using techniques of the present invention. To perform this transformation, the SQE determines the syntactic structure of the data by parsing (or decomposing) each data object into syntactic units, determines the grammatical roles and relationships of the syntactic units, and represents these relationships in a normalized data structure. A set of heuristics is used to determine which relationships among the syntactic units are important for yielding greater accuracy in the results subsequently returned in response to queries. The normalized data structures are then stored and

indexed. The SQE processes queries in a similar fashion by parsing them and transforming them into the same canonical form, which is then used to generate and execute queries against the data set.

In one embodiment, the parsing of each data object into syntactic units is performed by a natural language parser, which generates a hierarchical data structure (*e.g.*, a tree) of syntactic units. In other embodiments, the parser is a module that generates a syntactic structure (or lexical structure) that relates specifically to the objects of the data set. In yet another embodiment, the parser is an existing, off-the-shelf parser, that is modified to perform the SQE transformations of the data set or of queries.

In some embodiments, the canonical form is an enhanced data representation, such as an enhanced sentence representation. In one embodiment, the canonical form comprises a set of tables, which represent grammatical roles of and / or relationships between various syntactic units. In some embodiments, tables are created for the subject, object, preposition, subject/object, noun/noun modifier roles and / or relationships of the syntactic units.

In one embodiment, use of the normalized data structure allows data that appears in multiple and different languages and in different forms to be processed in a similar manner and at the same time. For example, a single query can be submitted against a corpus of documents written in different languages without first translating all of the documents to one language. In another embodiment, the data set may include parts that themselves contain multiple language elements. For example, a single document, like a tutorial on a foreign language, may be written in several languages. In another embodiment, the data set may include objects containing computer language. In yet another embodiment, the data set may include graphical images, with or without surrounding text, bitmaps, film, or other visual data. In yet another embodiment, the data set may include audio data such as music. In summary, the data set may include any data that can be represented in syntactical units and follows a grammar that associates roles to the syntactical units when they appear in a specified manner, even if the data may not traditionally be thought of in that fashion.

In one embodiment, the processed queries are natural language queries. In other embodiments, the queries are specific representations with form and / or meaning that is specifically related to the objects of the data set.

In one embodiment, the SQE comprises a Query Preprocessor, a Data Set Preprocessor, a Query Builder, a Data Set Indexer, an Enhanced Natural Language Parser ("ENLP"), a data set repository, and, in some embodiments, a user interface. After preprocessing the data set, the SQE parses the data set and determines the syntactic and grammatical roles of each term to generate enhanced data representations for each object (*e.g.*, sentence) in the data set. The SQE indexes and stores these enhanced data representations in the data set repository. Upon subsequently receiving a query, such as a natural language query, the SQE parses the query similarly and searches the stored indexed data set to locate data that contains similar terms used in similar grammatical roles.

In some embodiments, the SQE provides search operators based upon the grammatical roles and relationships of syntactic units of objects of data. For example, some embodiments provide a search that allows designation of the grammatical role of a unit of the query. For example, a term may be designated as a subject or an object of a sentence before it is used to search the data set for similar terms used in similar grammatical roles. In one embodiment, the SQE returns a list of related units (terms) based upon a grammatical role. For example, in response to a query that designates a term as a "subject" of a textual phrase, the SQE returns a list of verbs that appear in phrases that contain the term used as the subject of those phrases. Other embodiments return different parts of speech or terms that appear in particular grammatical roles.

In yet other embodiments, the SQE provides an ability to search for similar sentences in a data set of documents or similar objects, where similar means that the matching sentence contains similar words used in similar grammatical roles or syntactic relationships. In some embodiments, this ability is invoked by selection of a sentence in data that is returned as a result of another query. In yet other embodiments, the SQE provides an ability to search for similar paragraphs and similar documents. In other embodiments, the SQE provides an ability to search for similar objects.

In some embodiments, the SQE returns results to a query in an order that indicates responsiveness to the query. In some embodiments, the order is based upon the polysemy of terms in the query. In other embodiments, the order is based upon the inverse document frequency of terms in the query. In yet other embodiments, the ordering of results is based upon weightings of

the matched results from the data set. In some of these embodiments, the weightings are based upon the degree of matching of a particular part of speech. For example, the weighting may be based upon whether matching sentences contain identical verbs, entailed verbs, or verbs that are defined as close by some other metric, such as frequency or distribution in the data set.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Figure 1 shows a natural language query and the results returned by an example embodiment of a Syntactic Query Engine.

Figure 2 is an example block diagram of a Syntactic Query Engine.

Figure 3 is an example flow diagram of the steps performed by a Syntactic Query Engine to process data sets and natural language queries.

Figure 4 is an example screen display illustrating general search functionality of an example Syntactic Query Engine user interface.

Figure 5A is an example screen display illustrating a portion of a data set from which a natural language query result was extracted.

Figure 5B is an example screen display illustrating a search similar sentence operation.

Figure 5C is an example screen display illustrating results that correspond to the search similar sentence operation initiated in Figure 5B.

Figure 6 is an example screen display illustrating Syntactic Query Engine results from a natural language query that requests a map.

Figure 7 is an example screen display illustrating a map that corresponds to the query result selected in Figure 6.

Figure 8 is an example screen display illustrating Syntactic Query Engine results from a natural language query that requests a chart.

Figure 9 is an example screen display illustrating a chart that corresponds to the query result selected in Figure 8.

Figure 10 is an example screen display of an advanced search using a natural language query that contains a subject.

Figure 11 is an example screen display illustrating advanced search results from a query that contains a subject.

Figure 12 is an example screen display illustrating a portion of resulting sentences returned by a Syntactic Query Engine when a particular verb is selected from the verb list.

Figure 13 is an example screen display of advanced search functionality using a natural language query that contains a subject and an object.

Figure 14 is an example screen display illustrating advanced search results from a query that contains a subject and an object.

Figure 15 is an example screen display illustrating the designation of programmable attributes in a Syntactic Query Engine.

Figure 16 is a block diagram of the components of an example embodiment of a Syntactic Query Engine.

Figure 17 is a block diagram of the components of an Enhanced Natural Language Parser of an example embodiment of a Syntactic Query Engine.

Figure 18 is a block diagram of the processing performed by an example Enhanced Natural Language Parser.

Figure 19 is a block diagram illustrating a graphical representation of an example syntactic structure generated by the natural language parser component of an Enhanced Natural Language Parser.

Figure 20 is a table illustrating an example enhanced data representation generated by the postprocessor component of an Enhanced Natural Language Parser.

Figure 21 is an example block diagram of data set processing performed by a Syntactic Query Engine.

Figure 22 is an example block diagram of natural language query processing performed by a Syntactic Query Engine.

Figure 23 is an example block diagram of a general purpose computer system for practicing embodiments of a Syntactic Query Engine.

Figure 24 is an example flow diagram of the steps performed by a build_file routine within the Data Set Preprocessor component of a Syntactic Query Engine.

Figure 25 illustrates an example format of a tagged file built by the build_file routine of the Data Set Preprocessor component of a Syntactic Query Engine.

Figure 26 is an example flow diagram of the steps performed by the dissect_file routine of the Data Set Preprocessor component of a Syntactic Query Engine.

Figure 27 is an example flow diagram of the steps performed by a Syntactic Query Engine to process a natural language query.

Figure 28 is an example flow diagram of the steps performed by a preprocess_natural_language_query routine of the Query Preprocessor component of a Syntactic Query Engine.

Figure 29 is an example block diagram showing the structure of an example Data Set Repository of a Syntactic Query Engine.

Figure 30 is an example flow diagram of the steps performed by a parse_sentence routine of the Enhanced Natural Language Parser component of a Syntactic Query Engine.

Figure 31 is an example flow diagram of the steps performed by a determine_grammatical_roles subroutine within the parse_sentence routine of the Enhanced Natural Language Parser.

Figure 32 is an example flow diagram of the steps performed by a generate_subject_structure subroutine of the determine_grammatical_roles routine.

Figure 33 is an example flow diagram of the steps performed by a generate_object_structure subroutine of the determine_grammatical_roles routine.

Figure 34 is an example flow diagram of the steps performed by a generate_subject_modifier subroutine of the determine_grammatical_roles routine.

Figure 35 is an example flow diagram of the steps performed by a generate_generalized_subject_object subroutine of the determine_grammatical_roles routine.

Figure 36A is a graphical representation of an example parse tree generated by a natural language parser component of an Enhanced Natural Language Parser.

Figure 36B is an illustration of an enhanced data representation of an example natural language query generated by an Enhanced Natural Language Parser.

Figure 37 is an example flow diagram of the steps performed by a construct_output_string routine of the Enhanced Natural Language Parser.

Figure 38 is an example flow diagram of the steps performed by an index_data routine of the Data Indexer component of a Syntactic Query Engine.

Figures 39A and 39B are example flow diagrams of the steps performed by a build_query routine within the Query Builder component of a Syntactic Query Engine.

DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the present invention provide methods and systems for syntactically indexing and searching data sets to achieve more accurate search results. Example embodiments provide a Syntactic Query Engine ("SQE") that parses, indexes, and stores a data set, as well as processes queries subsequently submitted against the data set. The SQE parses each object in the data set and transforms it into a canonical form that can be searched efficiently using techniques of the present invention. To perform this transformation, the SQE determines the syntactic structure of the data by parsing (or decomposing) each data object into syntactic units, determines the grammatical roles and relationships of the syntactic units, and represents these relationships in a normalized data structure. A set of heuristics is used to determine which relationships among the syntactic units are important for yielding greater accuracy in the results subsequently returned in response to queries. The normalized data structures are then stored and indexed. The SQE processes queries in a similar fashion by parsing them and transforming them into the same canonical form, which is then used to generate and execute queries against the data set..

In one embodiment, the SQE includes, among other components, a data set repository and an Enhanced Natural Language Parser ("ENLP"). The ENLP parses the initial data set and the natural language queries and determines the syntactic and grammatical roles of terms in the data set / query. Then, instead of matching terms found in the query with identical terms found in the data set (as done in typical keyword search engines), the SQE locates terms in the data set that have similar grammatical roles to the grammatical roles of similar terms in the original query. In this manner, the SQE is able to achieve more contextually accurate search results more frequently than using traditional search engines.

One skilled in the art will recognize that, although the techniques are described primarily with reference to text-based languages and collections of documents, the same techniques may be applied to any collection of terms, phrases, units, images, or other objects that can be represented in syntactical

units and that follow a grammar that defines and assigns roles to the syntactical units, even if the data object may not traditionally be thought of in that fashion. Examples include written or spoken languages, for example, English or French, computer programming languages, graphical images, bitmaps, music, video data, and audio data. Sentences that comprise multiple words are only one example of a phrase or collection of terms that can be analyzed, indexed, and searched using the techniques described herein.

In addition, the term "natural language query" is used to differentiate the initial query from subsequent data queries created by an SQE in the process of transforming the initial query into a set of data-set-specific queries (*e.g.*, database queries) that are executed against the indexed data set. One skilled in the art will recognize, however, that the form and content of the initial query will relate to and depend upon the form of objects in the data set—*i.e.*, the language of the data set.

The Syntactic Query Engine is useful in a multitude of scenarios that require indexing, storage, and/or searching of, especially large, data sets, because it yields results to data set queries that are more contextually accurate than other search engines. In a text-based, document environment, the SQE identifies the syntax of a submitted natural language query or sentence within a data set (*e.g.*, which terms are nouns, adjectives, verbs, and other parts of speech, and the relationships between the terms), determines which terms are less likely to produce meaningful results (*e.g.*, words like “the”, “a”, and “who”), and ignores these terms. For example, given the natural language query,

> What types of scientific research does the Department of Defense fund?

the SQE identifies “scientific” as an adjective, “research” as a noun, “Department of Defense” as a noun phrase, and “fund” as a verb. The other terms in the sentence are ignored as being less meaningful. Based on the identified syntax, the SQE determines the grammatical role of each meaningful term (*e.g.*, whether the term is a subject, object, or governing verb). Based upon a set of heuristics, the SQE uses the determined roles to generate one or more data queries from the natural language query to execute against the data set, which has been indexed and stored previously using a similar set of heuristics. For example, in the query above, “Department of Defense” is determined to be a subject of the sentence, “fund” is determined to be the governing verb of the sentence, and “scientific” and “research” are both

determined to be objects of the sentence. Based on the determined grammatical roles, the SQE is able to generate an enhanced data representation and, hence, data queries that tend to return more contextually accurate results because the data set has been transformed to a similar enhanced data representation. Specifically, rather than identifying data in the data set that matches terms or parts of terms in the query (like traditional keyword search engines), the SQE executes the data queries to return data that contains similar terms used in similar grammatical roles to those terms and grammatical roles in the submitted natural language query. In summary, the SQE uses its determination of the grammatical roles of terms to transform a syntactic representation of data in the data set and in queries submitted against the data to a canonical representation that can be efficiently compared.

Figure 1 shows a natural language query and the results returned by an example embodiment of a Syntactic Query Engine. The example natural language query,

Does Argentina import or export gas?

shown in query box 102, when executed against a previously indexed data set, returns results in results area 103 that relate to Argentina's importation and exportation of gas, even though the terms "import" and "export" do not appear in all of the records of the results. Thus, the SQE is able to make what one would consider greater contextual associations between the designated natural language query and the data set than a traditional keyword search would provide. This capability is due to the SQE's ability to index data sets and perform syntactical searches based on determined grammatical roles of and relationships between terms in the query and terms within sentences in the data set, as opposed to keyword searches, yielding a more effective search tool. The SQE is thus able to "recognize" that certain terms in a data set may be relevant in relation to a submitted query simply because of their grammatical roles in the sentence. For example, the first sentence returned 104 refers to Argentina as a "customer" as opposed to an "importer." This sentence may not have been returned as a result of the shown natural language query had a traditional keyword search been performed instead, because "customer" is not identical or a part of the term "importer."

Figure 2 is an example block diagram of a Syntactic Query Engine. A document administrator 202 adds and removes data sets (for example, sets of documents), which are indexed and stored within a data set

repository 204 of the SQE 201. A subscriber 203 to a document service submits natural language queries to the SQE 201, typically using a visual interface. The queries are then processed by the SQE 201 against the data sets stored in the data set repository 204. The query results are then returned to the subscriber 203. In this example, the SQE 201 is shown implemented as part of a subscription document service, although one skilled in the art will recognize that the SQE may be made available in many other forms, including as a separate application/tool, integrated into other software or hardware, for example, cell phones, personal digital assistants ("PDA"), or handheld computers, or associated with other types of services. Additionally, although the example embodiment is shown and described as processing data sets and natural language queries that are in the English language, as discussed earlier, one skilled in the art will recognize that the SQE can be implemented to process data sets and queries of any language, or any combination of languages.

Figure 3 is an example flow diagram of the steps performed by a Syntactic Query Engine to process data sets and natural language queries. In step 301, the SQE receives a data set, for example, a set of documents. In step 302, the SQE preprocesses the data set to ensure a consistent data format. In step 303, the SQE parses the data set, identifying the syntax and grammatical roles of terms within the data set and transforming the data to a normalized data structure. In step 304, the SQE stores the parsed and transformed data set in a data set repository. After a data set is stored, the SQE can process natural language queries against the data set. In step 305, the SQE receives a natural language query, for example, through a user interface. In step 306, the SQE preprocesses the received natural language query, formatting the query as appropriate to be parsed. In step 307, the SQE parses the formatted query, identifying the syntactic and grammatical roles of terms in the query and transforming the query into a normalized data structure. Using the parsed query, in step 308, the SQE generates and submits data queries (*e.g.*, SQL statements) against the data set stored in the data set repository. Finally, in step 309, the SQE returns the results of the natural language query, for example, by displaying them through a user interface.

Figures 4 through 15 are example screen displays of an example Syntactic Query Engine user interface for submitting natural language queries and viewing query results. Figure 4 is an example screen display illustrating

general search functionality of an example Syntactic Query Engine user interface. The general search functionality allows a user, for example, to submit a natural language query in the form of a sentence, which may be a statement or a question. The user enters a query in query box 401 and, using the search button 403, submits the query to the SQE, for example, the SQE of Figure 2. The SQE displays the query results in results area 405. Each result that is returned by the SQE after performing a general search is a sentence extracted from the data set. Depending on the data set, other information may also be displayed with each result. For example, additional information displayed with the results shown in Figure 4 include the document title (*e.g.*, "Foreign reserves & the exchange rate"), the name of the country the document relates to (*e.g.*, "Somalia"), the date associated with the document (*e.g.*, "[28-DEC-1997]"), and the location (*e.g.*, directory) where the original source document is stored. One skilled in the art will recognize that depending on the type of data that comprises the data set, different types of related information may be displayed as part of a result in a result set. Each displayed result is also a link that, when selected, causes the SQE user interface to display a larger portion of the data set from which the selected result was extracted. For example, if the sentence 406 shown as,

> The now defunct Central Bank of Somalia suffered a setback after the fall of Mr. Siad Barre in 1991 when a reported $70m in foreign exchange disappeared,

is selected, the SQE will display a portion of the source document 407, labeled,

> Foreign reserves & the exchange rate,

from which that sentence was retrieved. Although this example refers to a "user" submitting natural language queries, one skilled in the art will recognize that natural language queries may be submitted to an SQE through means other than user input. For example, an SQE may receive input from another program executing on the same computer or, for example, across a network or from a wireless PDA device.

Figure 5A is an example screen display illustrating a portion of a data set from which a natural language query result was extracted. The portion displayed in text area 5A05 typically reflects the selected sentence that resulted from the query as well as a number of surrounding sentences in the document. Additional options are also available while viewing portions of the data set. For example, the user may select a sentence in text area 5A05 and select option

"Sentences" 5A01 from the Search Similar Menu 5A04, causing the SQE to perform a syntactic search (using the search and indexing techniques described herein) against the data set using the selected sentence as a new natural language query. The search similar sentence functionality is described in detail with reference to Figures 5B and 5C. Alternatively, a user may select an entire paragraph and select option "Paragraphs" 5A02 from the Search Similar Menu 5A04. Selecting the Search Similar Paragraphs option causes the SQE to perform a search against the data set to return other paragraphs within the data set that contain content similar to the selected paragraph. The user may also select option "Documents" 5A03 from the Search Similar Menu 5A04, causing the SQE to perform a search against the data set to return other documents within the data set that contain content similar to the selected document. In an exemplary embodiment of an SQE, the paragraph and document similarity searches are preferably performed using latent semantic regression techniques as described in U.S. Patent Application No. _____, filed on September 25, 2001 and entitled "An Inverse Inference Engine for High Performance Web Search," which is a continuation-in-part of U.S. Application No. 09/532,605 filed March 22, 2000, and claims priority from U.S. Provisional Application No. 60/235,255, filed on September 25, 2000. One skilled in the art will recognize that other types of searches, including syntactic searches as described herein, may be implemented for the "Sentences," "Documents," and "Paragraphs" options. For example, a keyword search or one or more syntactic searches may be used.

Figure 5B is an example screen display illustrating a search similar sentence operation. In Figure 5B, a sentence is selected within a portion of a data set from which a natural language query result was extracted. Selecting the sentence 5B06 and then selecting option "Sentences" 5B01 from the Search Similar Menu 5B04 causes the SQE to perform a syntactic search against the data set, returning results that are similar to the selected sentence 5B06. Figure 5C is an example screen display illustrating results that correspond to the search similar sentence operation initiated in Figure 5B. From the results displayed, one can see that the resulting sentences relate in a manner that goes beyond word or pattern matching.

Although discussed herein primarily with respect to documents, the SQE supports data sets comprising a variety of objects and data formats and is not limited to textual documents. For example, a data set may comprise

text documents of various formats, with or without embedded non-textual entities (*e.g.*, images, charts, graphs, maps, etc.), as well as documents that are in and of themselves non-textual entities. Figures 6 through 9 illustrate support of natural language queries that request specific non-textual data. One skilled in the art will recognize that support for other format combinations of stored and requested data are contemplated.

Figure 6 is an example screen display illustrating Syntactic Query Engine results from a natural language query that requests a map. When the user selects one of the returned results, for example, the "Angola [icon]" 608, the requested map is displayed. Figure 7 is an example screen display illustrating a map that corresponds to the query result selected in Figure 6.

Figure 8 is an example screen display illustrating Syntactic Query Engine results from a natural language query that requests a chart. When the user selects one of the returned results, for example, the "China [icon] " 808, the requested chart is displayed. Figure 9 is an example screen display illustrating a chart that corresponds to the query result selected in Figure 8.

In addition to queries that are formulated as one or more sentences, the SQE supports searching based upon designated syntactic and/or grammatical roles. The SQE advanced search functionality supports natural language queries that specify one or more terms, each associated with a grammatical role or part of speech.

Figure 10 is an example screen display of an advanced search using a natural language query that contains a subject. From this screen display, a user may enter a word or phrase as a subject 1001 and/or as an object 1002, the "subject" and "object" each being a grammatical role. Selecting the Search button 1003 submits the query to the SQE. In the example shown, a user enters "Bill Clinton" as the subject. When the user selects the Search button 1003, the SQE performs a syntactic search, returning a list of verbs from sentences within the data set which contain "Bill Clinton" as a subject.

Figure 11 is an example screen display illustrating advanced search results from a query that contains a subject. The results are displayed as a list of the verbs found in sentences within the stored data set in which the designated subject occurs as an identified subject of the sentence. The number in brackets after each listed verb indicates the number of times the designated subject and listed verb are found together in a sentence within the stored data

set. For example, the top entry of the middle column 1101 of the verb list indicates that the SQE identified 16 sentences within the data set which contain "Bill Clinton" as a subject and "visit" (or some form of the verb "visit") as the verb. The SQE determines the order in which the verb list is displayed. As illustrated in Figure 11, the verb list may be displayed according to decreasing frequency of occurrence of the designated subject and/or object in the data set. In an alternate embodiment, the verbs may be displayed in alphabetical order. In another embodiment, the verbs may be grouped and/or ordered based on the similarity of meanings among the displayed verbs. The similarity of meanings among multiple verbs can be determined using an electronic dictionary, for example, WordNet. The WordNet dictionary is described in detail in Christiane Fellbaum (Editor) and George Miller (Preface), *WordNet (Language, Speech, and Communication)*, MIT Press, May 15, 1998.

When the user selects a verb from the returned list, the SQE returns a set of resulting sentences with the designated subject and selected verb. Figure 12 is an example screen display illustrating a portion of resulting sentences returned by a Syntactic Query Engine when a particular verb is selected from the verb list. In the example shown, the verb "visit" has been selected from the verb list shown in Figure 11, causing the SQE to display the 16 sentences within the data set where "Bill Clinton" is found as a subject and "visit" is found as the verb. The SQE identifies verbs without regard to specific tense, which enhances the contextual accuracy of the search. For example, if the user selects "visit" as the verb, the SQE returns sentences that contain the verb "visited" (*e.g.*, results 1-4 and 6), and sentences that contain the verb "may visit" (*e.g.*, result 5). The results displayed in the results area 1205 are displayed in the same format as the results displayed in the results area 405 of Figure 4. Accordingly, selecting one of the displayed results causes the SQE to display a larger portion of the data set from which the selected result was extracted, as described with reference to Figure 5A.

As described, with reference to Figure 10, a user may designate a subject and/or an object when using the advanced search functionality. Figure 13 is an example screen display of advanced search functionality using a natural language query that contains a subject and an object. In Figure 13, the user designates "US" as a subject in subject field 1301 and "Mexico" as an object in object field 1302. When the user selects the Search button 1303, the SQE performs a syntactic search against the data set for sentences in which

"US" appears as an identified subject and "Mexico" appears as an identified object.

Figure 14 is an example screen display illustrating advanced search results from a query that contains a subject and an object. The top half 1406 of results area 1405 lists verbs returned where the designated subject appears as a subject of the sentence and the designated object appears as an object of the sentence. The lower half 1407 of results area 1405 lists verbs returned where the designated object appears as a *subject* of the sentence and the designated subject appears as an *object* of the sentence. Thus, the lower half 1407 displays results of a query using the inverse relationship between the subject and object. In this specific example, the top half 1406 of results area 1405 lists verbs found in sentences which contain "US" as a subject and "Mexico" as an object. The bottom half 1406 of results area 1408 lists verbs found in sentences which contain "Mexico" as a subject and "US" as an object. Returning results related to the inverse relationship can be useful because sentences within the data set that are contextually accurate results to the natural language query may be overlooked if the inverse relationship is not examined. As similarly described with reference to Figures 11 and 12, when the user selects a verb from the returned list, the SQE returns a set of resulting sentences with the designated subject, designated object, and selected verb. The resulting sentences are displayed in the same format as the results displayed in the results area 405 of Figure 4 and the results displayed in the results area 1205 of Figure 12. Accordingly, selecting one of the displayed sentences causes the SQE to display a larger portion of the data set from which the selected sentence was extracted, as described with reference to Figure 5A.

Although Figures 10-14 illustrate advanced search functionality with reference to natural language queries that specify a subject and/or an object, one skilled in the art will recognize that a multitude of syntactic and grammatical roles, and combinations of syntactic and grammatical roles may be supported. For example, some of the combinations contemplated for support by the advanced search functionality of the SQE are:

subject/object;
subject/verb/object;
subject/verb;
verb/object;
preposition/verb modifier/object;
verb/verb modifier/object;
verb/preposition/object;

verb/preposition/verb modifier/object;
subject/preposition/verb modifier;
subject/preposition/verb modifier/object;
subject/verb/verb modifier/object;
subject/verb/preposition;
subject/verb/preposition/object;
subject/verb/preposition/verb modifier;
subject/ verb/preposition/verb modifier/object; and
noun/noun modifier.

Such support includes locating sentences in which the designated terms appear in the associated designated syntactic or grammatical role, as well as locating, when contextually appropriate, sentences in which the designated terms appear but where the designated roles are interchanged. For example, as described above, it is contextually appropriate to interchange the grammatical roles of a designated subject and a designated object.

In addition to indexing and searching based on grammatical roles, the Syntactic Query Engine may be implemented to recognize any number of programmable attributes in natural language queries and data sets (described in detail as "preferences" with reference to Figure 15). In one embodiment, these attributes are used to filter the results of a syntactic search. Example attributes include the names of countries, states, or regions, dates, and document sections. One skilled in the art will recognize that an unlimited number of attributes may be defined and may vary across multiple data sets. For example, one data set may consist of text from a set of encyclopedias. For such a data set, a "volume" attribute may be defined, where there is one volume for each letter of the alphabet. A second data set may be a single book with multiple chapters. A "chapter" attribute may be defined, for the data set, allowing a user to search specific chapters.

Figure 15 is an example screen display illustrating the designation of programmable attributes in a Syntactic Query Engine. A user designates various attributes on the preferences window 1501. The SQE stores these attributes (preferences) when the user selects the Set Preferences button 1502. The attribute values are used by the SQE as filters when performing subsequent queries. For example, the user may select a specific country as country attribute 1503. When the SQE performs a subsequent natural language query, the results returned are only those sentences found in documents within the data set that relate to the country value specified as country attribute 1503.

A more detailed description of an example SQE illustrating additional user interface screens and example natural language queries and results is included in Appendix A.

An SQE as described may perform multiple functions (*e.g.*, data set parsing, data set storage, natural language query parsing, and data query processing) and typically comprises a plurality of components. Figure 16 is a block diagram of the components of an example embodiment of a Syntactic Query Engine. A Syntactic Query Engine comprises a Query Preprocessor, a Data Set Preprocessor, a Query Builder, a Data Set Indexer, an Enhanced Natural Language Parser ("ENLP"), a data set repository, and, in some embodiments, a user interface. The Data Set Preprocessor 1603 converts received data sets to a format that the Enhanced Natural Language Parser 1604 recognizes. The Query Preprocessor 1610 converts received natural language queries to a format that the Enhanced Natural Language Parser 1604 recognizes. The Enhanced Natural Language Parser ("ENLP") 1604, parses sentences, identifying the syntax and grammatical role of each meaningful term in the sentence and the ways in which the terms are related to one another and transforming the sentences into a canonical form—an enhanced data representation. The Data Set Indexer 1607 indexes the parsed data set and stores it in the data set repository 1608. The Query Builder 1611 generates and executes formatted queries (*e.g.*, SQL statements) against the data set indexed and stored in the data set repository 1608.

In operation, the SQE 1601 receives as input a data set 1602 to be indexed and stored. The Data Set Preprocessor 1603 prepares the data set for parsing by assigning a Document ID to each document that is part of the received data set, performing OCR processing on any non-textual entities that are part of the received data set, and formatting each sentence according to the ENLP format requirements. The Enhanced Natural Language Parser ("ENLP") 1604 parses the data set, identifying for each sentence, a set of terms, each term's part of speech and associated grammatical role and transforming this data into an enhanced data representation. The Data Set Indexer 1607 formats the output from the ENLP and sends it to the data set repository 1608 to be indexed and stored. After a data set is indexed, a natural language query 1609 may be submitted to the SQE 1601 for processing. The Query Preprocessor 1610 prepares the natural language query for parsing. The preprocessing may include, for example, spell checking, verifying that there is only one space

between each word in the query, and identifying the individual sentences if the query is made up of more than one sentence. One skilled in the art will recognize that the steps performed by the Data Set Preprocessor or the Query Preprocessor may be modified based on the requirements of the natural language parser. Any preprocessing steps necessary to prepare a data set or a natural language query to be parsed are contemplated for use with techniques of the present invention. The ENLP 1604 then parses the preprocessed natural language query transforming the query into the canonical form and sends its output to the Query Builder. The Query Builder 1611 uses the ENLP 1604 output to generate one or more data queries. Data queries differ from natural language queries in that they are in a format specified by the data set repository, for example, SQL. The Query Builder 1611 may generate one or more data queries associated with a single natural language query. The Query Builder 1611 executes the generated data queries against the data set repository 1608 using well-known database query techniques and returns the data query results as Natural Language Query Results 1612. Note that when the SQE is used within a system that interfaces with a user, the SQE also typically contains a user interface component 1613. The user interface component 1613 interfaces to a user in a manner similar to that shown in the display screens of Figures 4-15.

Figure 17 is a block diagram of the components of an Enhanced Natural Language Parser of an example embodiment of a Syntactic Query Engine. The Enhanced Natural Language Parser ("ENLP") 1701 comprises a natural language parser 1702 and a postprocessor 1703. The natural language parser 1702 identifies, for each sentence it receives as input, the part of speech for each term in the sentence and syntactic relationships between the terms within the sentence. An SQE may be implemented by integrating a proprietary natural language parser into the ENLP, or by integrating an existing off-the-shelf natural language parser, for example, Minipar, available from Nalante, Inc., 245 Falconer End, Edmonton, Alberta, T6R 2V6. The postprocessor 1703 examines the natural language parser 1702 output and, from the identified parts of speech and syntactic relationships, determines the grammatical role played by each term in the sentence and the grammatical relationships between those terms. The postprocessor 1703 then generates an enhanced data representation from the determined grammatical roles and syntactic and grammatical relationships.

Figure 18 is a block diagram of the processing performed by an example Enhanced Natural Language Parser. The natural language parser 1801 receives a sentence 1803 as input, and generates a syntactic structure, such as parse tree 1804. The generated parse tree identifies the part of speech for each term in the sentence and describes the relative positions of the terms within the sentence. The postprocessor 1802 receives the generated parse tree 1804 as input and determines the grammatical role of each term in the sentence and relationships between terms in the sentence, generating an enhanced data representation, such as enhanced sentence representation 1805.

Figure 19 is a block diagram illustrating a graphical representation of an example syntactic structure generated by the natural language parser component of an Enhanced Natural Language Parser. The parse tree shown is one example of a representation that may be generated by a natural language parser. The techniques of the methods and systems of the present invention, implemented in this example in the postprocessor component of the ENLP, enhance the representation generated by the natural language processor by determining the grammatical role of each meaningful term, associating these terms with their determined roles and determining relationships between terms. In Figure 19, the top node 1901 represents the entire sentence, "YPF of Argentina exports natural gas" Nodes 1902 and 1903 identify the noun phrase of the sentence, "YPF of Argentina," and the verb phrase of the sentence, "exports natural gas," respectively. The branches of nodes or leaves in the parse tree represent the parts of the sentence further divided until, at the leaf level, each term is singled out and associated with a part of speech. A configurable list of words are ignored by the parser as "stopwords." The stopword list comprises words that are deemed not indicative of the information being sought. Example stopwords are "a," "the," "and," "or," and "but." In one embodiment, question words (*e.g.*, "who," "what," "where," "when," "why," "how," and "does") are also ignored by the parser. In this example, nodes 1904 and 1905 identify the noun phrase 1902 as a noun, "YPF" and a prepositional phrase, "of Argentina." Nodes 1908 and 1909 divide the prepositional phrase 1905 into a preposition, "of," and a noun, "Argentina." Nodes 1906 and 1907 divide the verb phrase 1903 into a verb, "exports;" and a noun phrase, "natural gas." Nodes 1910 and 1911 divide the noun phrase 1907 into an adjective, "natural," and a noun, "gas."

Figure 20 is a table illustrating an example enhanced data representation generated by the postprocessor component of an Enhanced Natural Language Parser. This example enhanced data representation comprises nine different ways of relating terms within the sentence that was illustrated in the parse tree of Figure 19. The ways chosen and the number of ways used is based upon a set of heuristics, which may change as more knowledge is acquired regarding syntactic searching. In addition, one skilled in the art will recognize that the selected roles and relationships to be stored may be programmatically determined. In the example shown, row 2001 represents the relationship between "Argentina" as the subject of the sentence and "YPF" as a modifier of that subject. The SQE determines this relationship based on the location of the preposition, "of" between the two related terms in the sentence. Rows 2002 and 2003 represent the relationship between "YPF" as the subject of the sentence and "natural gas" and "gas," respectively, as objects of the sentence. Similarly, rows 2004 and 2005 represent the relationship between "Argentina" as the subject of the sentence and "natural gas" and "gas," respectively, as objects of the sentence. Rows 2006 and 2007, respectively, represent the relationship between the two nouns, "YPF" and "Argentina," each used as a subject and the verb "export." Rows 2008 and 2009 represent the relationship between the verb, "export" and the noun phrase, "natural gas" and the noun, "gas," each used as an object, respectively.

The enhanced data representation is indexed and stored to support the syntactic search functionality of the SQE. The original sentence "YPF of Argentina exports natural gas," will be returned by the SQE as a query result in response to any submitted query that can be similarly represented. For example, "What countries export gas?," "Does Argentina import gas?," and "Is Argentina an importer or exporter of gas?" will all cause the SQE to return to the represented sentence as a result.

The Syntactic Query Engine performs two functions to accomplish effective syntactic query processing. The first is the parsing, indexing, and storage of a data set. The second is the parsing and subsequent execution of natural language queries. These two functions are outlined below with reference to Figures 21 and 22.

Figure 21 is an example block diagram of data set processing performed by a Syntactic Query Engine. As an example, documents that make up a data set 2101 are submitted to the Data Set Preprocessor 2102 (*e.g.*,

component 1603 in Figure 16). If the data set comprises multiple files, as shown in Figure 21, the Data Set Preprocessor 2102 creates one tagged file containing the document set. The Data Set Preprocessor 2102 then dissects that file into individual sentences and sends each sentence to the ENLP 2104 (*e.g.*, component 1604 in Figure 16). After the ENLP 2104 parses each received sentence, it sends the generated enhanced data representation of each sentence to the Data Set Indexer 2105 (*e.g.*, component 1607 in Figure 16). The Data Set Indexer 2105 processes and formats the ENLP output, distributing the data to formatted text files. The text files are typically bulk loaded into the data set repository 2107 (*e.g.*, component 1608 in Figure 16). One skilled in the art will recognize that other methods of data set preprocessing, indexing, and storing may be implemented in place of the methods described herein, and that such modifications are contemplated by the methods and systems of the present invention. For example, the Data Set Indexer may insert data directly into the data set repository instead of generating text files to be bulk loaded.

After indexing and storing a data set, the SQE may perform its second function, processing natural language queries against the stored data set. Figure 22 is an example block diagram of natural language query processing performed by a Syntactic Query Engine. As an example, a natural language query 2201 is submitted to the Query Preprocessor 2202 of the SQE. The Query Preprocessor 2202 (*e.g.*, component 1610 in Figure 16) prepares the natural language query for parsing. The preprocessing step may comprise several functions, examples of which may be spell checking, text case verification and/or alteration, and excessive white-space reduction. The specific preprocessing steps performed by the Query Preprocessor 2202 are typically based on the format requirements of the natural language parser component of the ENLP. The SQE sends the preprocessed query to the ENLP 2204 (*e.g.*, component 1604 in Figure 16). The ENLP parses the query, generating an enhanced data representation of the query 2205, which is sent to the Query Builder 2206 (*e.g.*, component 1611 in Figure 16). This enhanced data representation identifies grammatical roles of and relationships between terms in the query. Using the enhanced data representation 2205, the Query Builder 2206 generates one or more data queries 2207 and executes them against the data set repository 2208. The data query results 2209 are returned to the

Query Builder to be returned to the user as natural language query results 2210.

Figure 23 is an example block diagram of a general purpose computer system for practicing embodiments of a Syntactic Query Engine. The computer system 2301 contains a central processing unit (CPU) 2302, Input/Output devices 2303, a display device 2304, and a computer memory (memory) 2305. The Syntactic Query Engine 2320 including the Query Preprocessor 2306, Query Builder 2307, Data Set Preprocessor 2308, Data Set Indexer 2311, Enhanced Natural Language Parser 2312, and data set repository 2315, preferably resides in memory 2305, with the operating system 2309 and other programs 2310 and executes on CPU 2302. One skilled in the art will recognize that the SQE may be implemented using various configurations. For example, the data set repository may be implemented as one or more data repositories stored on one or more local or remote data storage devices. Furthermore, the various components comprising the SQE may be distributed across one or more computer systems including handheld devices, for example, cell phones or PDAs. Additionally, the components of the SQE may be combined differently in one or more different modules. The SQE may also be implemented across a network, for example, the Internet or may be embedded in another device.

As described with reference to Figure 21, the Data Set Preprocessor 2102 performs two overall functions – building one or more tagged files from the received data set files and dissecting the data set into individual objects, for example, sentences. These functions are described in detail below with respect to Figures 24-26. Although Figures 24-26 present a particular ordering of steps and are oriented to a data set of objects comprising documents and queries comprising sentences, one skilled in the art will recognize that these flow diagrams, as well as all others described herein, are examples of one embodiment. Other sequences, orderings and groupings of steps, and other steps that achieve similar functions, are equivalent to and contemplated by the methods and systems of the present invention. These include steps and ordering modifications oriented toward non-textual objects in a data set, such as audio or video objects.

Figure 24 is an example flow diagram of the steps performed by a build_file routine within the Data Set Preprocessor component of a Syntactic Query Engine. The build_file routine generates text for any non-textual entities

within the dataset, identifies document structures (*e.g.*, chapters or sections in a book), and generates one or more tagged files for the data set. In one embodiment, the build_file routine generates one tagged file containing the entire data set. In alternate embodiments, multiple files may be generated, for example, one file for each object (*e.g.*, document) in the data set. In step 2401, the build_file routine creates a text file. In step 2402, the build_file routine determines the structure of the individual elements that make up the data set. This structure can be previously determined, for ,example by a system administrator and indicated within the data set using, for example, HTML tags. For example, if the data set is a book, the defined structure may identify each section or chapter of the book. In step 2403, the build_file routine tags the beginning and end of each document (or section, as defined by the structure of the data set). In step 2404, the routine performs OCR processing on any images so that it can create searchable text (lexical units) associated with each image. In step 2405, the build_file routine creates one or more sentences for each chart, map, figure, table, or other non-textual entity. For example, for a map of China, the routine may insert a sentence of the form,

This is a map of China.

In step 2406, the build_file routine generates an object identifier (*e.g.*, (a Document ID) and inserts a tag with the generated identifier. In step 2407, the build_file routine writes the processed document to the created text file. Steps 2402 through 2407 are repeated for each file that is submitted as part of the data set. When there are no more files to process, the build_file routine returns.

Figure 25 illustrates an example format of a tagged file built by the build_file routine of the Data Set Preprocessor component of a Syntactic Query Engine. The beginning and end of each document in the file is marked, respectively, with a <DOC> tag 2501 and a </DOC> tag 2502. The build_file routine generates a Document ID for each document in the file. The Document ID is marked by and between a <DOCNO> tag 2503 and a </DOCNO> tag 2504. Table section 2505 shows example sentences created by the build_file routine to represent lexical units for a table embedded within the document. The first sentence for Table 2505,

This table shows the Defense forces, 1996,

is generated from the title of the actual table in the document. The remaining sentences shown in Table 2505, are generated from the rows in the actual table in the document. Appendix B is a portion of a sample file created by the

build_file routine. One skilled in the art will recognize that various processes and techniques may be used to identify documents within the data set and to identify entities (*e.g.*, tables) within each document. The use of equivalent and/or alternative processes and markup techniques and formats, including HTML, XML, and SGML and non-tagged techniques are contemplated and may be incorporated in methods and systems of the present invention.

The second function performed by the Data Set Preprocessor component of the SQE is dissecting the data set into individual objects (*e.g.*, sentences) to be processed. Figure 26 is an example flow diagram of the steps performed by the dissect_file routine of the Data Set Preprocessor component of a Syntactic Query Engine. In step 2601, the routine extracts a sentence from the tagged text file containing the data set. In step 2602, the dissect_file routine preprocesses the extracted sentence, preparing the sentence for parsing. The preprocessing step may comprise any functions necessary to prepare a sentence according to the requirements of the natural language parser component of the ENLP. These functions may include, for example, spell checking, removing excessive white space, removing extraneous punctuation, and/or converting terms to lowercase, uppercase, or proper case. One skilled in the art will recognize that any preprocessing performed to put a sentence into a form that is acceptable to the natural language parser can be used with techniques of the present invention. In step 2603, the routine sends the preprocessed sentence to the ENLP. In step 2604, the routine receives as output from the ENLP an enhanced data representation of the sentence. In step 2605, the dissect_file routine forwards the original sentence and the enhanced data representation to the Data Set Indexer for further processing. Steps 2601-2605 are repeated for each sentence in the file. When no more sentences remain, the dissect_file routine returns.

The Data Set Indexer (*e.g.*, component 2105 in Figure 21) prepares the enhanced data representations generated from the data set (*e.g.*, the enhanced sentence representation illustrated in Figure 20) to be stored in the data set repository. In one example embodiment, the Data Set Indexer initially stores the enhanced data representation data in generated text files before loading the enhanced data representations into the data set repository, for example, using a bulk loading function of the data set repository. One skilled in the art will recognize that any of a wide variety of well-known techniques may be implemented to load a data set (including the generated

enhanced data representations) into the data set repository. For example, another technique for loading writes each record to the data set repository as it is generated instead of writing the records to text files to be bulk loaded.

As described, the SQE uses the ENLP to parse data that is being stored and indexed, as well as to parse queries (*e.g.*, natural language queries) that are submitted against a stored indexed data set. Similar to the preprocessing performed before parsing a data set, the SQE performs preprocessing on submitted queries.

Figure 27 is an example flow diagram of the steps performed by a Syntactic Query Engine to process a natural language query. In step 2701, the Query Preprocessor prepares the natural language query for the ENLP. In step 2702, the ENLP parses the preprocessed natural language query and generates an enhanced data representation of the query. In step 2703, the Query Builder, generates and executes data queries (*e.g.*, SQL statements) based on the ENLP output (the enhanced data representation).

Figure 28 is an example flow diagram of the steps performed by a preprocess_natural_language_query routine of the Query Preprocessor component of a Syntactic Query Engine. This routine preprocesses the query according to the requirements of the ENLP. Although described with respect to certain modifications to the query, one skilled in the art will recognize that many other preprocessing steps are possible, yield equivalent results, and are contemplated by the methods and systems of the present invention. In step 2801, the routine separates the query into multiple sentences if necessary, based on punctuation (*e.g.*, ";", ":", "?", "!", and ".", where the "." is not part of an abbreviation). In step 2802, the routine removes spaces between any terms that are separated by hyphens. For example, "seventeen – year – old" is converted to "seventeen-year-old". In step 2803, the routine removes extraneous spaces from the query, leaving the words separated by one space. In step 2804, the routine spell-checks the routine. In one embodiment, the routine automatically corrects any detected spelling errors. In another embodiment, the routine requests an indication of whether and how each spelling error is to be corrected.

The enhanced data representations describe one or more ways in which the meaningful terms of an object (*e.g.*, a sentence) may be related. (See description with reference to Figure 20.) As described earlier, enhanced data representations of the data in the data set are stored in the data set

repository, and enhanced data representations are used to generate data queries when the SQE processes a natural language query. The enhanced data representations describe the relationships between meaningful terms within each sentence, as determined by the ENLP. Each meaningful term may be associated with one or more syntactic or grammatical roles. Redundancy is not discouraged, because it may yield additional results. For example, a term may be part of two described relationships that are associated with different grammatical roles, *e.g.*, a term may appear as both an "object" and a "noun modifier." The relationships that are selected and represented are heuristically determined as those relationships that will tend to generate additional relevant results. A current embodiment uses the specific relationships described in Figure 20 and shown as stored in the data repository in Figure 29 and described by Figures 31-36. However, one skilled in the art will recognize that other relationships and grammatical roles may be described in the enhanced data representation, and the determination of these roles and relationships relates to the types of objects in the data set.

Figure 29 is an example block diagram showing the structure of an example Data Set Repository of a Syntactic Query Engine. The set of stored tables represent the roles and relationships between the determined meaningful terms for each parsed sentence of each document in the data set, as determined by the ENLP, and correspond to the enhanced data representations generated. The Subject Table 2901 stores one record for each determined subject/verb combination in each parsed sentence in each document. The Object Table 2902 stores each determined object/verb combination for each parsed sentence in each document. The Subject_Object table stores each identified subject/verb/object combination for each parsed sentence. In an alternate embodiment, the Subject_Object table is implemented as a view that is a dynamically maintained join between the subject and object tables, joined on the verb, Document ID, and Sentence ID fields. The Preposition table 2903 stores each verb/preposition/verb modifier combination for each parsed sentence in each document. The Noun_Modifier table 2904 stores each noun/noun modifier combination in each parsed sentence in each document. A noun modifier may be a noun, a noun phrase, or an adjective. The Sentence table 2905 stores the actual text for each sentence in each document. In addition, the Sentence table 2905 stores the Governing Verb, Related Subject, and Related Object of each sentence, as identified by

the postprocessor component of the ENLP. The governing verb is the main verb in a sentence. The related subject and related object are the subject and object, respectively, related to the governing verb. For example, in the sentence

The girl walks the dog.

"walks" is the governing verb, "girl" is the related subject, and "dog" is the related object. These fields may be left blank if the postprocessor is unable to determine the governing verb, related subject, or related object. The Date, Money Amount, Number, Location, Person, Corporate Name, and Organization fields of the Sentence table 2905 store binary indicators of whether or not the sentence contains a term that the SQE recognizes as an attribute of the attribute type indicated by the field name. The Attributes table 2906 is an optional table that stores the values of specific data types found within each document. As described above, attributes are settable parameters and may include, for example, names of countries, states, or regions, document sections, and dates. As described with respect to Figure 15, these attributes may be used by the SQE to filter data query results. The optional Parent table 2907 is used to indicate a hierarchical structure of objects in the data set. This allows a section, subsection, chapter, or other document portion to be identified by the Data Set Indexer component of a Syntactic Query Engine as a "document", while storing the relationships between multiple documents or document portions in a hierarchical fashion.

Figure 30 is an example flow diagram of the steps performed by a parse_sentence routine of the Enhanced Natural Language Parser component of a Syntactic Query Engine. In summary, the routine parses the designated sentence or phrase, identifies the grammatical roles of terms within the sentence, generates an enhanced data representation of the sentence, and constructs an output string. In step 3001, the natural language parser component of the ENLP parses the preprocessed sentence. In step 3002, the parse_sentence routine calls the determine_grammatical_roles subroutine (discussed in detail with reference to Figure 31) to determine the grammatical roles of and relationships between terms within the sentence. In step 3004, the routine calls the construct_output_string routine (discussed in detail with reference to Figure 37) to format the generated enhanced data representation for further processing.

Figure 31 is an example flow diagram of the steps performed by a determine_grammatical_roles subroutine within the parse_sentence routine of the Enhanced Natural Language Parser. In summary, the routine converts terms, as appropriate, to a standard form (*e.g.*, converting all verbs to active voice), identifies attributes (*e.g.*, names of countries) within the sentence, determines the grammatical roles of meaningful terms in the sentence, and initiates the generation of an enhanced data representation. In step 3101, the routine converts each word that is identified as a subordinate term to an associated governing term. Subordinate terms and governing terms are terms that are related in some way, similar to multiple tenses of a verb. For example, the governing term "Ireland" is associated with subordinate terms "Irish," "irish," "Irishman," "irishman," "Irishwoman," "irishwoman," and "ireland." Converting subordinate terms to an associated governing term ensures that a standard set of terms is used to represent data, for example relating multiple terms to a country, as shown in the example above, thus increasing potential contextual matches. Subordinate terms that are identified as occurring within a noun phrase, for example, "Korean," occurring with the noun phrase "Korean War," are not converted to the associated governing term to preserve the specific meaning of the noun phrase. For example, "Korean War" has a specific meaning that is not conveyed by the terms "Korea" and "war" independently. Appendix C is an example list of subordinate terms and their associated governing terms for an example SQE. Typically, this information is stored by the SQE as a configurable list that is related to the data set, preferably initialized when the SQE is installed, for example by a system administrator.

In step 3102, the determine_grammatical_roles routine converts the verbs within the sentence to active voice. This ensures that the SQE will identify data within the data set in response to queries regardless of verb tense. (See the example described with reference to Figure 1.) In step 3103, the routine identifies attribute values, which can later be used to filter search results.

Steps 3104-3111 are described with reference to an example query,

Does Argentina import or export natural gas from the south of Patagonia? (Q)

Figure 36A is a graphical representation of an example parse tree generated by a natural language parser component of an Enhanced Natural Language Parser. The example parse corresponds to this query. An enhanced data representation of the example query is shown in Figure 36B.

In steps 3104-3111, the determine_grammatical_roles routine builds the data structures that correspond to the enhanced data representation. Specifically, in step 3104, the routine generates Subject data structures identifying terms that may be the subject of the sentence (similar to the Subject table described with reference to Figure 29). For example, given query Q above, Table 1 shows the generated Subject structures.

Table 1

Subject	Verb
Argentina	import
Argentina	export

The steps performed in generating the Subject structures are described in detail with reference to Figure 32. In step 3105, the routine generates Object data structures identifying terms that may be the object of the sentence (similar to the Object table described with reference to Figure 29). For example, given query Q above, Table 2 shows the generated Object structures.

Table 2

Verb	Object
import	gas
import	natural gas
export	gas
export	natural gas

The steps performed in generating the Object structures are described in detail with reference to Figure 33. In step 3106, the routine generates Preposition structures that are similar to the structure of the Preposition table described with reference to Figure 29. For example, given query Q above, Table 3 shows the generated Preposition structures.

Table 3

Verb	Preposition	Modifier
import	from	south
import	from	Patagonia
export	from	south
export	from	Patagonia

In step 3107, the routine generates Subject/Object structures to represent the ways in which terms that are identified as potential subjects and objects of a sentence may be related. Each subject is paired with each object that is associated with the same verb. In addition, each subject is paired with each modifier in the Preposition structure that is associated with the same verb. For example, given query Q above, Table 4 shows the generated Subject/Object structures.

Table 4

Subject	Object
Argentina	gas
Argentina	natural gas
Argentina	south
Argentina	Patagonia

In step 3108, the routine generates Subject/Modifier structures to describe the relationships between related nouns, noun phrases, and adjectives. The generate_subject_modifier routine is described in detail with reference to Figure 34.

In step 3109, the routine determines whether or not the sentence being parsed is a query (as opposed to an object of a data set being parsed for storing and indexing). If the designated sentence or phrase is a query, the routine continues in step 3110, else it returns. In steps 3110 and 3111, the routine generates Generalized Subject/Object structures and Generalzied Subject/Modifier structures. These structures may be used by the Query Builder to generate additional data queries to return results that may be contextually relevant to the submitted natural language query. The generate_generalized_subject_object routine is described in detail with reference to Figure 35. The Generalized Subject/Modifier structures are generated from previously generated Subject/Object structures. The object is

designated as the subject in the new Generalized Subject/Modifier structure, and the subject is designated as the modifier in the new Generalized Subject/Modifier structure.

Figure 32 is an example flow diagram of the steps performed by a generate_subject_structure subroutine of the determine_grammatical_roles routine. This routine identifies, for each verb in the sentence, each noun, noun phrase, or adjective that may be a subject associated with the designated verb and stores it in a Subject structure. In step 3201, the routine searches for all verbs in the syntactic data representation (*e.g.*, a parse tree) generated by the natural language parser. In step 3202, the routine sets the current node to an identified verb. In steps 3203-3206, the routine loops searching for all terms that are potentially subjects related to the identified verb. Specifically, in step 3203, the routine moves toward the beginning of the sentence (*e.g.*, to the next leaf to the left in the parse tree from the current node). In step 3204, the routine examines the node and determines whether or not it is a noun, a noun phrase, or an adjective. If the routine determines that the current node is a noun, a noun phrase, or an adjective, then it is identified as a subject and the routine continues in step 3205, else it continues in step 3206. In step 3205, the routine creates a Subject structure identifying the current node as the subject and the previously identified verb (the current verb) as the verb. Additionally, if the current node is a noun phrase, the routine creates a Subject structure identifying the tail of the noun phrase (*e.g.*, "gas" of the noun phrase "natural gas") as a subject associated with the current verb. The routine then continues searching for additional subjects related to the current verb by looping back to step 3203. In step 3206, the routine examines the current node and determines whether or not it is the left-most leaf or a verb from a different verb phrase. If the current node is not the left-most leaf or a verb from a different verb phrase, the routine continues searching for additional subjects related to the current verb by returning to the beginning of the loop, in step 3203, else it continues in step 3207. In step 3207, the routine determines whether or not all of the identified verbs have been processed. If there are more verbs to process, the routine continues in step 3202 and begins another loop with the new verb as the current node, otherwise it returns.

Figure 33 is an example flow diagram of the steps performed by a generate_object_structure subroutine of the determine_grammatical_roles routine. This routine identifies, for each verb in the sentence, each noun, noun

phrase, or adjective that may be an object associated with the designated verb and stores it in an Object structure. In step 3301, the routine searches for all verbs in the syntactic data representation generated by the natural language parser. In step 3302, the routine sets the current node to an identified verb. In steps 3303-3306, the routine loops searching for all terms that are potentially objects related to the identified verb. Specifically, in step 3303, the routine moves toward the end of the sentenct (*e.g.*, to the next leaf to the right in the parse tree from the current node). In step 3304, the routine examines the node and determines whether or not it is a noun, a noun phrase, or an adjective. If the routine determines that the current node is a noun, a noun phrase, or an adjective, then it is identified as an object and the routine continues in step 3305, else it continues in step 3306. In step 3305, the routine creates an Object structure identifying the current node as the object and the previously identified verb (the current verb) as the verb. Additionally, if the current node is a noun phrase, the routine creates an Object structure identifying the tail of the noun phrase (*e.g.*, "gas" of the noun phrase "natural gas") as the object associated with the current verb. The routine then continues searching for additional objects related to the current verb by looping back to step 3303. In step 3306, the routine examines the current node and determines whether or not it is the right-most leaf, a verb from a different verb phrase, or a preposition other than "of." If the current node is not the right-most leaf, a verb from a different verb phrase, or a preposition other than "of," the routine continues searching for additional objects related to the current verb by returning to the beginning of the loop in step 3303, else it continues in step 3307. In step 3307, the routine determines whether or not all of the identified verbs have been processed. If there are more verbs to process, then the routine continues in step 3302, and begins another loop with the new verb as the current node, otherwise it returns.

Figure 34 is an example flow diagram of the steps performed by a generate_subject_modifier subroutine of the determine_grammatical_roles routine. This routine identifies, for each noun in the sentence, each other noun or adjective that may be related to the designated noun and stores it in a Subject/Modifier structure. In step 3401, the routine searches for all nouns and noun phrases in the syntactic data representation (*e.g.*, a parse tree) generated by the natural language parser. In step 3402, the routine sets the current node to an identified noun or noun phrase. In steps 3403-3406, the routine loops

searching for all terms that may be related to the identified noun. Specifically, in step 3403, the routine moves toward the beginning of the sentenct (*e.g.*, to the next leaf to the left in the parse tree from the current node). In step 3404, the routine examines the node and determines whether or not it is a noun, a noun phrase, or an adjective. If the routine determines that the current node is a noun, a noun phrase, or an adjective, then it is identified as a modifier and the routine continues in step 3405, else it continues in step 3406. In step 3405, the routine creates a Subject/Modifier structure identifying the current node as the modifier and the previously identified noun as the subject. Additionally, if the current node is a noun phrase, the routine creates a Subject/Modifier structure identifying the tail of the noun phrase (*e.g.*, "gas" of the noun phrase "natural gas") as the modifier associated with the current noun. The routine then continues searching for additional modifiers related to the current noun by looping back to step 3403. In step 3406, the routine examines the current node and determines whether or not it is the preposition, "of." If the current node is the preposition "of," the routine continues searching, by returning to the beginning of the loop, in step 3403, else it continues in step 3407. In step 3407, the routine determines whether or not all of the identified nouns have been processed. If there are more nouns to process, the routine continues in step 3402 and begins another loop with the new noun as the current node, otherwise it returns.

As described with reference to Figure 31, the determine_grammatical_roles routine calls the generate_generalized_subject_object and generate_generalized_subject_modifier subroutines (steps 3110 and 3111 of Figure 31) when the sentence being parsed is a query instead of an object in a data set being indexed for storage.

Figure 35 is an example flow diagram of the steps performed by a generate_generalized_subject_object subroutine of the determine_grammatical_roles routine. This routine identifies, for each noun in the sentence, each other noun or adjective that may be an object related to the designated noun and stores it in a Subject/Object structure. In step 3501, the routine searches for all nouns and noun phrases in the syntactic data representation (*e.g.*, a parse tree) generated by the natural language parser. In step 3502, the routine sets the current node to an identified noun (or noun phrase). In steps 3503-3509, the routine loops searching for all terms that may

be an object related to the identified noun. Specifically, in step 3503, the routine moves toward the end of the sentence (*e.g.*, to the next leaf to the right in the parse tree from the current node). In step 3504, the routine examines the node and determines whether or not it is a preposition other than "of." If the routine determines that the current node is a preposition other than "of," then it continues in step 3505, else it continues in step 3508. In step 3508, the routine examines the node and determines whether or not it is a verb or the last node. If the routine determines that the current node is a verb or the last node, then it continues in step 3510, else it continues searching, by returning to the beginning of the loop, in step 3503. In step 3505, the routine moves toward the end of the sentence (*e.g.*, to the next leaf to the right in the parse tree from the current node). In step 3506, the routine examines the current node and determines whether or not it is a noun, a noun phrase, or an adjective. If the routine determines that the current node is a noun, a noun phrase, or an adjective, then it is identified as an object and the routine continues in step 3507, else it continues in step 3509. In step 3507, the routine creates a Generalized Subject/Object structure identifying the current node as the object and the previously identified noun as the subject. Additionally, if the current node is a noun phrase, the routine creates a Subject/Object structure identifying the tail of the noun phrase (*e.g.*, "gas" of the noun phrase "natural gas") as the object associated with the current "subject" node. After creating one or more Generalized Subject/Object structures, the routine continues looping in step 3505, looking for any other nouns, noun phrases, or adjectives to relate to the identified "subject" noun. In step 3509, the routine determines whether or not the current node is the preposition "of." If the current node is the preposition "of," then the routine continues looping in step 3505, else it continues in step 3510. In step 3510, the routine determines whether or not all of the identified nouns and noun phrases have been processed. If there are more nouns or noun phrases to process, the routine continues in step 3502 and begins another loop with the new noun as the current node, otherwise it returns.

Figure 36B is an illustration of an enhanced data representation of an example natural language query generated by an Enhanced Natural Language Parser. The natural language query,

> Does Argentina import or export natural gas from the south of Patagonia?

(query Q) is described by the roles and relationships shown in rows 1-28. Rows 1 and 2 are generated by the generate_subject_structure subroutine described with reference to Figure 32. Rows 3-6 are generated by the generate_object_structure subroutine described with reference to Figure 33. Rows 7-10 are generated by the generate Preposition structures routine described with reference to step 3106 of Figure 31. Rows 11-14 are generated by the generate Subject/Object structures routine described with reference to step 3107 of Figure 31. Row 15 is generated by the generate Subject/Modifier structures routine described with reference to step 3108 of Figure 31. Rows 16-20 are generated by the generate_generalized_subject_object routine described with reference to Figure 35. Specifically, rows 16-19 are generated in step 3507 of Figure 35. Row 20 is generated in step 3511 of Figure 35. Rows 21-28 are generated by the generate Generalized Subject/Modifier structures routine described with reference to step 3111 of Figure 31.

At this point, the ENLP has determined the syntax, grammatical roles, and relationships between terms of the sentence. The ENLP has also generated an enhanced data representation for the sentence with all of the structures described with reference to Figure 31. The ENLP next constructs an output string (step 3003 of Figure 30) that will be used by the Data Indexer to index and store the enhanced data representation when the SQE is processing an object of the data set, or by the Query Builder to generate data queries that will be executed against the data set repository.

An output string generated by an example ENLP is of the form:

{Parameter String};{Parameter String};...;{Parameter String}|{Attribute List};{Verb List};{Word List};{ Sentence}

Each {Parameter String} is a set of six single quoted parameters, separated by semi-colons, of the form:

'subject';'verb';'preposition';'verb modifier';'object';'noun modifier'

where a wildcard character, for example "#" or "*" may be substituted for one or more of the parameters.

The {Attribute List} is a list of <Attribute Name;Attribute Value> pairs, separated by semi-colons, for example:

country;France;country;Japan

The {Verb List} is a list of all of the identified verbs in the sentence separated by semi-colons. It includes even the verbs that appear in the {Parameter String} parameters.

The {Word List} is a distinct list of all the words found in the Parameter Strings that are not also Attribute Values in the Attribute List separated by semi-colons.

The {Sentence} is the sentence that the ENLP processes after any preprocessing that may include modifying the text case and correcting spelling.

Figure 37 is an example flow diagram of the steps performed by a construct_output_string routine of the Enhanced Natural Language Parser. This routine sorts the generated parameter sets that comprise the enhanced data representation of the natural language query based on which sets are most likely to generate data queries with contextually accurate results. The routine then constructs an output string of the format described. In step 3701, the routine generates parameter strings from the structures generated as described with reference to steps 3104-3111 of Figure 31. In step 3702, the routine sorts the generated parameter strings, preferably in order of increasing ambiguity of terms, such that the parameter strings comprising terms with less ambiguity (more apt to return contextually accurate results) rank higher than those comprising terms with more ambiguity (less apt to return contextually accurate results). Although any method or combination of methods may be used to order/sort the generated parameter strings, two example sorting methods assign a weight to each generated parameter string based on a determination of term ambiguity.

Using the first method, the SQE assigns a weight to each parameter string based on a, preferably previously stored, Inverse Document Frequency ("IDF") of each parameter. The IDF of a particular parameter is equal to the inverse of the number of times that particular term appears in the data set. For example, a word that appears only one time in a data set has an IDF value of 1 (1 divided by 1), while a word that appears 100 times in a data set has an IDF value of 0.01 (1 divided by 100). In one embodiment, the weight assigned to a parameter string is equal to the sum of the IDF values of each parameter in the string. Terms that are not found in the data set are assigned an IDF of –1. In this embodiment, the parameter strings comprising the terms that appear least frequently in the data set are given a higher weight because they are most apt to return results pertinent to the natural language query.

A second method may be employed by an example SQE to weight the parameter strings according to the polysemy of each parameter. The polysemy of a term is the number of meanings that the term has. The weight assigned to each parameter string is equal to the inverse of the sum of the polysemy values for each parameter in the string. Polysemy values may be obtained for example from a dictionary service, an example of which is WordNet. One skilled in the art will recognize that any such mechanism for determining polysemy values may be used. According to convention, the minimum polysemy value is 1 (indicating that a word has only one meaning). In one embodiment of the SQE, if a polysemy value cannot be determined for a word, the SQE assigns it a polysemy value of 0.5 (indicating that the word likely has a context-specific meaning). In this embodiment, the parameter strings comprising the terms with the least ambiguity of meaning (the lower polysemy values) are given a higher weight because they are most apt to return pertinent results to the natural language query.

In step 3703, the routine adds the ordered parameter strings to the output string. When used to parse a query (not to index an object of the data set) these parameter strings are subsequently used by the Query Builder to generate data queries (*e.g.*, SQL queries). In an alternate embodiment, in order to limit the number of data queries that will be generated, a subset of the ordered parameter strings are added to the output string. In one embodiment, the first *n* parameter strings are added where *n* is a configurable number. In another embodiment, when a weight is assigned to each parameter string, a percent value of the maximum assigned weight may be used to limit the parameter strings that are included in the output string. For example, if the highest weight assigned is 10, the SQE may use 70% as a limit, including in the output string only those parameter strings that are assigned a weight of at least 7 (*i.e.*, 70% of the maximum assigned weight). One skilled in the art will recognize that any limiting technique may be used to restrict the number of parameter strings included in the output string if it is desirable to limit the number of data queries. In step 3704, the routine adds any identified attribute values to the output string, according to the described output string format. The identified attribute values may be used by the Query Builder to filter the data query results. In step 3705, the routine adds a list of identified verbs to the output string, according to the described output string format. In step 3706, the routine adds to the output string a list of the words that are present in the

parameter strings and are not in the list of attribute values. In step 3707, the routine adds the sentence to the output string. Steps 3705-3707 are performed when processing a query to add additional data to the output string that may be used by the Query Builder to generate additional data queries in the event that the data queries generated based on the standard parameter sets do not yield a sufficient number of results.

When the SQE is indexing a data set, the described ENLP output is forwarded to the Data Indexer Component of the SQE to be stored in the data set repository. Figure 38 is an example flow diagram of the steps performed by an index_data routine of the Data Indexer component of a Syntactic Query Engine. In step 3801, the routine creates one text file for each table in the data set repository. In step 3802, the routine assigns a Sentence identifier (*e.g.* Sentence ID) to the parsed sentence. In step 3803, the routine writes data, as appropriate, to each text file based on the received ENLP output. The described output parameter strings are used to populate the text files that are bulk loaded into the data repository. Steps 3802 and 3803 are repeated for each enhanced data representation received from the ENLP. The specific format of the text files is typically dictated by the bulk loading requirements of the software used to implement the data set repository.

When the SQE is processing a query (as opposed to indexing a data set), after parsing the query and forwarding the ENLP out to the Query Builder, the Query Builder generates and executes data queries against the data repository. Figures 39A and 39B are example flow diagrams of the steps performed by a build_query routine within the Query Builder component of a Syntactic Query Engine. The routine executes once for each designated parameter string (output by the ENLP) and generates and executes one or more data queries against the data set repository based on the values of the parameters that make up the designated parameter string.

Specifically, in step 3901, the build_query routine examines the designated parameters to make a preliminary determination regarding the source of the natural language query. As described with reference to Figures 4 and 10, respectively, the SQE supports both general searches and advanced searches. In the example embodiment, general searches return sentences; advanced searches return a list of verbs or other designated parts of speech or grammatical roles that can be further used to search for sentences. The steps performed by the build_query routine differ depending on the source of the

natural language query. The advanced search functionality of the example embodiment described allows natural language queries to be submitted that designate a subject and/or an object and retrieve one or more verbs. The steps performed by the build_query routine in alternate embodiments may vary depending on the syntactic and grammatical roles that can be designated within the advanced search functionality. One skilled in the art will understand how to modify the routine for a specific SQE implementation. If the designated parameter string has parameters that are only a subject and/or an object (with all of the other parameters wildcards), the routine continues in step 3904, else it continues in step 3902 to build and execute a data query, having determined that the natural language query originated from the general search functionality. In step 3902, the build query routine builds and executes a single data query based on the parameters in the designated parameter string and returns resultant sentences from the data set repository. In step 3904, if the parameters of the designated string are only a subject and/or an object, then the routine builds and executes a data query, based on the subject and/or object parameters, that returns a related verb list. The related verb list comprises distinct verbs and includes a frequency count of the number of times each verb appears in sentences within the data set. The routine temporarily stores the related verb list for later use. In step 3905, the routine determines whether or not the natural language query was submitted as an advanced search, and, if so, continues in step 3906, else continues in step 3909. In step 3906, the routine determines whether or not both a subject and an object are designated. If both are designated, the routine continues in step 3907, else it returns the related verb list (the results of the query executed in step 3904). In step 3907, the routine builds and executes a data query, based on the designated subject and object in *reverse* roles (*i.e.*, the designated subject as the object and the designated object as the subject), that returns a related verb list (as described in step 3904), and returns the related verb lists resulting from steps 3904 and 3907.

If, in step 3905 the routine determines that the natural language query was not submitted through an advanced search, then in steps 3909-3917 the routine generates and executes a series of data queries that attempt to locate objects in the data set in a heuristic manner based upon the subject and/or object of the designated parameter string and verbs that are similar to the verbs specified in the natural language query. In particular, the routine

generates and executes data queries based upon the designated subject and/or object in combination with (1) the requested verbs; (2) verbs that are entailed from the requested verbs; and (3) verbs that are related to the requested verbs, such as those produced in the related verb list resulting from step 3904. If these queries do not generate sufficient results (which is preferably modifiable), then the routine executes the same set of data queries with the subject and/or object appearing in *inverse* grammatical roles. In addition, weights are associated with the resultant objects (*e.g.*, sentences) to indicate from which data query they came, so that the overall result output can be ordered in terms of what results are most likely to address the initial query. These weights are preferably configurable, for example, by a system administrator of the SQE.

Specifically, in step 3909, the routine builds and executes a data query using the designated subject and/or object and any verb that appears in the initial (natural language) query (a requested verb). Because a search was already performed for verbs that correspond to the designated subject and/or object in step 3904, the results of the step 3904 data query can be used to streamline the data query of step 3909. In particular, a query is preferably generated and executed using the verbs that appear in both the {Verb List} of the output string generated by the ENLP and the related verb list returned by the query in step 3904 (the intersection of these two lists). These comprise the verbs that are in the initial query that have also been found to be present in the data set in objects that contain a similar subject and/or object. (Since the verbs in the {Verb List} include all of the verbs present in the natural language query, any designated parameter string that also includes a verb as one of the parameters will be accounted for also in the {Verb List}.) The routine associates a default weight with the resulting sentences indicating that these sentences came from a match of verbs present in the initial query.

In step 3910, the routine builds and executes a data query using the designated subject and/or object and verbs that are entailed from the requested verbs (verbs that appear in the initial query). As in step 3909, the results of the data query of step 3904 can be used to streamline this data query. In particular, a query is preferably generated and executed using the verbs that entail each of the verbs that appear in both the {Verb List} of the output string generated by the ENLP and the related verb list returned by the query in step 3904 (the intersection of these two lists). Entailed verbs are available through

existing applications, for example, WordNet, and are verbs that, based on meaning, are necessarily required prior to an action described by another verb. For example, given the verb, "snore," "sleep" is an entailed verb because (typically) sleeping occurs prior to snoring. The routine associates a weight referred to as an "entailed weight" with the resulting sentences indicating that these sentences came from a match of verbs that entail from verbs present in the initial query.

In step 3911, the routine builds and executes a data query using the designated subject and/or object and verbs that are related to the requested verbs (verbs that appears in the initial query). In one embodiment, the related verbs are verbs that appear in the related verb list returned by the query in step 3904 that are *not* in the {Verb List} of the output string generated by the ENLP. These are the verbs that are present in the data set in objects that contain a similar subject and/or object and that are not also requested verbs. The routine associates a weight referred to as a "verb similarity weight" with each of the resulting sentences indicating that these sentences came from a match of verb that is related to a verb present in the initial query. In one embodiment, the verb similarity weight differs with each related verb and is a configurable weighting of relative weights assigned by some external application. For example, a dictionary such as WordNet can be used to associated a "similarity measure" with all of the verbs present in a data set. These similarity measures can be further weighted by a multiplier to generate weights that indicate that the resulting sentences are less useful than those returned from data queries involving requested verbs or entailed verbs. In an alternate embodiment, the different weights for resulting sentences are determined by another application, for example, WordNet.

In step 3912, the routine determines whether or not the number of results returned by the data queries generated in steps 3909-3911 is greater than *k*, where *k* is preferably a configurable number. If the number of results is greater than *k*, the routine returns the results determined thus far, else it continues in step 3914. In steps 3914-3917, the routine generates and executes data queries that are the same as those of corresponding steps 3909-3911, with the exception that the roles of the designated subject and designated object are reversed. That is, the designated subject becomes the object and the designated object becomes the subject in the generated data query. Weights are also assigned accordingly to the resulting sentences. As

discussed with reference to Figure 14, querying the data set using the inverse subject/object relationship may return additional, contextually accurate, results that may not be returned when using the original subject/object relationship. After executing these queries, the resulting weighted sentences are returned.

In some embodiments, the results of the natural language query are sorted when they are returned. One skilled in the art will recognize that any sorting method may be used to sort the query results. In one embodiment, the results are first sorted based on the weights (default, entailed, and verb similarity weights) returned with the results as described with reference to Figures 39A and 39B. Next, the results are sorted based on attribute values designated, for example, by a user. For example, if a user specifies the name of a country in a natural language query, resulting sentences that also contain the specified country name are ranked higher than resulting sentences that do not. Finally, the resulting sentences that are returned by data queries generated from more than one parameter string are ranked higher than those from a single data query. Other arrangements and combinations are contemplated.

Although specific embodiments of, and examples for, methods and systems of the present invention are described herein for illustrative purposes, it is not intended that the invention be limited to these embodiments. Equivalent methods, structures, processes, steps, and other modifications within the spirit of the invention fall within the scope of the invention. The various embodiments described above can be combined to provide further embodiments. Aspects of the invention can be modified, if necessary, to employ methods, systems and concepts of these various patents, applications and publications to provide yet further embodiments of the invention. In addition, those skilled in the art will understand how to make changes and modifications to the methods and systems described to meet their specific requirements or conditions. For example, the methods and systems described herein can be applied to any type of search tool or indexing of a data set, and not just the SQE described. In addition, the techniques described may be applied to other types of methods and systems where large data sets must be efficiently reviewed. For example, these techniques may be applied to Internet search tools implemented on a PDA, web-enabled cellular phones, or embedded in other devices. Furthermore, the data sets may comprise data in

any language or in any combination of languages. In addition, the user interface components described may be implemented to effectively support wireless and handheld devices, for example, PDAs, and other similar devices, with limited screen real estate. These and other changes may be made to the invention in light of the above-detailed description. Accordingly, the invention is not limited by the disclosure.

APPENDIX A

Insightful™
intelligence from data

Description

A Syntactic Query Engine

June 13, 2000

Insightful Corporation
1700 Westlake Ave. N, Suite 500
Seattle, WA 98109.9891, USA
Tel: (206) 283-8802
FAX: (206)283-6310

TABLE OF CONTENTS

1 Abstract

The techniques described herein are used to create a Syntactic Query Engine that provides enhanced document indexing as well as syntactic-based searches. The approach comprises the following four steps:

INDEXING

1. Existing parsing technology is adapted to output an abstract representation for sentences that is suitable for cross-document indexing. A central idea is to collapse selected nodes in a full linguistic parse tree into a simpler dependency structure that also captures more valuable information through field redundancy
2. Smart indexing structures are created for storing, searching and manipulating sentence structures. The dependency structures output by the modified parser are stored in sets of disjoint clustered tables, each storing the same "pair" of syntactic relationships detected across a large corpus.

SEARCH

3. New search operators are developed based on the indexing structures described above. These include structured searches where keywords can be constrained to obey a precise syntactic role, such as subject, object, governing verb of a sentence, preposition or verb modifier. A methodology for sentence similarity searching is also implemented.
4. A new Graphical User Interface (GUI) concept for browsing the results of the syntactic search has been developed. This GUI introduces a new paradigm for search engines. First, it allows the user to specify the syntactic role that keywords in the input query must obey. For instance, the user may enter the keywords "Bill Clinton" with the constraint that these keywords must be the governing subjects of an action or sentence. Second, it allows three levels of progressive information discovery, corresponding to three successive screen views. At the first level, the GUI displays a list of all actions (verbs or verb phrases) found in a large corpus of documents that obey the query constraints. Using the above example, these actions could be a list of the following keywords: *win, speak, travel to, rule, lie, deny, address, etc.* This list gives the user a birdseye view of all possible actions and involvements of the entity or object about which he or she is seeking information. Each of the verbs or actions in the list is automatically linked (*e.g.*, using a hyperlink or other linking technique) to a sentence that describes the complete relationship. At the second level, the user can select any of these actions and view a complete list of sentences for each action. Each of the sentences is in turn linked with the full text of the document. At the third level, the user can select any of these sentences and view the sentence in the context of the full document.

2 Technical Background

The notion of "document as a bag of sentences" indexing rests on the principles of linguistic normalization. Linguistic normalization maps semantically equivalent sentences into one canonical sentence representation. It operates at three levels: morphological, semantic and syntactic. Morphological normalization is usually referred to as stemming or conflation. Semantic normalization involves recognition of keyword relationships such as synonyms, antonyms and meronyms. Syntactic normalization may involve transformational rules that recognize the semantic equivalence of different phrase structures.

The enhanced indexing and search technology described herein encompasses proprietary algorithms for morphological, semantic and syntactic modeling of languages. A strength of this approach is that the models are statistical in nature, and can be adapted to cross-lingual and even multimedia data dimensions. For instance, for morphological normalization, a conflation technique has been developed based on the statistics of n-grams. A weighted similarity measure is used to produce a similarity matrix that is clustered via hierarchical agglomerative clustering methods. This morphological model is language independent and robust to OCR and speech recognition errors. As such it can be applied to the output of an ASR system operating, for example, on a video audio track.

2.1 Syntactic Indexing Framework

2.1.1 Parser Technology

The enhanced indexing and search technology is being used to develop commercial Q&A solutions (*e.g.*, a Syntactic Query Engine) based on a variety of parsing technologies. These include: 1) principle based parsing and 2) stochastic parsing. Minipar (D. Lin, 1993) is an example of the first. It is a principle-based parser. Unlike rule-based grammars, which tend to produce a large number of rules to describe specific language patterns, Minipar is based on more fundamental and universal principles based on the government-binding theory (Chomsky, 1981). It achieves a relatively small number of candidate syntactic structures by applying principles to descriptions of the structures rather than to the structures themselves. Minipar carries out the parsing through an efficient message-passing algorithm. A stochastic parser is also being developed based on the Structured Language Model (SLM) conceived by C. Chelba and F. Jelinek (2000). The key features of this parser are its capacity to model and handle context dependence, its flexibility, and its computational tractability. For example, the parser has the capability of deciding part of speech and governor phrases for each term in a sentence, based on a search over the whole sentence. Hence the decisions are context-dependent as opposed to decisions made by parsers that just look at a limited history (terms preceding the particular term in the sentence) of each term, such as parsers based on hidden Markov models (HMM) (i.e. stochastic linear grammars). The parsing structure output by the parser can be represented by a binary tree. This offers a computational advantage, making the parsing algorithm tractable in a similar way to stochastic parsers based on context-free grammars. EM-type (Estimation-Maximization) re-estimation formulas (such as those used in HMM), can be derived to ease training of the parser parameters. Also,

solutions based on sub-optimal parsing strategies, such as Monte Carlo techniques, are being developed, that speed up the search for the most likely parsing of a given sentence. Functionality from multiple parsers may be incorporated into a more robust product, and the rule-based grammar approach may be extended to a number of other languages.

A parser labels every word of every sentence in every document as follows:

<term, part-of-speech, dependency-relation-with-head, head-term, document#, sentence#>

where:
term= word (stemmed or root word)
part-of-speech (pos) = noun, verb, adjective, etc.
head= term
dependency-relation-with-head= syntactic relationship between head and term.
e.g. modifier, noun-noun, subject, object.
Document # = document identifier
Sentence # = sentence number within the document #

Each record defines a node of a hierarchical tree structure of the kind shown in Figure 1.

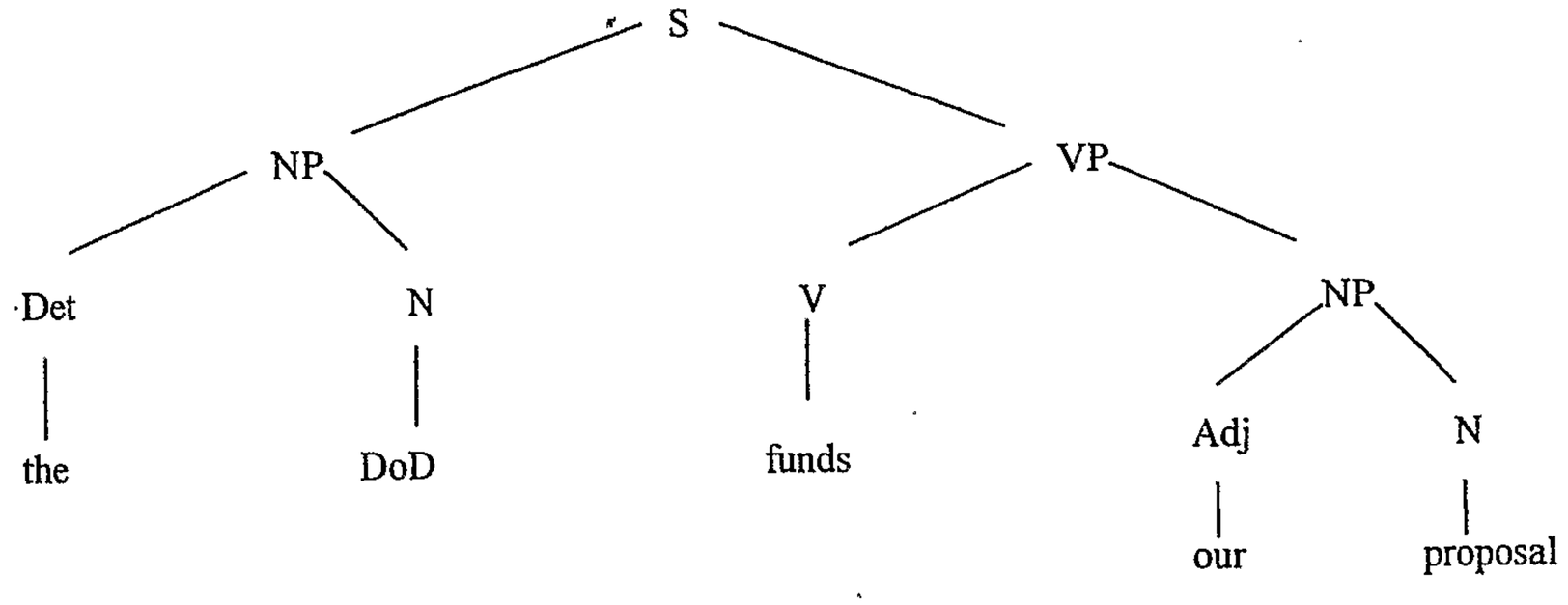


Figure 1: A phrase structure tree

The Syntactic Query Engine reconstructs augmented parse trees by joining all records from a single sentence. However, this structure does not provide efficient information access. This complex output is reduced to a "fact database" by collapsing information in the subject, verb and object categories into two relational tables. In deriving the subject-verb-object triplets the Syntactic Query Engine discards noun modifiers that have little or no semantic value.

2.1.2 Smart Syntactic Structures

As previously stated, a full linguistic parse tree does not lead to a tractable syntactic indexing scheme for cross-document analysis. The enhanced indexing and search techniques provide a set of rules for converting an augmented tree representation into a more robust and scalable data structure. The basic idea involves collapsing selected nodes

to reduce the complexity of the dependency structures. Through government-binding principles, a sentence can be described by a set of triplets of the form [head-term, term, relation], where term is a word in the sentence, head-term is the word's governor word, and relation specifies the particular syntactic relation binding the term to is head-term. Certain triplets are more informative than others in the sense that they by themselves convey most of a sentence message content. Among these triplets, triplets of the form [head is verb, term is noun/adjective, term is bound to the main subject of the verb], [head is verb, term is noun/adjective, term is bound to the object of the verb], or [head is verb+preposition, term is noun/adjective, term is bound to the preposition following the verb] are singled out. The last triplets are useful in answering queries regarding location (via the prepositions in, on, by, over, etc.), description (e.g. as), association (e.g. with, of), direction (e.g. to, from), time (e.g. after, before), purpose (e.g. for), etc. Other triplets are also relevant, such as those linking subject and objects directly: [noun A, noun B, term A and term B are connected through a verb as subject-object]. Even though these last triplets may not seem relevant within a sentence, they could potentially solve for ambiguities within a document. Also, when no verbs are present in the text (e.g. a title) triplets of the form [head is noun term, term is noun/adjective, noun term governs noun/adjective term] convey most of the relevant information.

The enhanced indexing and search techniques extract these triplets from a parser's output for a sentence, by introducing rules that act on the full parse tree for that sentence. First, the governing verb(s) in the sentence is (are) identified. For each verb (or verbal form) in the sentence , direct links representing the subject or the object associated to the verb are identified. Then, all terms that are indirectly and eventually bound either to the subject or to the object are identified. This search is not straightforward. Sentences phrased in passive-voice style, or with several clauses in them, are particularly difficult. Also, prepositions do not always immediately follow the verb; sometimes they even appear earlier than their associated verbs. These cases, as well as many others, are detected by traversing the parsing tree structure, and keeping records of the relative node position of the terms in the tree (e.g. node at the same height of the tree might be related to the same verb). Several terms may be eventually bound to both the subject and the object through the same verb. Additional rules are applied to filter intervening or contradictory triplets and non-diagnostic parts of speech. Finally, the transformed output from a single sentence is stored into a set of disjoint tables. These tables may be queried at a later time, by means of join and merge operators

2.1.3 Elementary Coreferencing Rules

The enhanced indexing and search techniques are usually able to bind pronouns (*e.g.*, "it", "that", “them”) to their "antecedent", since parsers frequently capture antecedents. The following example serves as an illustration. Consider the sentence "African bees attack humans that provoke them", and an example parsing tree associated to it (as output by Minipar):

```
E1    (()    fin C  *     )
1     (African     ~ A   2     mod    (gov bee))
2     (bees  bee N  3     s      (gov attack))
```

3 (attack ~ V_N_N E1 i (gov fin))
E3 (() bee N 3 subj (gov attack) (antecedent 2))
4 (humans human N 3 obj (gov attack))
E0 (() fin C 4 rel (gov human))
5 (that ~ THAT E0 whn (gov fin) (antecedent 4))
6 (provoke ~ V_N_N E0 i (gov fin))
E4 (() that THAT 6 subj (gov provoke) (antecedent 4))
7 (them ~ N 6 obj (gov provoke))

There are two terms whose antecedents need to be found. They are "that" and "them". The first term, "that", is the subject of "provoke". Here its antecedent is given by the parser. However, if this were not the case (as might occur with more complex sentences), the antecedent could easily be obtained by noticing that the clause associated with "that", hangs directly under the parent node, "humans". Hence, "humans" is the corresponding antecedent. The antecedent of the second term "them" is more difficult to elucidate. First it is the subject of "provoke", hence it cannot co-refer "humans" since this term is the subject of "provoke". Hence, nouns that are not bound to the term "humans" are determined among those found in the immediate precedent clause. In this case the only such noun is "bees", so this is the corresponding antecedent. Co-referencing in a document is similarly done. The ordered sequence of paragraphs is seen as a linear tree, where each node represents a paragraph, and each paragraph is headed by the preceding one. The first paragraph is headed by a symbolic root node (the document). Each paragraph node is similarly decomposed as a linear tree formed by the nested parsing trees of all sentences within the paragraph. The main idea is to see sentences within a single paragraph as consecutive "clauses" which are nested one after the other. These nested parsing trees form an unbalanced paragraph parsing tree that is heavier to its right. Co-reference is solved for as it would be solved for in a single parse tree.

Table 1 is an example set of data structures employed in an example implementation of sentence decomposition for a prototype collection of documents. The table entries correspond to the first four sentences of the same abstract. All four tables are incremented with parsed information from additional abstracts. The prototype collection involved 50,000 abstracts and the implementation supported the scalability of the approach to large collections.

INSIGHTFUL CONFIDENTIAL

Table 1

Subjects Table

Subject	Verb	DocID	SentenceID
cervical	be	20450337	1
Spondylotic	be	20450337	1
Myelopathy	be	20450337	1
aging	result	20450337	2
Process	result	20450337	2
Degenerative	cause	20450337	2
Change	cause	20450337	2
Cervical	cause	20450337	2
Spine	cause	20450337	2
Symptom	develop	20450337	3
differential	include	20450337	4
Diagnosis	include	20450337	4

Objects Table

Verb	Object	DocID	SentenceID
be	cause	20450337	1
be	the most	20450337	1
be	common	20450337	1
be	spinal cord	20450337	1
be	dysfunction	20450337	1
be	older	20450337	1
be	person	20450337	1
cause	compression	20450337	2
cause	spinal cord	20450337	2
develop	insidiously	20450337	3
characterize	symptom	20450337	3
include	condition	20450337	4
include	myelopathy	20450337	4
include	multiple sclerosis	20450337	4
include	amyotrophic	20450337	4
include	lateral	20450337	4
include	sclerosis	20450337	4
include	masses	20450337	4
include	metastatic tumor	20450337	4
include	spinal cord	20450337	4
press	metastatic tumor	20450337	4

VerbModifier Table

Verb	Preposition	VerbModifier	DocID	SentenceID
result	in	degenerative	20450337	2
result	in	change	20450337	2
result	in	spine	20450337	2
result	in	advanced	20450337	2
result	in	stage	20450337	2
result	in	compression	20450337	2
result	in	spinal cord	20450337	2
cause	in	advanced	20450337	2
cause	in	stage	20450337	2
characterize	by	neck	20450337	3
characterize	by	stiffness	20450337	3
characterize	by	arm	20450337	3
characterize	by	pain	20450337	3
characterize	by	numbness	20450337	3
characterize	by	hand	20450337	3
characterize	by	weakness	20450337	3
characterize	by	leg	20450337	3
result	in	myelopathy	20450337	4
result	such as	multiple sclerosis	20450337	4
press	on	spinal cord	20450337	4

Sentence Table

Sentence	DocID	SentenceID
Cervical spondylotic myelopathy is the most common cause of spinal cord dysfunction in older persons	20450337	1
The aging process results in degenerative changes in the cervical spine that, in advanced stages, can cause compression of the spinal cord	20450337	2
Symptoms often develop insidiously and are characterized by neck stiffness, arm pain, numbness in the hands, and weakness of the hands and legs	20450337	3
The differential diagnosis includes any condition that can result in myelopathy, such as multiple sclerosis, amyotrophic lateral sclerosis and masses (such as metastatic tumors) that press on the spinal cord	20450337	4

2.1.4 Storage and Access

Using the enhanced search and indexing techniques, the Syntactic Query Engine stores the modified parser output as sets of disjoint tables of similar syntactic primitives. For the sake of simplicity, an example is presented with a set of relational database tables that store only three types of coarse syntactic relationships: 1) subject-verb; 2) verb-object; and 3) verb-preposition-verbModifier. (Any number of other syntactic relationships could be stored.) Correspondingly, the output includes a: 1) *subjects table* configured as *{Subject, Verb, SentenceID, DocID}*; 2) *objects table* configured as *{Verb, Object, SentenceID, DocID}*; 3) *verb modifiers table* configured as *{Verb, Preposition, VerbModifier, SentenceID, DocID}*; 4) *sentence table*, configured as *{SentenceID, DocID}*. Indices are built for each table. For instance, (1) has two indices: the first based on *Subject*, the second on *DocID* and *SentenceID*; (3) has three indices: the first based on *Verb*, the second on *VerbModifier*, and the third on *DocID* and *SentenceID*.

The output from each sentence is used to populate all three tables, each table storing the same type of syntactic relationship across a corpus. Table 1 shows an example of a possible sentence decomposition for a single Medline abstract. Note that redundancy is introduced by collapsing terms and prepositional phrases that are bound to the object and subject category into a single category. This means that more than one object or subject entity may be associated with the same governing verb of a sentence. However, the ambiguity can be resolved with a join or merge operation of a *subjects table* with an *objects table*. For instance, a union of these two tables based on the verb "cause" and the object "compression" AND/OR "spinal cord" would produce the following answer for subject: "degenerative change cervical spine". The prototype implementation of the Syntactic Query Engine applied to a subset of 50,000 Medline abstracts indicates that this redundancy does not appear to compromise performance substantially (< 0.1 sec for 5 nested joins across the full set of tables for the entire database), but increases the robustness of the data bank. These storage structures also address the difficult problem of co-referencing or tracking entities in a single document or phrase. In this implementation, data is stored based on the key of the most frequently used index. This enables fast data retrieval from disk as all data is stored in the neighboring disk blocks. In order to solve performance problems related to incremental indexing and improve scalability even more, modified database schemas and key sorting methods based on clustered tables, hash clusters, binary or R-trees may be used.

2.2 Operations on Data Structures

2.2.1 Syntax Operators, Similarity Metrics and Robust Co-referencing

The modified parser output or triplet representation relies on the identification of the governing verb of a sentence, and is verb centric. Verbs or actions govern subjects and objects. The smart structures introduced above support global search operators that can be used to profile the syntactic role of named entities across a document collection. Returning to the 50,000 Medline abstracts, the database could be queried for all subject or object roles of an entity, say "CB1" (Figure 2). The columns show the governing verbs or

actions that are bound to "CB1" according to that precise syntactic relationship across the entire corpus, arranged in order of decreasing statistical significance.

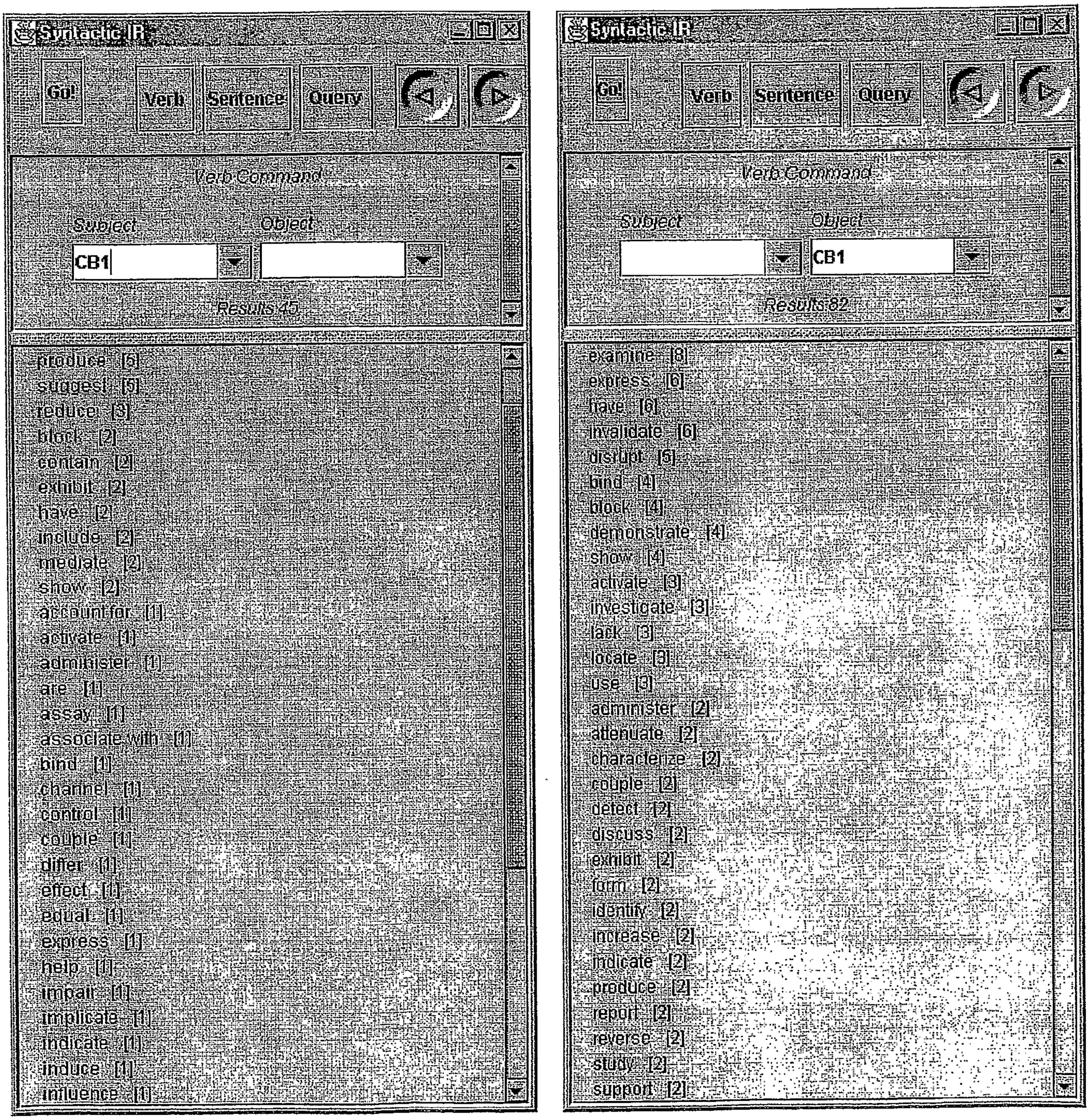


Figure 2

Figure 2 shows example syntactic profiling for keyword "CB1". Shown are possible subject or object roles that "CB1" fills in a corpus of 50,000 Medline abstracts used in the example implementation. These "action" roles are arranged in decreasing order of statistical significance.

The linguistic principle of "headness" implies that any phrase has a single head. This head is usually a noun in NPs and the main verb in the case of VPs. Although the head-modifier relation implies semantic dependence, the parser output defines a purely syntactic relationship at the single sentence level. However, the head may serve as an index for a list of phrases with occurrence frequencies across a collection. Syntactic profiling operators (like subject and object roles, and others involving prepositions and verb modifiers) are used to semantically cluster keywords based on their syntactic relationships. For instance, entities in a corpus can be associated with a "bag" of subject actions and a "bag" of object actions, as well as modifiers involving those actions. Each action in each one of these bags has a weight (the relative frequency of the noun-action pair within a document collection). Two keywords or entities are then said to be similar if their corresponding subject and object bags of actions are similar. This similarity can be quantitatively measured by the Euclidean distance between the relative frequency distributions associated with their bags of actions. Two nouns are very likely to be semantic or at least functional synonyms if their syntactic similarity is large. Hence, the enhanced search and indexing techniques solve the co-reference problem by studying the observed bag of actions statistics associated with the keywords in a corpus.

The following example illustrates this point. Consider the sentences "Several rats developed brain tumors. They spread to other organs." The possible antecedents of "They" are "brain tumors" and "rats".

action	BrainTumor	Rat	Organ	action	BrainTumor	Rat	Organ
affect	0.00	0.00	0.05	induce	0.04	0.05	0.00
appear	0.04	0.00	0.00	investigate	0.00	0.00	0.04
cause	0.00	0.00	0.04	involve	0.00	0.00	0.06
cause	0.00	0.02	0.00	model	0.00	0.08	0.00
characterize	0.00	0.00	0.04	observe	0.00	0.00	0.08
compare	0.04	0.03	0.00	occur	0.04	0.00	0.00
contain	0.04	0.06	0.00	originate	0.04	0.00	0.00
damage	0.04	0.00	0.00	overlook	0.04	0.00	0.00
decide	0.04	0.00	0.00	pose	0.04	0.00	0.00
decrease	0.00	0.03	0.05	prevent	0.00	0.00	0.04
define	0.04	0.00	0.00	produce	0.00	0.03	0.00
demonstrate	0.00	0.00	0.05	prolong	0.04	0.00	0.00
detect	0.00	0.00	0.05	provide	0.00	0.00	0.05
develop	0.04	0.03	0.00	receive	0.00	0.08	0.00
examine	0.00	0.00	0.05	release	0.04	0.00	0.00
exhibit	0.00	0.04	0.00	represent	0.00	0.00	0.04
express	0.00	0.06	0.04	result	0.00	0.05	0.00
factor	0.00	0.00	0.04	reveal	0.00	0.02	0.00
feed	0.00	0.05	0.00	show	0.12	0.14	0.06
form	0.00	0.00	0.04	suggest	0.00	0.03	0.00
give	0.00	0.03	0.00	target	0.00	0.00	0.06
implant	0.04	0.00	0.00	undergo	0.00	0.03	0.00
include	0.12	0.00	0.11	understand	0.04	0.00	0.00
increase	0.08	0.06	0.05	use	0.08	0.08	0.00

Table 2

In Table 2, relative frequencies for the top 20 actions associated to each of the terms "brain Tumor", "rat", and "organ" are shown. The first column corresponds to the union of all three terms' top 20 actions. Note: columns one and three do not add up to one due to numerical round off. Table 2 shows the union of the subject action bags (with the 20 most frequent actions) for "brain tumor" and "rat", as well as the object action bag (with the 20 most frequent actions) for "organ". The relative frequencies were computed for the top 20 actions for each of these terms in the 50,000 Medline abstracts. A zero entry in the table indicates that the action is not among the top 20 actions associated with the term heading the column. The corresponding similarities (Euclidean distances) between "brain tumor" and "organ", and "rat" and "organ" are 0.072 and 0.093, respectively. Hence "brain tumors" is the most likely antecedent of "They". Note that the relative weights 1 - (0.072/(0.072+0.093)= 0.56, and 1 - (0.093/(0.072+0.093)=0.44,
could be interpreted as corresponding probabilities of "brain tumor" and "rat" being the antecedent of "They" within the second sentence's context.

When a co-reference is not exact, a measure of uncertainty about the possible antecedents helps in any further processing of the sentences. Hence a weight or probability can be associated to triplets formed by "guessing" or "imputing" the antecedent of a co-reference. All triplets are placed in this framework by assigning a weight of 1.0 to all well-defined triplets. For the example, the associated weighted bag-of-sentences would be ([develop, several, verb-subject, 1], [develop, rat, verb-subject, 1], [develop, tumor, verb-object, 1], [develop, brain, verb-object, 1], [spread to, organ, verb-object, 1], [spread to, other, verb-object, 1], [spread to, tumor, verb-subject, 0.56], [spread to, rat, verb-subject, 0.44]). The process can be further refined by allowing a second pass over the text in order to update the weights (for example by replacing the weights with the average weights over all sentences co-referring the same object). The premise here is that a measure of uncertainty about possible antecedents is far better than a wrong antecedent.

Using the enhanced search and indexing techniques, the Syntactic Query Engine implements a scheme for phrase weighting and document similarity, which is similar to statistical weighting schemes employed for term-document matrices in information retrieval. The frequency of occurrence of the governing verb of a sentence or subject of a sentence across a document collection is normalized, in a fashion similar to IDF (inverse document frequency) weighting. With regard to within sentence and within documents normalization schemes, phrase frames may contain nested phrase frames at different depths. The main head carries the most semantic information, while head modifiers increase the amount of semantic information carried by the frame. However, the amount of information added to the head by a modifier is inversely proportional to its depth. This data is used to measure sentence statistics within and across documents, and to define document similarity based on sentence structure and content.

3 Implementation

3.1 Syntactic Queries on Medline Abstracts

In an example implementation, a set of 50,000 Medline abstracts is indexed and searched. Admissible queries on the Medline abstracts database are of the same form as sentences in the abstracts. Syntactic queries can target useful subject-verb-object relationships in the database of parsed sentences that results from all indexing steps. Wild-cards are allowed to be substituted for certain parts of speech in the queries. The wild-card in the query must be a valid term in the lexicon preceded by a "#" symbol. Thus the wild-card represents any term in a sentence that shares the same part-of-speech, head-term and dependency relation with the term preceded by a #. Queries are parsed and represented in a similar fashion as sentences, except for the last two keys, namely document # and sentence #, which are dropped in the representation of queries.

Example implementations based on many different embodiments of the technology have been developed:

- Client/server architecture with Java GUI client and CORBA as the communication protocol
- Web server architecture with HTML GUI layout compatible with Netscape and Internet Explorer Web browser
- Dedicate ASP control interface (illustrated in Figure 3)

Other implementations are also contemplated. For example, embodiments of this technology are not restricted to a workstation or desktop implementation, and may be independent of a particular operating system. Also, for example, the interface concept is highly compatible, and ideal for deployment on portable devices, including cellular phones.

An example GUI interface provides five levels of control (Figure 4): two levels for querying and three levels for browsing information returned in response to a query. The levels of control for querying comprise: 1) query type selection; 2) query entry. The levels of control for browsing the results of a query comprise displays of: 1) word dependency maps ranked in decreasing order of statistical relevance; 2) summary list of complete sentences about a target entity, linked to the word dependency maps; 3) full text of the abstract, linked to each sentence in the list above.

The following example (*prepared by a non-biologist)* illustrates the query process:

QUERY

1. Choose from one of the following in a display menu: (a) find functional dependency relationships; (b) find object-subject relationships; (c) search syntactic collocations by example

2. The user chooses (a), and enters the name of an enzyme or gene, whose functions he/she wishes to have summarized. The query could be for instance: "Find all verb dependencies of PLCBeta1"

BROWSING THE RESULTS OF A QUERY

1. The interface presents two statistically ranked lists of actions profiling:

PLCBeta1 as an object:
1) activate → PLCBeta1 (157 occurrences)
2) inhibit → PLCBeta1 (52 occurrences)
3) block→ PLCBeta1 (7 occurrences)
...

PLCBeta1 as a subject:

1) PLCBeta1 → hybridize (89 occurrences)
2) PLCBeta1 → interrupts (12 occurrences)
3) PLCBeta1 → mediates (11 occurrences)
4) PLCBeta1 → regulates (10 occurrences)
...

2. Each entry in the ranked list above is linked to parent phrases in the corpus of 333,000 Medline abstracts. For instance, when the user selects the keyword "activate", he/she is presented with the list of 157 sentences that summarize what activates PLCBeta1:

   1) "Low concentration of tubulin activated PLCBeta1, whereas higher concentrations inhibited the enzyme."
   2) "Tubulin, Gq and phosphatidylinositol 4,5-bisphosphate interact to activate PLCBeta1"
   3) "A unique ability of tubulin to regulate PLCBeta1 was observed"
   .....

3. Each of the sentences in the list above is linked to the original Medline abstract. The user can select a sentence of particular interest and read the full abstract.

Figures 5 through 7 illustrate an example Java embodiment of the GUI. Figures 8 through 10 illustrate an example HTML/Web browser implementation of the GUI. Finally, Figure 11 illustrates an example sentence similarity operator.

INSIGHTFUL CONFIDENTIAL

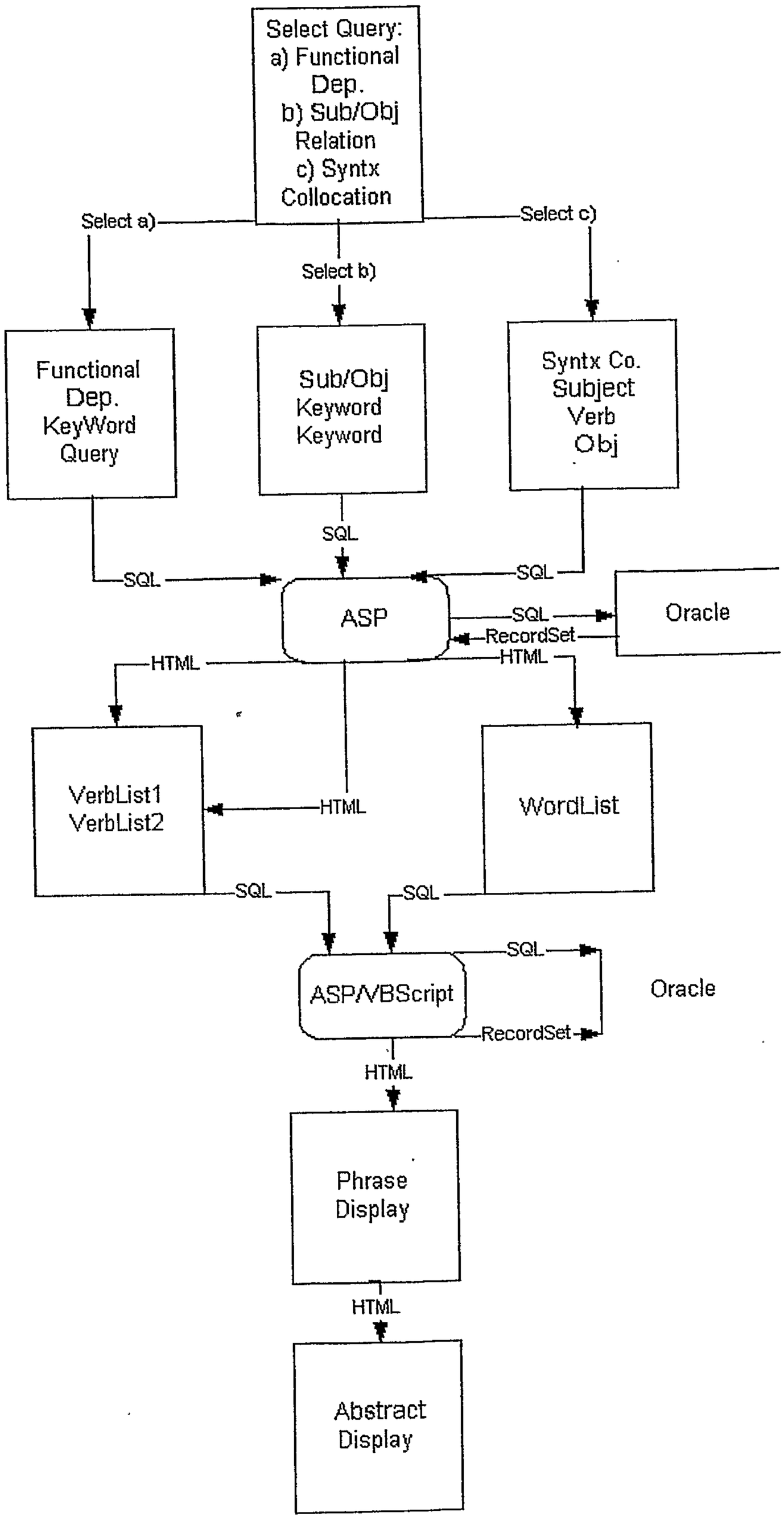
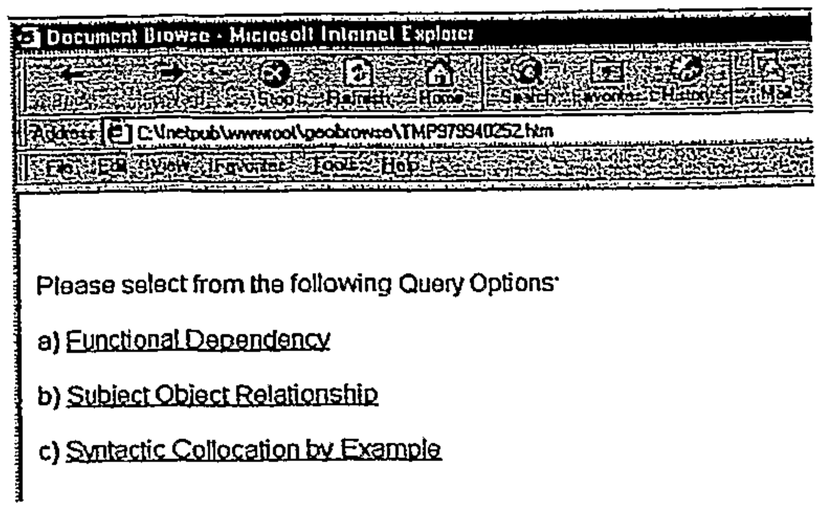


Figure 3: prototype GUI process flowchart

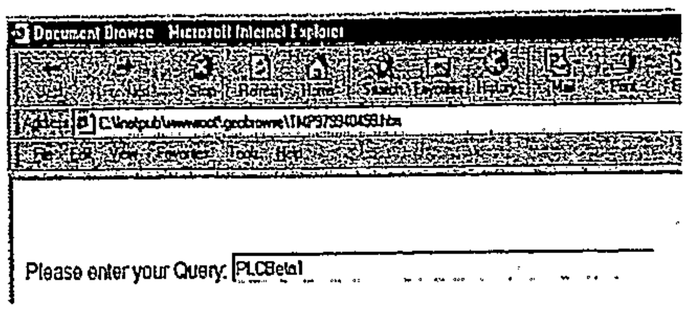
60

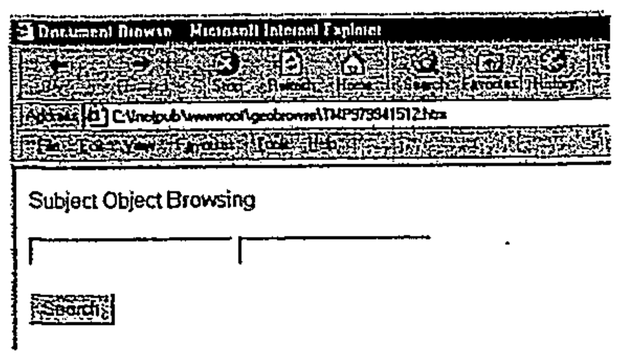
INSIGHTFUL CONFIDENTIAL

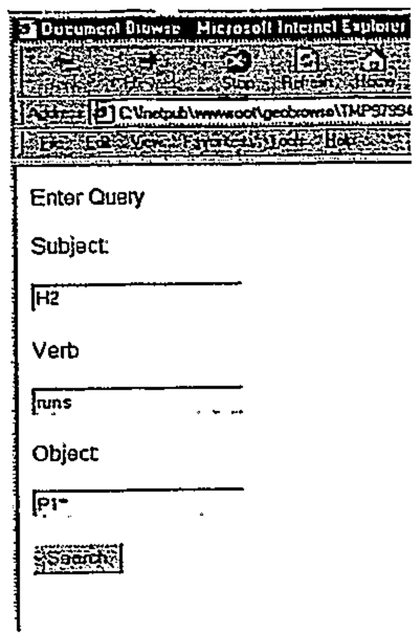
LEVEL 1



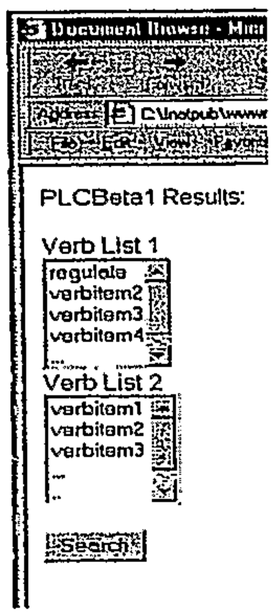
LEVEL 2







LEVEL 3



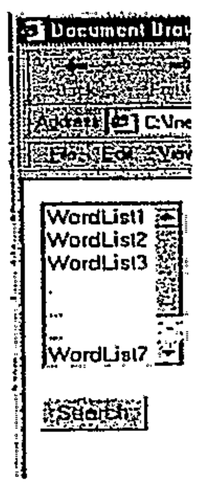
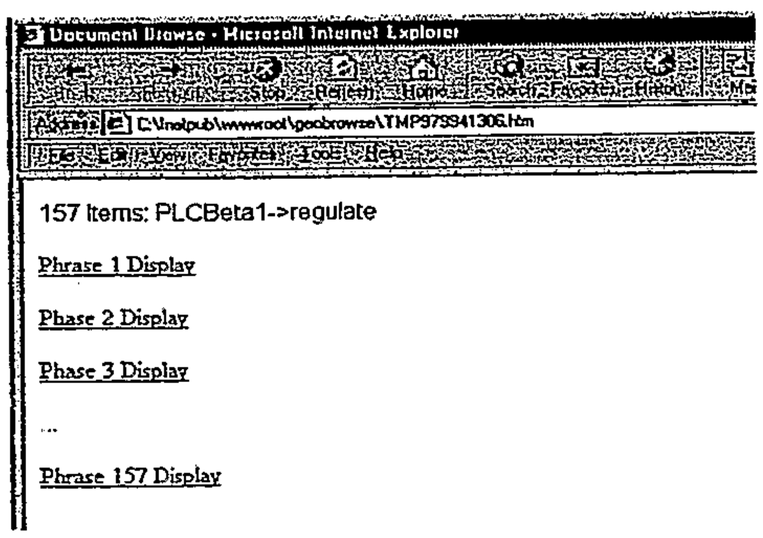
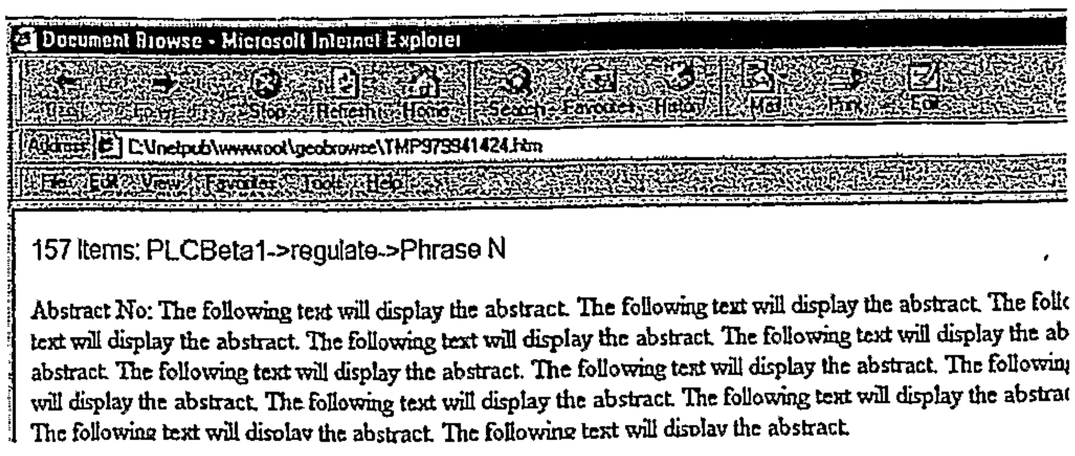


Figure 4: the five levels of GUI control

LEVEL 4



LEVEL 5



INSIGHTFUL CONFIDENTIAL

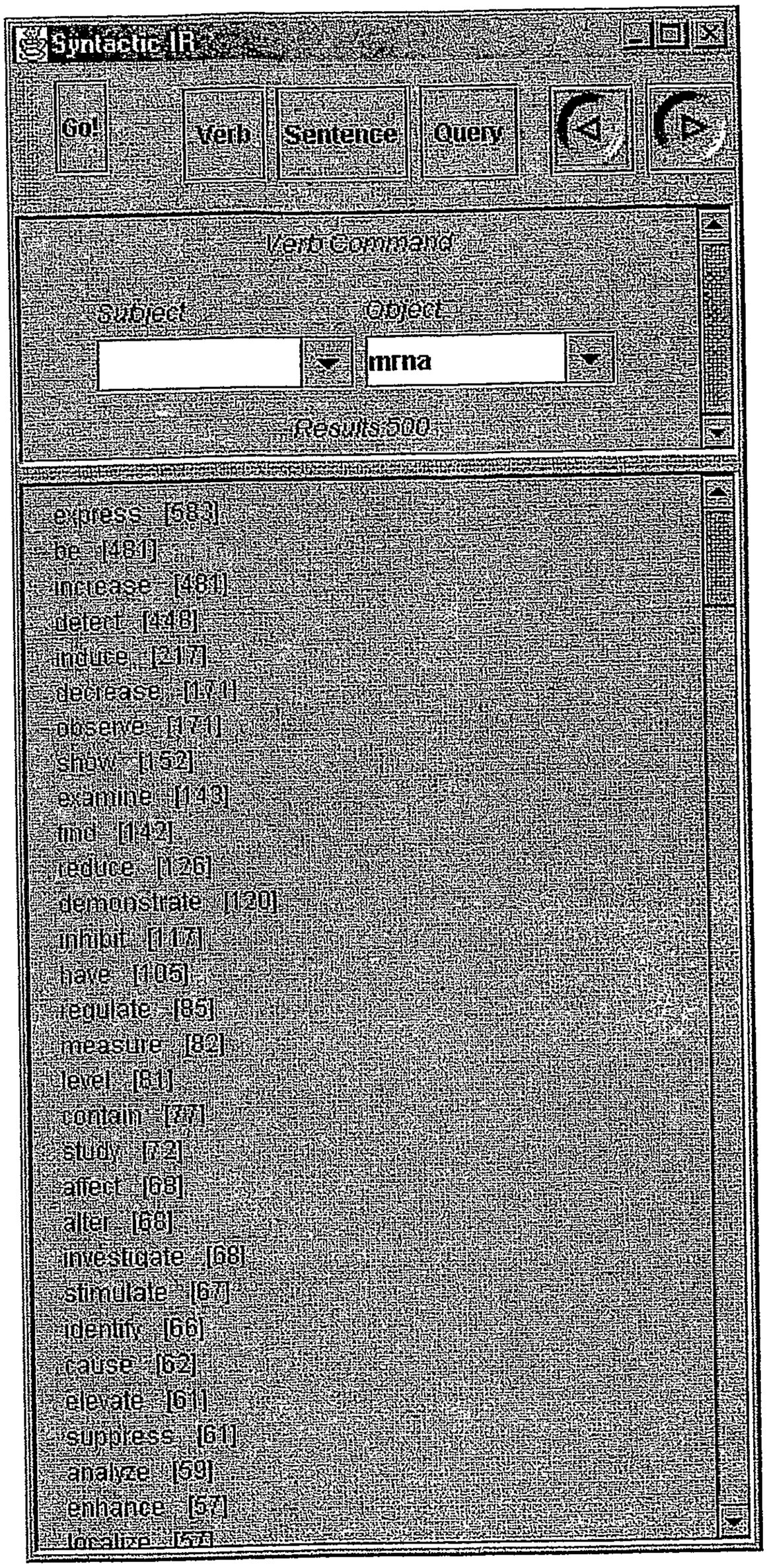


Figure 5

Figure 5 shows an example search tool based on syntactic parser output. A list of verbs or "actions" is shown for which mRNA (messenger RNA) is the object. The verbs are arranged in decreasing order of their statistical significance in a database of 50,000 Medline abstracts.

INSIGHTFUL CONFIDENTIAL

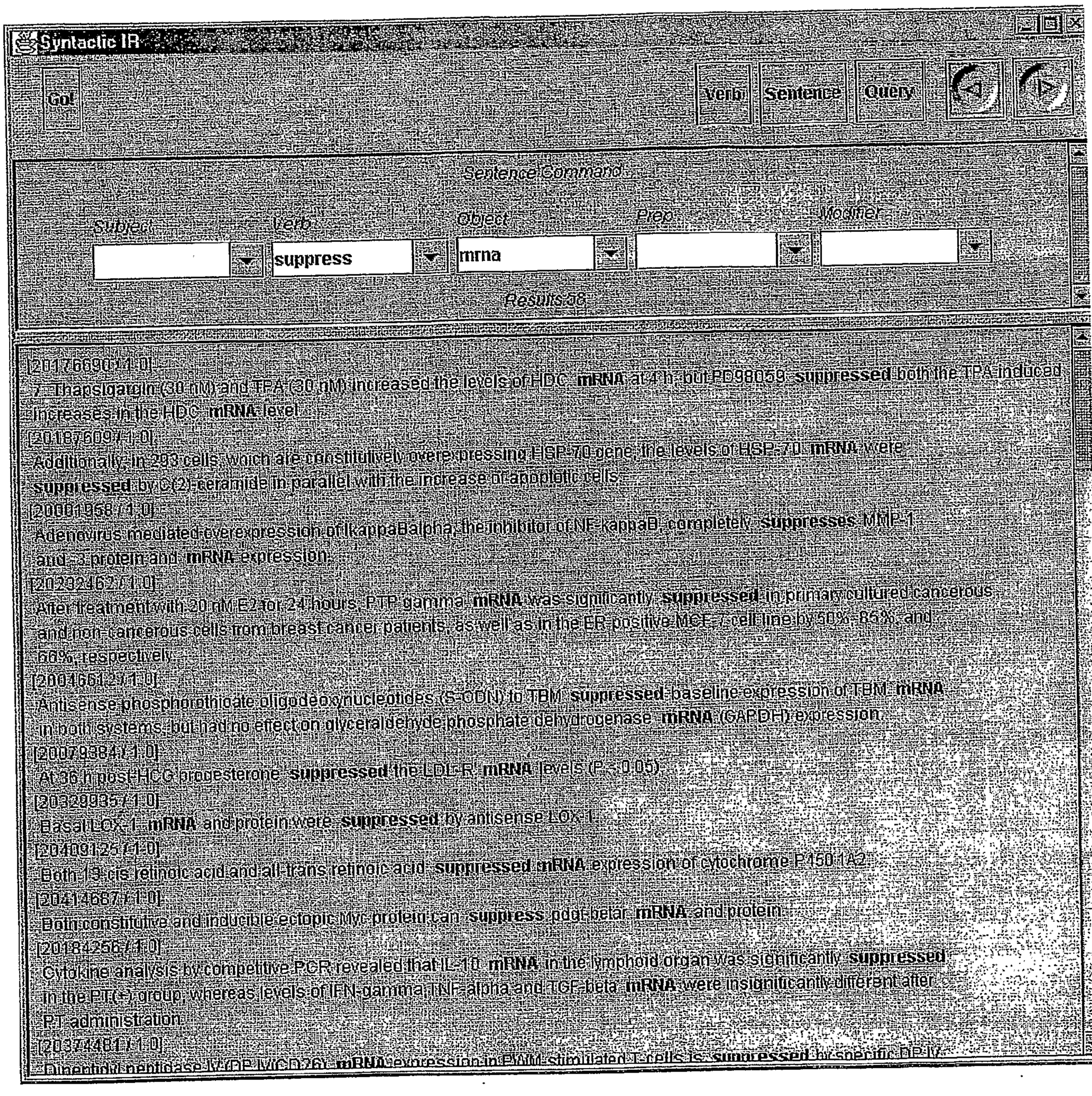


Figure 6

Figure 6 is a continuation of the example of Figure 4. Selecting the verb "suppress" retrieves all sentences that show what suppresses (=verb) mRNA (=object). Note that the search tool can resolve active, as well as passive verb forms.

INSIGHTFUL CONFIDENTIAL

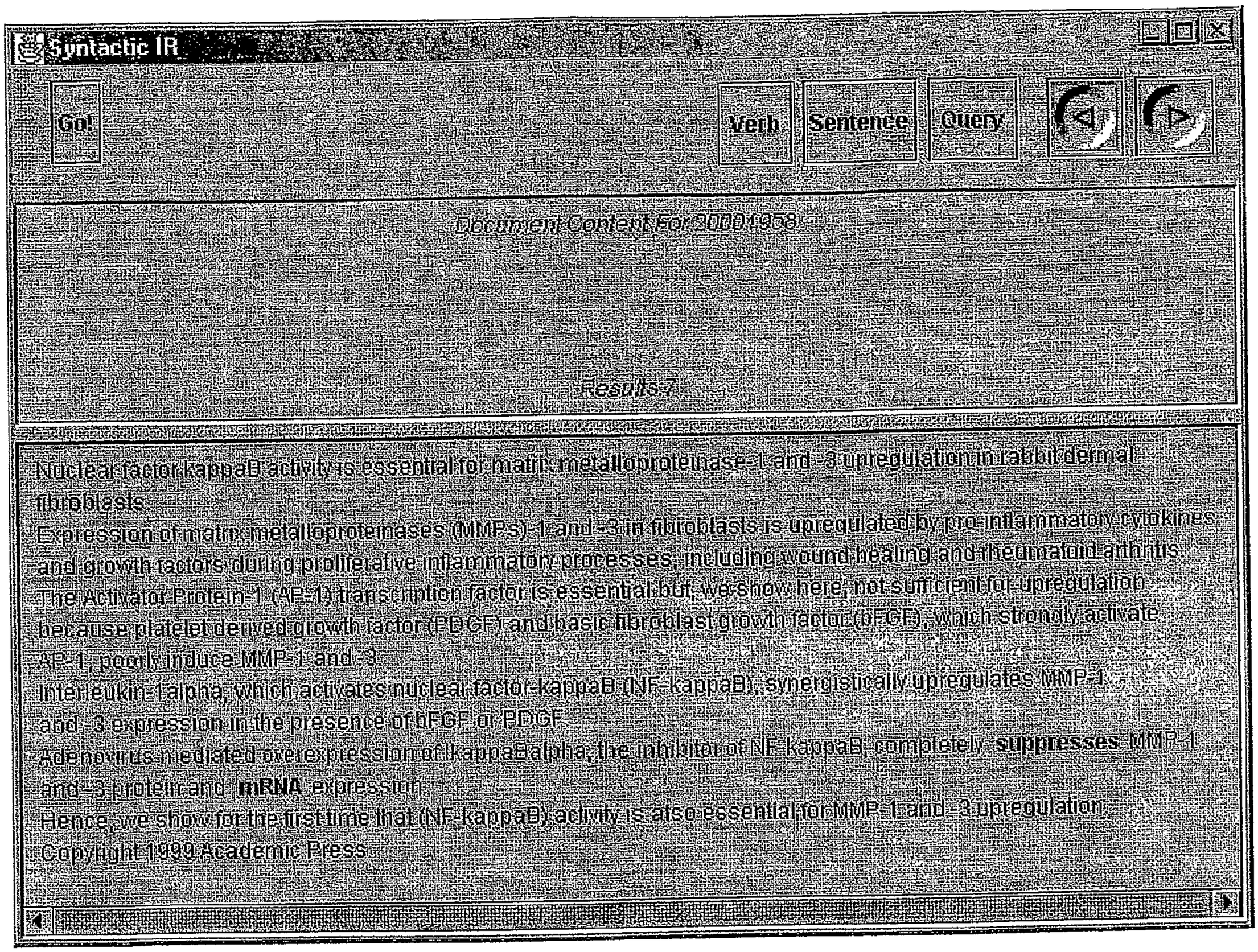


Figure 7

Figure 7 is a continuation of the example of Figure 5. Selecting a sentence returns the full text abstract.

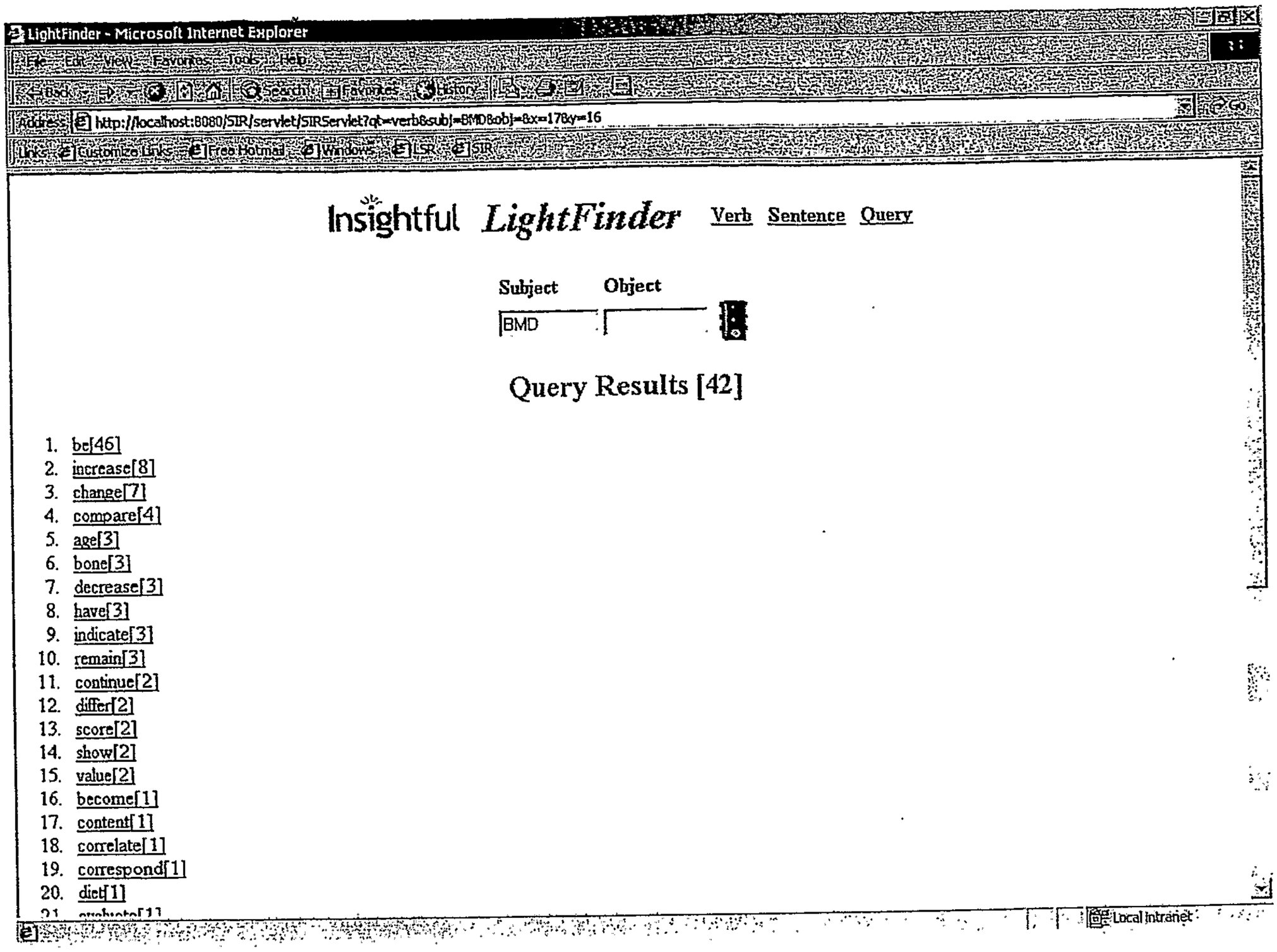


Figure 8

Figure 8 is an example web server embodiment. The web server search tool displays a list of verbs or "actions" for which BMD (Bone Mineral Density) is the subject. The verbs are arranged in decreasing order of their statistical significance in a database of 50,000 Medline abstracts.

INSIGHTFUL CONFIDENTIAL

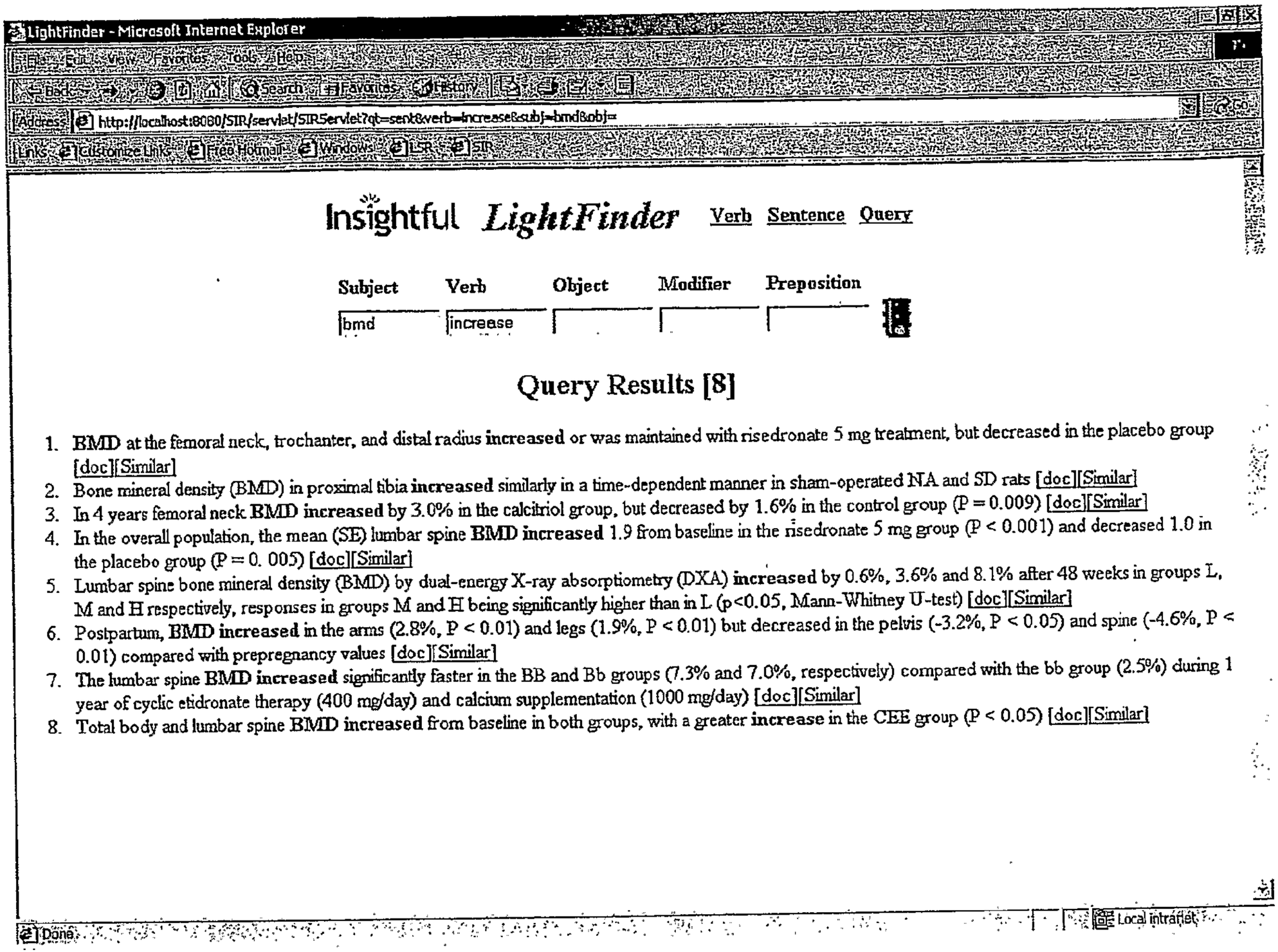


Figure 9

Figure 9 is a continuation of the example of Figure 7. Selecting the verb "increase" retrieves all sentences that show all circumstances where BMD (=subject) increases (=verb) mRNA.

INSIGHTFUL CONFIDENTIAL

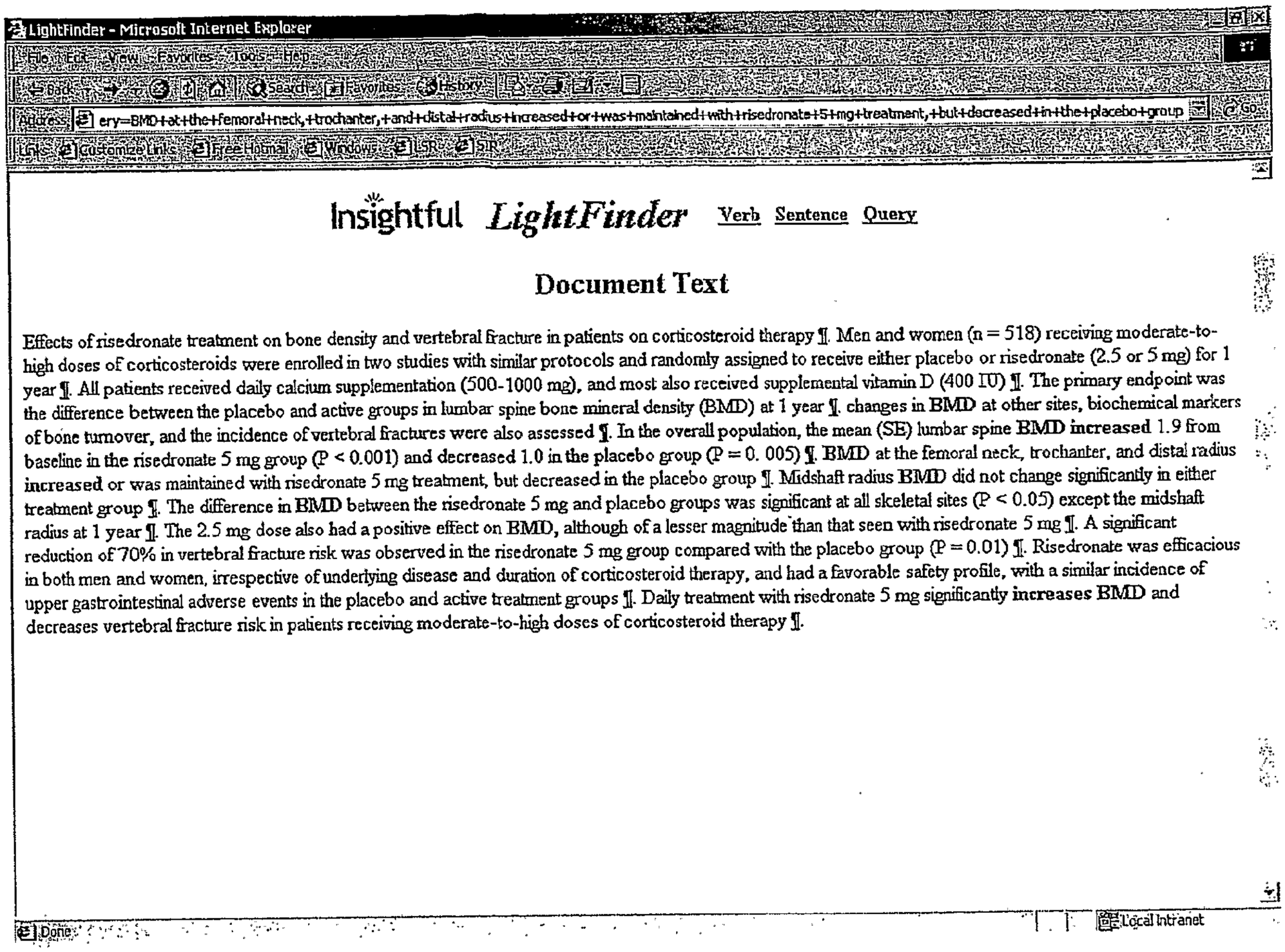


Figure 10

Figure 10 is a continuation of the example of Figure 8. Selecting any sentence returns the full text abstract showing the sentence in context.

INSIGHTFUL CONFIDENTIAL

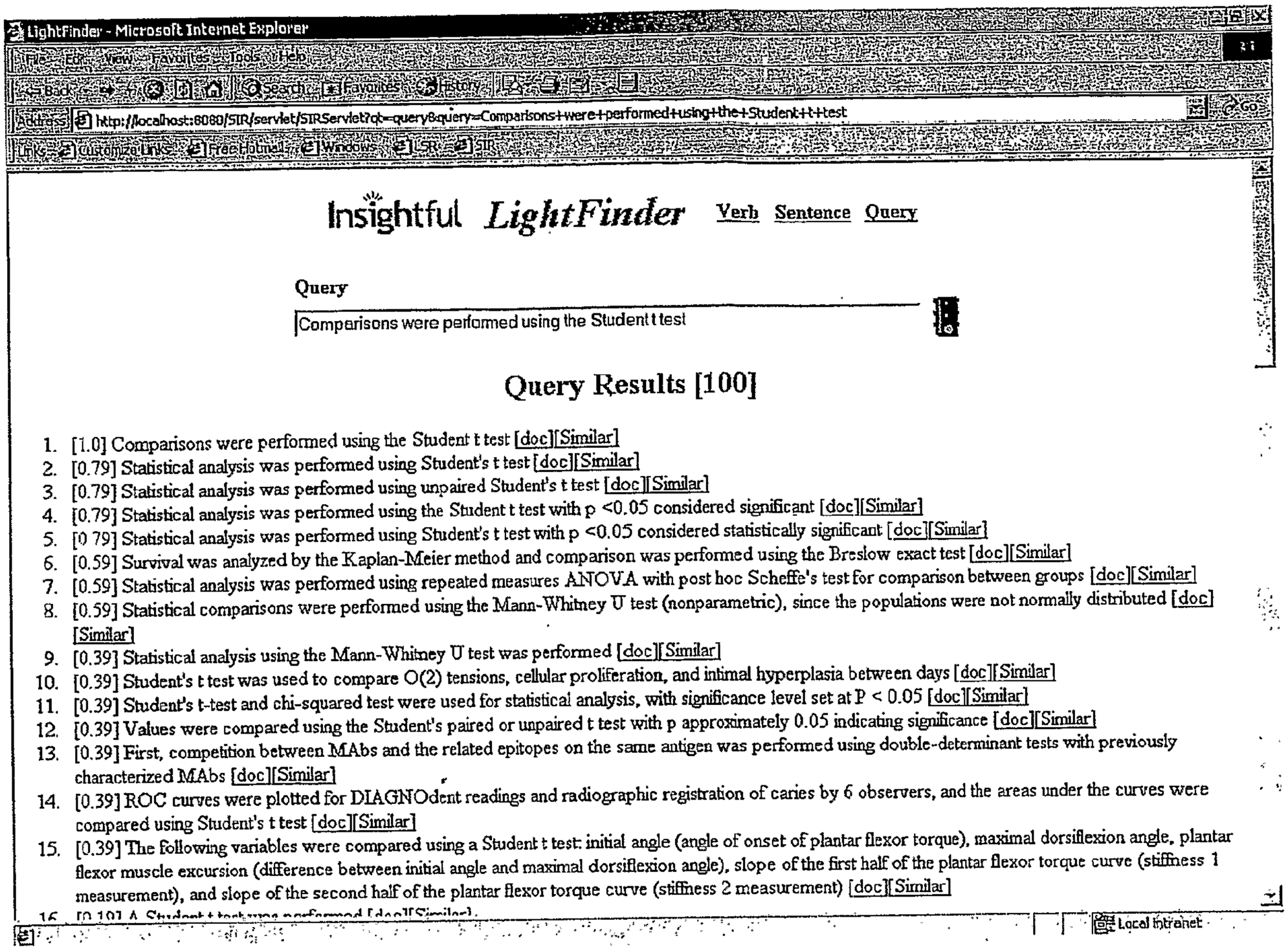


Figure 11

Figure 11 is an example illustration of the sentence similarity operator. The search tool receives the sentence "Comparisons were performed using the Student t test", and determines a list of similar sentences in the corpus.

APPENDIX B

)C>
)CNO>

)OCNO>
L>
litical background
IL>
EXT>
ited Arab Emirates.
untry Profile Uae 1997 / 1998.
litical background.
ernational relations & defence.
e UAE is a small country surrounded by powerful neighbours in an unstable part of the world

international relations over the past 20 years have reflected this and it has followed a course delicate diplomacy.
hough it has been part of the six-member pro-Western Gulf Co- operation Council (GCC) ce the latter was founded in 1981, the UAE has tended to take a more accommodating nce than its co-members, especially Saudi Arabia and Kuwait, towards perceived external eats such as Iran.
is reflects the fact that both Dubai and Sharjah have substantial Iranian populations, and that n is an important economic partner for the UAE.
lations with Iran.
th the outbreak of the Iran-Iraq war in 1980, the UAE found itself walking a diplomatic htrope.
vas one of the first Arab countries to receive Iranian officials after the 1979 Islamic revolution d continued to welcome them even after the war with Iraq had started.
wever, it also contributed to Iraq's war effort when it became clear that Iraq might lose the uggle if it did not receive backing.
er the Gulf war of 1991 the UAE made great efforts to forge closer ties with Iran but in 1992 n took control of the whole of the Gulf island of Abu Musa which it had shared with Sharjah ce 1971.
e action made the UAE nervous and since then this unease has translated into overt position to perceived Iranian expansionism.
nce 1995 alarm has grown as Iran has strengthened its grip on the islands of Abu Musa, eater Tunb and Lesser Tunb.
e UAE has frequently called for the dispute to be resolved either through bilateral gotiations or by the International Court of Justice in The Hague.
position has been supported by the GCC and the Arab League, both of which have argued at a normalisation in ties with Iran is dependent on the affair being resolved.
elations with Iran have caused internal friction within the federation.
ven its strong trade links and large indigenous Iranian population, Dubai has maintained a nciliatory stance towards Tehran, arguing that close contacts will ultimately lead to a solution the dispute.
contrast, Sharjah and Ras al-Khaimah, whose territorial waters adjoin the three islands, have lled for greater steps to be taken in forcing the Iranian regime either to hand back the islands , at the very least, to allow a joint administration to be set up.
elations with Iraq.
espite supporting Iraq in the Iran-Iraq war, the UAE has also had much to fear from the nbitions of the Iraqi leader, Saddam Hussein.
1988 Baghdad accused the UAE, along with Kuwait, of breaking OPEC oil quotas, driving ices down and depriving Iraq of revenue to rebuild its economy after the war with Iran.
ne fact that this was one reason given by Iraq for its invasion of Kuwait in 1990 was not lost n the UAE.
evertheless, Sheikh Zayed has taken a lead in calling on the UN to relax sanctions against aq in order to ease the suffering of its people.
ore important to Sheikh Zayed, however, is the desire to build up Iraq to balance Iranian ower.
ubai, too, is interested in having sanctions on Iraq lifted to take advantage of the trade that ould then ensue.
il wealth.
ver the past few years the country's oil wealth has been a mixed blessing.
he UAE was criticised for allegedly not bearing enough of the burden in the Gulf war effort gainst Iraq, particularly as economically, it emerged as one of the few regional beneficiaries.
lthough it mobilised its troops and paid around $10bn towards the cost of the war, it gained ore from the crisis than it lost.
benefited, for instance, from higher oil revenue as prices rose and it increased production to ake up for lost Iraqi and Kuwaiti output.
ubai gained from the relocation of business from Kuwait.
uring the 1990s the UAE has taken a back seat in OPEC, generally supporting the line dopted by Saudi Arabia.
ubai's refusal to make pro- rata cuts in line with the federation's oil production quota has laced the burden of compliance solely on Abu Dhabi.
his has caused some difficulties for Abu Dhabi, given that its foreign oil partners have loughed significant sums into raising the emirate's sustainable oil capacity, much of which has emained idle.

Indeed, Abu Dhabi is reckoned to have over 500,000 barrels/day (b/d) of spare capacity in place.
However, a 25% drop in Dubai's oil production over the period 1993-97 has provided some relief for Abu Dhabi allowing it to raise output over the period by about 150,000 b/d to an estimated 1.9m b/d.
New OPEC quotas, which came into force on January 1st 1998, have also proved a blessing for the oil-rich emirate, as the 205,000 b/d increase in the UAE quota (taking it to 2.366m b/d) has been covered solely by Abu Dhabi.
Total Abu Dhabi production now stands at over 2.1m b/d.
Defence pacts.
Since the end of the Gulf war, the UAE and the rest of the GCC have been reluctant to rely for their security on the so-called Damascus Declaration group of countries: the GCC, Egypt and Syria.
The latter two sent troops to the Gulf to help to oust Iraq from Kuwait, after which they agreed to station troops permanently in the region.
However, GCC states, even though they suffer from a lack of manpower, have been reluctant to put the arrangement into practice.
Even internally, the GCC has been unable to agree on a joint force.
Oman, for instance, has persistently advocated the expansion of the current joint force from a few thousand to 100,000 troops, but the UAE has opposed this because it fears domination by Saudi Arabia.
Instead, the UAE has looked to the West, in particular to the US, to guarantee its defence.
In 1991 a loose defence pact was signed, giving the US rights to base troops and equipment in the UAE.
Since then the UAE (in fact mainly Abu Dhabi) has spent billions of dollars on buying sophisticated military equipment, but it lacks the personnel and skills to make full use of them.
The purchases were essentially designed to draw the US and other Western powers such as France into close arrangements with the UAE.
This table shows the Defence forces, 1996.
Armed forces is _NUM.
Army is _NUM.
Navy is _NUM.
Air Force is _NUM.
Source: International Institute for Strategic Studies, The Military.
Compared with other Arab countries, the UAE's armed forces as a percentage of the national population are large.
This is because federal arrangements coexist with individual emirate defence provisions.
Although the different emirates nominally unified their forces in 1976, there is still little federal identity.
Since the end of the Gulf war, the number of foreign nationals in the military has fallen to around 30% of the total.
Defence is a major item of government expenditure.
In 1996 the federal ministries of defence, justice and the interior together spent just over 40% of the total budget.
Abu Dhabi and, to a lesser extent, Dubai make separate defence purchases, with the costs considered a contribution to the federal budget (see Reference table 1).
A significant increase in UAE defence spending has been expected since 1995, when the UAE announced its intention to spend $10bn-12bn on new weaponry.
The procurement programme was held up as a sign of the federation's desire to bolster its defensive capabilities in the face of the perceived Iranian threat and to play a more active role in safeguarding shipping through the strategic straits of Hormuz.
This table shows the Military indicators.
(1996 unless otherwise indicated).
Total active armed forces is _NUM.
Military expenditure ($ m) is _NUM.
Military expenditure (% of GDP) is _NUM.
Military expenditure per head ($) is _NUM.
Arms imports ($ m; 1994) is _NUM.
Arms imports (% total imports; 1994) is _NUM.
Cumulative arms transfers deliveries ($ m; 1992-94) is _NUM.
Armed forces per '000 people (soldiers) is _NUM.
Sources: International Institute for Strategic Studies, The Military.
Balance 1996/97; US Arms Control and Disarmament Agency, World Military-
Despite years of discussions with prospective suppliers, only one major deal has been signed, a $3.4bn contract with the French fighter aircraft firm, Dassault, for 30 new Mirage 2000-09 generation fighters, some in replacement of 22 existing Mirages.
This should leave UAE officials more time to evaluate offers for a further order for 50 aircraft worth some $4.6bn.
Lockheed Martin, a US manufacturer of F-16 aircraft, is the likeliest winner, with the UK's British Aerospace (BAe) and the Eurofighter 2000 next in line.
No award has yet been made for a $1bn order to supply the navy with up to eight ocean-capable patrol boats (OCPBs) and a $1bn-2bn order for six small frigates.
Pressure to award the second contract has eased after the purchase in 1996 of two second-hand frigates from the Netherlands.
These are due to be delivered in the first half of 1998.
The delays in awarding the contracts have been attributed to Abu Dhabi re-evaluating the

ramme and assessing means of alternative financing, including the possibility of payment
.
XT>
)C>
C>
CNO>

)CNO>
>
ıomy
>
XT>
ad Arab Emirates.
ntry Profile Uae 1997 / 1998.
ıomy.
ıomic structure.
table shows the Main economic indicators, 1996.
l GDP growth(a) (%) is _NUM.
sumer price inflation (%) is _NUM.
ent account ($ m) is _NUM.
ırnal debt(a) ($ bn) is _NUM.
ıange rate (Dh:$) is _NUM.
ulation(b) (m) is _NUM.
EIU estimate.
)fficial preliminary estimate.
mainstay of the UAE economy is oil and gas (see Reference tables 10 and 11).
ar the largest oil producer is Abu Dhabi, but some contribution comes from Dubai and, to a
h lesser extent, Sharjah and Ras al-Khaimah (see Reference table 13).
pite oil's importance, its contribution to GDP has been declining in recent years from about
in 1980 to just 35% in 1996.
ertheless, the significance of this decline should not be overstated.
h of the non-oil economy depends on the public sector and public-sector contracts, and the
ıunt of public spending is directly related to oil revenue.
emment consumption is a major element in demand in the economy, equivalent to over 16
f GDP (see Reference tables 8 and 9).
ık oil prices during the period 1993-95 were reflected in a low average level of GDP growth
Reference table 7).
table shows the Gross domestic product by emirate.
rent prices).
5 is _NUM.
m is _TABLE % share .Dh m % share.
Dhabi is _NUM.
ai is _NUM.
ırjah is _NUM.
al-Khaimah is _NUM.
ıan is _NUM.
ıirah is _NUM.
m al-Qaiwain is _NUM.
al is _NUM.
ırces: Federal Ministry of Planning, Annual Economic Report 1996; UAE.
ai is the centre for regional trade.
mportance to the region in this respect has been growing, especially since the Gulf war, and
extends as far as the southern republics of the former Soviet Union.
exports are the mainstay of the trading system, with Dubai's foreign trade more than twice
value of its own national GDP.
s table shows the Re-exports by emirate.
n).
)5 is _NUM.
ai is _NUM.
ırjah is _NUM.
Dhabi is _NUM.
ıer (mainly non-recorded) is _NUM.
ai is _NUM.
e value of UAE re-exports in 1996 was around $10.1bn (30% of total exports).
ignificant proportion-about 40%-is unrecorded.
is comprises goods leaving Dubai by dhow (a vessel characteristic of the Gulf) and
rchandise passing through the Jebel Ali Free Zone, also in Dubai.
otal, Dubai accounts for around 85% of re-exports.
is table shows the UAE merchandise exports.
m).
95 is _NUM.
tal is _NUM.
ude oil is _NUM.
s is _NUM.
tal oil & gas is _NUM.
-exports is _NUM.
Other non-oil exports is _NUM.
Total non-oil exports is _NUM.
The UAE economy is rather larger than Kuwait's, but less than one-third the size of Saudi Arabia's.
Its GDP per head is broadly similar to Kuwait's, but it is well behind that of leading industrial countries.
This table shows the Comparative economic indicators, 1996.
UAE is _TABLE Saudi Arabia Kuwait Japan.
GDP ($ bn) is _NUM.
GDP per head ($) is _NUM.
Oil production ('000 b/d) is _NUM.
Exports of goods ($ bn) is _NUM.
Imports of goods ($ bn) is _NUM.
</TEXT>
</DOC>

APPENDIX C

2
ghan Afghanistan
ghani Afghanistan
banian Albania
gerian Algeria
erican US
dorran Andorra
golan Angola
guillan Anguille
tiguan Antigua
tilles Netherland-Antilles
gentine Argentina
gentinian Argentina
uban Aruba
stralian Australia
strian Austria
erbaijani Azerbaijan
hamian Bahamas
hraini Bahrain
ngladeshi Bangladesh
rbadian Barbados
sothian Basotho
tswanian Batswana
larusian Belarus
lgian Belgium
lizean Belize
ninese Benin
rmudan Bermuda
utanese Bhutan
olivian Bolivia
osnia Bosnia-and-Hercegovina
osnian Bosnia-and-Hercegovina
otswanian Botswana
razilian Brazil
razzaville-Congolese Congo-(Brazzaville)
ritish United-Kingdom
runeian Brunei
ulgarian Bulgaria
urkina Burkina-Faso
urkinabe Burkina-Faso
urma Myanmar-(Burma)
urmese Myanmar-(Burma)
urundian Burundi
ambodian Cambodia
ameroonian Cameroon
anadian Canada
ape-Verdian Cape-Verde
ayman Cayman-Islands
entral-African Central-African-Republic
hadian Chad
hilean Chile
hinese China
olombian Colombia
omoran Comoros
ongo Congo-(Democratic-Republic)
ongolese Congo-(Democratic-Republic)
osta-Rican Costa-Rica
ote-D'Ivoire Cote-d`Ivoire
roatian Croatia
uban Cuba

ypriot Cyprus
zech Czech-Republic
zechoslovak Czech-Republic
zechoslovakian Czech-Republic
ane Denmark
anish Denmark
emocratic-Republic-of-the-Congo Congo-(Democratic-Republic)
jiboutian Djibouti
ominican Dominican-Republic
utch Netherlands
cuadorian Ecuador
gyptian Egypt
nglish United-Kingdom
nglishman United-Kingdom
nglishmen United-Kingdom
nglishwoman United-Kingdom
nglishwomen United-Kingdom
quatoguinean Equatorial-Guinea
ritrean Eritrea
stonian Estonia
thiopian Ethiopia
ijian Fiji
innish Finland
rench France
abonese Gabon
ambian The-Gambia
erman Germany
hanaian Ghana
reek Greece
reenlandic Greenland
renadian Grenada
uaman Guam
uatemalan Guatemala
uianese Guiana
uinea-Bissauan Guinea-Bissau
uinean Guinea
uyanese Guyana
aitian Haiti
elenian Greece
ercegovina Bosnia-and-Hercegovina
onduran Honduras
ong-Kongnese Hong-Kong
ungarian Hungary
celander Iceland
celandic Iceland
ndian India
ndonesian Indonesia
ranian Iran
raqi Iraq
rish Ireland
rishman Ireland
rishwoman Ireland
sraeli Israel
talian Italy
vorian Ivory-Coast
Jamaican Jamaica
Japanese Japan
Jordanian Jordan
Kazak Kazakhstan
Kazakh Kazakhstan

akstan Kazakhstan
yan Kenya
ean South-Korea
ovo Bosnia-and-Hercegovina
aiti Kuwait
gyz Kyrgyzstan
tian Laos
vian Latvia
anese Lebanon
othian Lesotho
erian Liberia
yan Libya
chtensteiner Liechtenstein
huanian Lithuania
embourger Luxembourg
ia Libya
ian Libya
auan Macau
edonian Macedonia
agascan Madagascar
oran Mahora
awian Malawi
aysian Malaysia
divian Maldives
ian Mali
tese Malta
shallese Marshall-Islands
tiniquais Martinique
ritanian Mauritania
ritian Mauritius
ican Mexico
davia Moldova
davian Moldova
dovan Moldova
acan Monaco
golian Mongolia
tenegro Yugoslavia
tserratian Montserrat
occan Morocco
swanian Motswana
zambican Mozambique
nibian Namibia
palese Nepal
v-Caledonian New-Caledonia
v-Zealander New-Zealand
caraguan Nicaragua
gerian Nigeria
gerien Niger
rth-Korean North-Korea
rwegian Norway
ani Oman
cific-Islander Pacific-Islands
kistani Pakistan
lestine The-Occupied-Territories
lestinian The-Occupied-Territories
namanian Panama
pua-New-Guinean Papua-New-Guinea
puan Papua-New-Guinea
raguayan Paraguay
ruvian Peru

ilippine Philippines
ilippino Philippines
lish Poland
rtuguese Portugal
erto-Rican Puerto-Rico
tari Qatar
manian Romania
ssian Russia
ssian-Federation Russia
andan Rwanda
int-Lucian Saint-Lucia
lvadoran El-Salvador
moan Samoa
nmarinese San-Marino
o-Tomean Sao-Tome-and-Principe
udi Saudi-Arabia
udi-Arabian Saudi-Arabia
otland United-Kingdom
ottish United-Kingdom
negalese Senegal
rbia Yugoslavia
rbia-Montenegro Yugoslavia
rbian Yugoslavia
ychellois Seychelles
erra-Leonean Sierra-Leone
ngaporean Singapore
ovak Slovakia
ovakian Slovakia
ovenian Slovenia
lomon-Islander Solomon-Islands
mali Somalia
uth-African South-Africa
uth-Korean South-Korea
viet Russia
viet-Union Russia
paniard Spain
panish Spain
i-Lankan Sri-Lanka
danese Sudan
rinam Suriname
rinamese Suriname
vazi Swaziland
vede Sweden
vedish Sweden
viss Switzerland
yrian Syria
aiwanese Taiwan
ajik Tajikistan
ajiki Tajikistan
anzanian Tanzania
hai Thailand
imorese East-Timor
ogolese Togo
ongan Tonga
rinidad Trinidad-and-Tobago
rinidadian Trinidad-and-Tobago
unisian Tunisia
urk Turkey
urk-Islander Turks-and-Caicos-Islands
urkish Turkey

ırkmen Turkmenistan
K. United-Kingdom
S. US
S.A. US
S.S.R. Russia
{ United-Kingdom
; US
;A US
;SR Russia
;SR Russia
jandan Uganda
krainian Ukraine
kranian Ukraine
ıited-States US
ıited-States-of-America US
ruguayan Uruguay
zbek Uzbekistan
anuatuan Vanuatu
enezuelan Venezüela
ietnamese Vietnam
irgin-Islander British-Virgin-Islands
est-Samoan Western-Samoa
emeni Yemen
ıgoslav Yugoslavia
ugoslavian Yugoslavia
airean Zaire
ambian Zambia
imbabwean Zimbabwe
fghan Afghanistan
fghani Afghanistan
fghanistan Afghanistan
lbania Albania
lbanian Albania
lgeria Algéria
lgerian Algeria
merican US
ndorra Andorra
ndorran Andorra
ngola Angola
ngolan Angola
nguillan Anguille
nguille Anguille
ntigua Antigua
ntiguan Antigua
ntilles Netherland-Antilles
rgentina Argentina
rgentine Argentina
rgentinian Argentina
ruba Aruba
ruban Aruba
ustralia Australia
ustralian Australia
ustria Austria
ustrian Austria
zerbaijan Azerbaijan
zerbaijani Azerbaijan
ahamas Bahamas
ahamian Bahamas
ahrain Bahrain
ahraini Bahrain

ngladesh Bangladesh
ngladeshi Bangladesh
rbadian Barbados
rbados Barbados
sothian Basotho
sotho Basotho
tswana Batswana
tswanian Batswana
larus Belarus
larusian Belarus
lgian Belgium
lgium Belgium
lize Belize
lizean Belize
nin Benin
ninese Benin
rmuda Bermuda
rmudan Bermuda
utan Bhutan
utanese Bhutan
livia Bolivia
livian Bolivia
snia Bosnia-and-Hercegovina
snia-and-hercegovina Bosnia-and-Hercegovina
snia-and-hercegovina Bosnia-and-Hercegovina
snia-and-hercegovina Bosnia-and-Hercegovina
snian Bosnia-and-Hercegovina
tswana Botswana
tswanian Botswana
razil Brazil
razilian Brazil
razzaville-congolese Congo-(Brazzaville)
ritish United-Kingdom
ritish-virgin-islands British-Virgin-Islands
runei Brunei
runeian Brunei
ulgaria Bulgaria
ulgarian Bulgaria
urkina Burkina-Faso
urkina-faso Burkina-Faso
urkinabe Burkina-Faso
urma Myanmar-(Burma)
urmese Myanmar-(Burma)
urundi Burundi
urundian Burundi
ambodia Cambodia
ambodian Cambodia
ameroon Cameroon
ameroonian Cameroon
anada Canada
anadian Canada
ape-verde Cape-Verde
ape-verdian Cape-Verde
ayman Cayman-Islands
ayman-islands Cayman-Islands
entral-african Central-African-Republic
entral-african-republic Central-African-Republic
had Chad
hadian Chad
hile Chile

hilean Chile
hina China
hinese China
olombia Colombia
olombian Colombia
omoran Comoros
omoros Comoros
ongo Congo-(Democratic-Republic)
ongo-(brazzaville) Congo-(Brazzaville)
ongo-(democratic-republic) Congo-(Democratic-Republic)
ongo-(democratic-republic) Congo-(Democratic-Republic)
ongolese Congo-(Democratic-Republic)
osta-rica Costa-Rica
osta-rican Costa-Rica
ote-d'ivoire Cote-d`Ivoire
ote-d'ivoire Cote-d`Ivoire
roatia Croatia
roatian Croatia
uba Cuba
uban Cuba
ypriot Cyprus
yprus Cyprus
zech Czech-Republic
zech-republic Czech-Republic
zechoslovak Czech-Republic
zechoslovakian Czech-Republic
lane Denmark
lanish Denmark
lemocratic-republic-of-the-congo Congo-(Democratic-Republic)
lenmark Denmark
ljibouti Djibouti
ljiboutian Djibouti
lominican Dominican-Republic
lominican-republic Dominican-Republic
lutch Netherlands
ast-timor East-Timor
cuador Ecuador
cuadorian Ecuador
gypt Egypt
gyptian Egypt
l-salvador El-Salvador
nglish United-Kingdom
nglishman United-Kingdom
nglishmen United-Kingdom
nglishwoman United-Kingdom
nglishwomen United-Kingdom
quatoguinean Equatorial-Guinea
quatorial-guinea Equatorial-Guinea
ritrea Eritrea
ritrean Eritrea
estonia Estonia
estonian Estonia
ethiopia Ethiopia
ethiopian Ethiopia
fiji Fiji
fijian Fiji
finland Finland
finnish Finland
france France
french France

ıbon Gabon
ıbonese Gabon
ımbian The-Gambia
ɜrman Germany
ɜrmany Germany
ıana Ghana
ıanaian Ghana
ɜeece Greece
ɜeece Greece
ɜeek Greece
ɜeenland Greenland
ɜeenlandic Greenland
ɜenada Grenada
ɜenadian Grenada
ıam Guam
ıaman Guam
ıatemala Guatemala
ıatemalan Guatemala
ıiana Guiana
ıianese Guiana
ıinea Guinea
ıinea-bissau Guinea-Bissau
ıinea-bissauan Guinea-Bissau
ıinean Guinea
ıyana Guyana
ıyanese Guyana
aiti Haiti
aitian Haiti
elenian Greece
ercegovina Bosnia-and-Hercegovina
onduran Honduras
onduras Honduras
ong-kong Hong-Kong
ong-kongnese Hong-Kong
ungarian Hungary
ungary Hungary
celand Iceland
celander Iceland
celandic Iceland
ndia India
ndian India
ndonesia Indonesia
ndonesian Indonesia
ran Iran
ranian Iran
raq Iraq
raqi Iraq
reland Ireland
rish Ireland
rishman Ireland
rishwoman Ireland
srael Israel
sraeli Israel
talian Italy
taly Italy
vorian Ivory-Coast
vory-coast Ivory-Coast
jamaica Jamaica
jamaican Jamaica
japan Japan

)anese Japan
:dan Jordan
:danian Jordan
:ak Kazakhstan
:akh Kazakhstan
:akhstan Kazakhstan
:akstan Kazakhstan
:ya Kenya
:yan Kenya
:ean South-Korea
:ovo Bosnia-and-Hercegovina
:wait Kuwait
:waiti Kuwait
:gyz Kyrgyzstan
:gyzstan Kyrgyzstan
)s Laos
)tian Laos
tvia Latvia
tvian Latvia
)anese Lebanon
)anon Lebanon
sothian Lesotho
sotho Lesotho
)eria Liberia
)erian Liberia
)ya Libya
)yan Libya
echtenstein Liechtenstein
echtensteiner Liechtenstein
thuania Lithuania
thuanian Lithuania
xembourg Luxembourg
xembourger Luxembourg
)ia Libya
)ian Libya
cau Macau
cauan Macau
cedonia Macedonia
cedonian Macedonia
dagascan Madagascar
dagascar Madagascar
hora Mahora
horan Mahora
lawi Malawi
lawian Malawi
laysia Malaysia
laysian Malaysia
ldives Maldives
ldivian Maldives
li Mali
lian Mali
lta Malta
ltese Malta
arshall-islands Marshall-Islands
arshallese Marshall-Islands
artiniquais Martinique
artinique Martinique
auritania Mauritania
auritanian Mauritania
auritian Mauritius

uritius Mauritius
xican Mexico
xico Mexico
ldavia Moldova
ldavian Moldova
ldova Moldova
ldovan Moldova
nacan Monaco
naco Monaco
ngolia Mongolia
ngolian Mongolia
ntenegro Yugoslavia
ntserrat Montserrat
ntserratian Montserrat
roccan Morocco
rocco Morocco
tswana Motswana
tswanian Motswana
zambican Mozambique
zambique Mozambique
anmar-(burma) Myanmar-(Burma)
mibia Namibia
mibian Namibia
pal Nepal
palese Nepal
therland-antilles Netherland-Antilles
therlands Netherlands
w-caledonia New-Caledonia
w-caledonian New-Caledonia
w-zealand New-Zealand
w-zealander New-Zealand
caragua Nicaragua
caraguan Nicaragua
ger Niger
geria Nigeria
gerian Nigeria
gerien Niger
rth-korea North-Korea
rth-korean North-Korea
rway Norway
rwegian Norway
man Oman
mani Oman
acific-islander Pacific-Islands
acific-islands Pacific-Islands
akistan Pakistan
akistani Pakistan
alestine The-Occupied-Territories
alestinian The-Occupied-Territories
anama Panama
anamanian Panama
apua-new-guinea Papua-New-Guinea
apua-new-guinean Papua-New-Guinea
apuan Papua-New-Guinea
araguay Paraguay
araguayan Paraguay
eru Peru
eruvian Peru
hilippine Philippines
hilippines Philippines

```
hilippino       Philippines
oland   Poland
olish   Poland
ortugal         Portugal
ortuguese       Portugal
uerto-rican     Puerto-Rico
uerto-rico      Puerto-Rico
atar    Qatar
atari   Qatar
omania Romania
omanian         Romania
ussia   Russia
ussia   Russia
ussian Russia
ussian-federation       Russia
wanda   Rwanda
wandan Rwanda
aint-lucia      Saint-Lucia
aint-lucian     Saint-Lucia
alvadoran       El-Salvador
amoa    Samoa
amoan   Samoa
an-marino       San-Marino
anmarinese      San-Marino
ao-tome-and-principe    Sao-Tome-and-Principe
ao-tomean       Sao-Tome-and-Principe
audi    Saudi-Arabia
audi-arabia     Saudi-Arabia
audi-arabian    Saudi-Arabia
cotland         United-Kingdom
cottish         United-Kingdom
enegal Senegal
enegalese       Senegal
erbia   Yugoslavia
erbia-montenegro        Yugoslavia
erbian Yugoslavia
eychelles       Seychelles
eychellois      Seychelles
ierra-leone     Sierra-Leone
ierra-leonean   Sierra-Leone
ingapore        Singapore
ingaporean      Singapore
lovak   Slovakia
lovakia         Slovakia
lovakian        Slovakia
lovenia         Slovenia
lovenian        Slovenia
olomon-islander         Solomon-Islands
olomon-islands Solomon-Islands
omali   Somalia
omalia Somalia
outh-africa     South-Africa
outh-african    South-Africa
outh-korea      South-Korea
outh-korea      South-Korea
outh-korean     South-Korea
oviet   Russia
oviet-union     Russia
pain    Spain
paniard         Spain
```

```
panish Spain
ri-lanka        Sri-Lanka
ri-lankan       Sri-Lanka
udan    Sudan
udanese         Sudan
urinam Suriname
uriname         Suriname
urinamese       Suriname
wazi    Swaziland
waziland        Swaziland
wede    Sweden
weden   Sweden
wedish Sweden
witzerland      Switzerland
yria    Syria
yrian   Syria
aiwan   Taiwan
aiwanese        Taiwan
ajik    Tajikistan
ajiki   Tajikistan
ajikistan       Tajikistan
anzania         Tanzania
anzanian        Tanzania
hai     Thailand
hailand         Thailand
he-US   US
he-USA US
he-gambia       The-Gambia
he-occupied-territories         The-Occupied-Territorie
he-us   the-US
he-usa the-US
imorese         East-Timor
ogo     Togo
ogolese         Togo
onga    Tonga
ongan   Tonga
rinidad         Trinidad-and-Tobago
rinidad-and-tobago      Trinidad-and-Tobago
rinidadian      Trinidad-and-Tobago
unisia Tunisia
unisian         Tunisia
urk     Turkey
urk-islander    Turks-and-Caicos-Islands
urkey   Turkey
urkish Turkey
urkmen Turkmenistan
urkmenistan     Turkmenistan
urks-and-caicos-islands         Turks-and-Caicos-Islanc
.k.     U.K.
.s.     US
.s.a.   US
.s.s.r.         U.S.S.R.
ganda   Uganda
gandan Uganda
k       UK
kraine Ukraine
krainian        Ukraine
kranian         Ukraine
nited-kingdom   United-Kingdom
```

```
nited-kingdom  United-Kingdom
nited-kingdom  United-Kingdom
nited-states   US
nited-states-of-america        US
nited-states-of-america        US
ruguay Uruguay
ruguayan       Uruguay
sa     US
ssr    USSR
zbek   Uzbekistan
zbekistan      Uzbekistan
anuatu Vanuatu
anuatuan       Vanuatu
enezuela       Venezuela
enezuelan      Venezuela
ietnam Vietnam
ietnamese      Vietnam
irgin-islander British-Virgin-Islands
est-samoan     Western-Samoa
estern-samoa   Western-Samoa
emen   Yemen
emeni  Yemen
ugoslav        Yugoslavia
ugoslavia      Yugoslavia
ugoslavia      Yugoslavia
ugoslavia      Yugoslavia
ugoslavian     Yugoslavia
aire   Zaire
airean Zaire
ambia  Zambia
ambian Zambia
imbabwe        Zimbabwe
imbabwean      Zimbabwe
```

CLAIMS

1. A method in a computer system for transforming a document of a data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

for each sentence,

parsing the sentence to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful terms of the sentence from the syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term;

determining an additional grammatical role for at least one of the meaningful terms, such that the at least one meaningful term is associated with at least two different grammatical roles; and

storing in an enhanced data representation data structure a representation of each association between a meaningful term and its determined grammatical roles, in a manner that indicates a grammatical relationship between a plurality of the meaningful terms and such that at least one meaningful term is associated with a plurality of grammatical relationships.

2. The method of claim 1 wherein heuristics are used to determine the additional grammatical role for the at least one of the meaningful terms.

3. The method of claim 2 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with an object as the additional grammatical role.

4. The method of claim 2 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with a subject as the additional grammatical role.

5. The method of claim 2 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with a verb as the additional grammatical role.

6. The method of claim 2 wherein a meaningful term is associated with a subject as the determined grammatical role and is associated with an object as the additional grammatical role.

7. The method of claim 2 wherein a meaningful term is associated with a object as the determined grammatical role and is associated with a subject as the additional grammatical role.

8. The method of claim 2 wherein a meaningful term is associated with a noun modifier as the determined grammatical role and is associated with a subject as the additional grammatical role.

9. The method of claim 2 wherein a meaningful term is associated with a noun modifier as the determined grammatical role and is associated with an object as the additional grammatical role.

10. The method of claim 1 wherein the determined additional grammatical role is a part of grammar that is not implied by the position of the at least one meaningful term relative to the structure of the sentence.

11. The method of claim 1 wherein heuristics are used to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

12. The method of claim 1 wherein the determining the grammatical role for each meaningful term and the determining of the additional grammatical role for at least one of the meaningful terms yields a plurality of grammatical relationships between meaningful terms that are identical.

13. The method of claim 1 wherein the determining of a grammatical role for each meaningful term includes determining whether the term is at least one of a subject, object, verb, part of a prepositional phrase, noun modifier, and verb modifier.

14. The method of claim 1 wherein the document is part of a corpus of heterogeneous documents.

15. The method of claim 1 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

16. The method of claim 1 wherein the enhanced data representation data structure is used to index a corpus of documents.

17. The method of claim 1 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

18. The method of claim 17 wherein results are returned that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

19. The method of claim 18 wherein the objects in the corpus are sentences and sentences are returned that satisfy the query.

20. The method of claim 18, further comprising returning paragraphs that contain similar terms to those found in an indicated sentence.

21. The method of claim 18, further comprising returning documents that contain similar terms to those found in an indicated sentence.

22. The method of claim 17 wherein terms that are associated with designated grammatical roles are returned for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

23. The method of claim 22 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

24. The method of claim 17 further comprising adding additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

25. The method of claim 24 wherein heuristics are used to determine the additional grammatical relationships.

26. The method of claim 24 wherein at least one of entailed verbs and related verbs are used to add additional grammatical relationships.

27. The method of claim 17 wherein weighted results are returned that satisfy the query.

28. A computer-readable memory medium containing instructions for controlling a computer processor to transform a document of a data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, by:

for each sentence,

parsing the sentence to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful terms of the sentence from the syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term;

determining an additional grammatical role for at least one of the meaningful terms, such that the at least one meaningful term is associated with at least two different grammatical roles; and

storing in an enhanced data representation data structure a representation of each association between a meaningful term and its determined grammatical roles, in a manner that indicates a grammatical relationship between a plurality of the meaningful terms and such that at least one meaningful term is associated with a plurality of grammatical relationships.

29. The computer-readable memory medium of claim 28 wherein heuristics are used to determine the additional grammatical role for the at least one of the meaningful terms.

30. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with an object as the additional grammatical role.

31. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with a subject as the additional grammatical role.

32. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a verb modifier as the determined grammatical role and is associated with a verb as the additional grammatical role.

33. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a subject as the determined grammatical role and is associated with an object as the additional grammatical role.

34. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a object as the determined grammatical role and is associated with a subject as the additional grammatical role.

35. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a noun modifier as the determined grammatical role and is associated with a subject as the additional grammatical role.

36. The computer-readable memory medium of claim 29 wherein a meaningful term is associated with a noun modifier as the

determined grammatical role and is associated with an object as the additional grammatical role.

37. The computer-readable memory medium of claim 28 wherein the determined additional grammatical role is a part of grammar that is not implied by the position of the at least one meaningful term relative to the structure of the sentence.

38. The computer-readable memory medium of claim 28 wherein heuristics are used to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

39. The computer-readable memory medium of claim 28 wherein the determining the grammatical role for each meaningful term and the determining of the additional grammatical role for at least one of the meaningful terms yields a plurality of grammatical relationships between meaningful terms that are identical.

40. The computer-readable memory medium of claim 28 wherein the determining of a grammatical role for each meaningful term includes determining whether the term is at least one of a subject, object, verb, part of a prepositional phrase, noun modifier, and verb modifier.

41. The computer-readable memory medium of claim 28 wherein the document is part of a corpus of heterogeneous documents.

42. The computer-readable memory medium of claim 28 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

43. The computer-readable memory medium of claim 28 wherein the enhanced data representation data structure is used to index a corpus of documents.

44. The computer-readable memory medium of claim 28 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

45. The computer-readable memory medium of claim 44 wherein results are returned that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

46. The computer-readable memory medium of claim 45 wherein the objects in the corpus are sentences and sentences are returned that satisfy the query.

47. The computer-readable memory medium of claim 45, the instructions further controlling the computer processor by returning paragraphs that contain similar terms to those found in an indicated sentence.

48. The computer-readable memory medium of claim 45, the instructions further controlling the computer processor by returning documents that contain similar terms to those found in an indicated sentence.

49. The computer-readable memory medium of claim 44 wherein terms that are associated with designated grammatical roles are returned for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

50. The computer-readable memory medium of claim 49 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

51. The computer-readable memory medium of claim 44, the instructions further controlling the computer processor by adding additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

52. The computer-readable memory medium of claim 51 wherein heuristics are used to determine the additional grammatical relationships.

53. The computer-readable memory medium of claim 51 wherein at least one of entailed verbs and related verbs are used to add additional grammatical relationships.

54. The computer-readable memory medium of claim 44 wherein weighted results are returned that satisfy the query.

55. A syntactic query engine for transforming a document of a data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

parser that is structured to decompose each sentence to generate a parse structure for the sentence having a plurality of syntactic elements; and

postprocessor that is structured to

receive from the parser the parse structure of the sentence;

determine a set of meaningful terms of the sentence from the syntactic elements;

determine from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term;

determine an additional grammatical role for at least one of the meaningful terms, such that the at least one meaningful term is associated with at least two different grammatical roles; and

store, in an enhanced data representation data structure, a representation of each association between a meaningful term and its determined grammatical roles, in a manner that indicates a grammatical relationship between a plurality of the meaningful terms and such that at least one meaningful term is associated with a plurality of grammatical relationships.

56. The query engine of claim 55 wherein the postprocessor uses heuristics to determine the additional grammatical role for the at least one of the meaningful terms.

57. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a verb modifier as the determined grammatical role and with an object as the additional grammatical role.

58. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a verb modifier as the determined grammatical role and with a subject as the additional grammatical role.

59. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a verb modifier as the determined grammatical role and with a verb as the additional grammatical role.

60. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a subject as the determined grammatical role and with an object as the additional grammatical role.

61. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a object as the determined grammatical role and with a subject as the additional grammatical role.

62. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a noun modifier as the determined grammatical role and with a subject as the additional grammatical role.

63. The query engine of claim 56 wherein the postprocessor associates a meaningful term with a noun modifier as the determined grammatical role and with an object as the additional grammatical role.

64. The query engine of claim 55 wherein the determined additional grammatical role is a part of grammar that is not implied by the position of the at least one meaningful term relative to the structure of the sentence.

65. The query engine of claim 55 wherein the postprocessor uses heuristics to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

66. The query engine of claim 55 wherein the determining the grammatical role for each meaningful term and the determining of the additional grammatical role for at least one of the meaningful terms yields a plurality of grammatical relationships between meaningful terms that are identical.

67. The query engine of claim 55 wherein the determining of a grammatical role for each meaningful term includes determining whether the term is at least one of a subject, object, verb, part of a prepositional phrase, noun modifier, and verb modifier.

68. The query engine of claim 55 wherein the document is part of a corpus of heterogeneous documents.

69. The query engine of claim 55 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

70. The query engine of claim 55 wherein the enhanced data representation data structure is used to index a corpus of documents.

71. The query engine of claim 55, further comprising a query processor that uses the enhanced data representation data structure to execute a query against objects in a corpus of documents.

72. The query engine of claim 71 wherein the query processor returns results that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

73. The query engine of claim 72 wherein the objects in the corpus are sentences and the query processor returns sentences that satisfy the query.

74. The query engine of claim 72 wherein the query processor returns paragraphs that contain similar terms to those found in an indicated sentence.

75. The query engine of claim 72 wherein the query processor returns documents that contain similar terms to those found in an indicated sentence.

76. The query engine of claim 71 wherein the query processor returns terms that are associated with designated grammatical roles for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

77. The query engine of claim 76 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

78. The query engine of claim 71 wherein the query processor adds dding additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

79. The query engine of claim 78 wherein heuristics are used to determine the additional grammatical relationships.

80. The query engine of claim 78 wherein the query processor uses at least one of entailed verbs and related verbs to add additional grammatical relationships.

81. The query engine of claim 71 wherein the query processor returns weighted results that satisfy the query.

82. A method in a computer system for transforming a document of a data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

for each sentence,

parsing the sentence to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful terms of the sentence from these syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term, wherein at least one of the grammatical roles for a meaningful term is at least one of a verb modifier of a prepositional phrase and a noun modifier of a noun phrase; and

storing in an enhanced data representation data structure a representation of each meaningful term associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

83. The method of claim 82, further comprising storing the full grammar of the sentence.

84. The method of claim 82, further comprising, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, associating the one of the meaningful terms with a subject grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as a subject of the sentence.

85. The method of claim 84 wherein the noun modifier is a modifier of a noun that is used as an object of the sentence.

86. The method of claim 84 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

87. The method of claim 82, further comprising, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, associating the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

88. The method of claim 87 wherein the noun modifier is a modifier of a noun that is stored as an object of the sentence.

89. The method of claim 87 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

90. The method of claim 82, further comprising, when it is determined that a verb modifier of a prepositional phrase is a grammatical role associated with one of the meaningful terms, associating the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

91. The method of claim 82 wherein heuristics are used to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

92. The method of claim 82 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the enhanced data representation data structure.

93. The method of claim 82 wherein the document is part of a corpus of heterogeneous documents.

94. The method of claim 82 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

95. The method of claim 82 wherein the enhanced data representation data structure is used to index a corpus of documents.

96. The method of claim 82 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

97. The method of claim 96 wherein results are returned that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

98. The method of claim 97 wherein the objects in the corpus are sentences and sentences are returned that satisfy the query.

99. The method of claim 97, further comprising returning paragraphs that contain similar terms to those found in an indicated sentence.

100. The method of claim 97, further comprising returning documents that contain similar terms to those found in an indicated sentence.

101. The method of claim 96 wherein terms that are associated with designated grammatical roles are returned for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

102. The method of claim 101 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

103. The method of claim 96 further comprising adding additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

104. The method of claim 103 wherein heuristics are used to determine the additional grammatical relationships.

105. The method of claim 103 wherein at least one of entailed verbs and related verbs are used to add additional grammatical relationships.

106. The method of claim 96 wherein weighted results are returned that satisfy the query.

107. A computer-readable memory medium containing instructions for controlling a computer processor to transform a document of a

data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, by:

for each sentence,

parsing the sentence to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful terms of the sentence from these syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term, wherein at least one of the grammatical roles for a meaningful term is at least one of a verb modifier of a prepositional phrase and a noun modifier of a noun phrase; and

storing in an enhanced data representation data structure a representation of each meaningful term associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

108. The computer-readable memory medium of claim 107, the instructions further controlling the computer processor to store the full grammar of the sentence.

109. The computer-readable memory medium of claim 107, the instructions further controlling the computer processor by, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, associating the one of the meaningful terms with a subject grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as a subject of the sentence.

110. The computer-readable memory medium of claim 109 wherein the noun modifier is a modifier of a noun that is used as an object of the sentence.

111. The computer-readable memory medium of claim 109 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

112. The computer-readable memory medium of claim 107, the instructions further controlling the computer processor by, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, associating the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

113. The computer-readable memory medium of claim 112 wherein the noun modifier is a modifier of a noun that is stored as an object of the sentence.

114. The computer-readable memory medium of claim 109 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

115. The computer-readable memory medium of claim 107, the instructions further controlling the computer processor by, when it is determined that a verb modifier of a prepositional phrase is a grammatical role associated with one of the meaningful terms, associating the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

116. The computer-readable memory medium of claim 107 wherein heuristics are used to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

117. The computer-readable memory medium of claim 107 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the enhanced data representation data structure.

118. The computer-readable memory medium of claim 107 wherein the document is part of a corpus of heterogeneous documents.

119. The computer-readable memory medium of claim 107 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

120. The computer-readable memory medium of claim 107 wherein the enhanced data representation data structure is used to index a corpus of documents.

121. The computer-readable memory medium of claim 107 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

122. The computer-readable memory medium of claim 121 wherein results are returned that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

123. The computer-readable memory medium of claim 122 wherein the objects in the corpus are sentences and sentences are returned that satisfy the query.

124. The computer-readable memory medium of claim 122, the instructions further controlling the computer processor to return paragraphs that contain similar terms to those found in an indicated sentence.

125. The computer-readable memory medium of claim 122, the instructions further controlling the computer processor to return documents that contain similar terms to those found in an indicated sentence.

126. The computer-readable memory medium of claim 121 wherein terms that are associated with designated grammatical roles are returned for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

127. The computer-readable memory medium of claim 126 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

128. The computer-readable memory medium of claim 121, the instructions further controlling the computer processor to add additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

129. The computer-readable memory medium of claim 128 wherein heuristics are used to determine the additional grammatical relationships.

130. The computer-readable memory medium of claim 128 wherein at least one of entailed verbs and related verbs are used to add additional grammatical relationships.

131. The computer-readable memory medium of claim 121 wherein weighted results are returned that satisfy the query.

132. A syntactic query engine for transforming a document of a data set into a canonical representation, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

parser that is structured to decompose each sentence to generate a parse structure for the sentence having a plurality of syntactic elements; and

postprocessor that is structured to

receive from the parser the parse structure of the sentence;

determine a set of meaningful terms of the sentence from the syntactic elements;

determine from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful term, wherein at least one of the grammatical roles for a meaningful term is at least one of a verb modifier of a prepositional phrase and a noun modifier of a noun phrase; and

store in an enhanced data representation data structure a representation of each meaningful term associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

133. The query engine of claim 132 wherein the postprocessor stores the full grammar of the sentence.

134. The query engine of claim 132 wherein the postprocessor, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, is further structured to associate the one of the meaningful terms with a subject grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as a subject of the sentence.

135. The query engine of claim 134 wherein the noun modifier is a modifier of a noun that is used as an object of the sentence.

136. The query engine of claim 134 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

137. The query engine of claim 132 wherein the postprocessor, when it is determined that a noun modifier grammatical role is associated with one of the meaningful terms, is further structured to associate the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

138. The query engine of claim 137 wherein the noun modifier is a modifier of a noun that is stored as an object of the sentence.

139. The query engine of claim 137 wherein the noun modifier is a modifier of a noun that is used as a subject of the sentence.

140. The query engine of claim 132 wherein the postprocessor, when it is determined that a verb modifier of a prepositional phrase is a grammatical role associated with one of the meaningful terms, is further structured to associate the one of the meaningful terms with an object grammatical role, thereby indicating that the one of the meaningful terms is to be stored also as an object of the sentence.

141. The query engine of claim 132 wherein heuristics are used to determine which grammatical relationships are to be stored in the enhanced data representation data structure.

142. The query engine of claim 132 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the enhanced data representation data structure.

143. The query engine of claim 132 wherein the document is part of a corpus of heterogeneous documents.

144. The query engine of claim 132 wherein the document comprises text and graphics and a sentence is created to correspond to and to describe each portion of graphics.

145. The query engine of claim 132 wherein the enhanced data representation data structure is used to index a corpus of documents.

146. The query engine of claim 132 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

147. The query engine of claim 146, further comprising a query processors that returns results that satisfy the query when an object in the corpus contains similar terms associated with similar grammatical roles to the terms and their associated roles as stored in the enhanced data representation.

148. The query engine of claim 147 wherein the objects in the corpus are sentences and the query processor returns sentences that satisfy the query.

149. The query engine of claim 147 wherein the query processor returns paragraphs that contain similar terms to those found in an indicated sentence.

150. The query engine of claim 147 wherein the query processor returns documents that contain similar terms to those found in an indicated sentence.

151. The query engine of claim 146 wherein the query processor returns terms that are associated with designated grammatical roles for each object in the corpus that contains similar terms associated with similar grammatical roles to the terms and associated roles of designated relationships from the enhanced data representation data structure.

152. The query engine of claim 151 wherein heuristics are used to determine the designated relationships from the enhanced data representation data structure.

153. The query engine of claim 146 wherein the query processor adds additional grammatical relationships to the enhanced data representation data structure to be used to execute a query against objects in a corpus of documents.

154. The query engine of claim 153 wherein heuristics are used to determine the additional grammatical relationships.

155. The query engine of claim 153 wherein the query processor uses at least one of entailed verbs and related verbs to add additional grammatical relationships.

156. The query engine of claim 146 wherein the query processor returns weighted results that satisfy the query.

157. A method in a computer system for storing a normalized data structure representing a document of a data set, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

for each sentence,

determining a set of meaningful terms of the sentence and at least one grammatical role for each meaningful term; and

storing sets of grammatical relationships between a plurality of meaningful terms based upon the determined grammatical role of each meaningful term relative to a meaningful term that is being used as a governing verb, wherein, for each meaningful term that is being used as a governing verb, the normalized data structure contains a set of meaningful terms that are subjects relative to the governing verb, a set of meaningful terms that are objects relative to the governing verb, and at least one of a set of meaningful terms that are verb modifiers of prepositional phrases that contain the governing verb and a set of meaningful terms that are noun modifiers of noun phrases that relate to the governing verb.

158. The method of claim 157, further comprising storing meaningful terms that correspond to a designated attribute.

159. The method of claim 158 wherein the designated attribute is at least one of country name, date, money, amount, number, location, person, corporate name, and organization.

160. The method of claim 157 wherein the sets of meaningful terms are stored as a plurality of tables.

161. The method of claim 160 wherein the tables comprise a subject table, an object table, a subject-object table, and at least one of a preposition table and a noun modifier table.

162. The method of claim 161 wherein the tables further comprise a sentence table that stores the text of the sentence.

163. The method of claim 161 wherein the tables further comprise an attributes table that stores meaningful terms that are associated with designated attributes.

164. The method of claim 160 wherein the preposition table contains for each meaningful term used as a verb in the sentence, a list of meaningful terms that are prepositions of the meaningful term used as the verb

and at least one meaningful term that is a verb modifier associated with each preposition.

165. The method of claim 160 wherein the noun modifier table contains a list of meaningful terms that are noun modifiers of a meaningful terms that is used as a noun in the sentence.

166. The method of claim 157 wherein the tables are tables in a data base.

167. The method of claim 157 wherein the tables are stored as part of a file system.

168. The method of claim 157 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the normalized data structure.

169. The method of claim 157 wherein the document is part of a corpus of heterogeneous documents.

170. The method of claim 157 wherein the normalized data structure is used to index a corpus of documents.

171. The method of claim 157 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

172. A data processing system comprising a computer processor and a memory, the memory containing structured data that stores a normalized representation of sentence data, the structured data being manipulated by the computer processor under the control of program code and stored in the memory as:

a subject table having a set of meaningful term pairs, each pair having a meaningful term that is associated with a grammatical role of a verb and a meaningful term that is associated with a grammatical role of a subject relative to the verb;

an object table having a set of meaningful term pairs, each pair having a meaningful term that is associate with a grammatical role of a verb and a meaningful term that is associated with a grammatical role of an object relative to the verb;

a representation of associations between the subject table and the object table, the representation indicating, for each meaningful term associated with the grammatical role of the verb, the meaningful terms that are associated with the grammatical role of subject relative to the verb and the meaningful terms that are associated with the grammatical role of object relative to the verb;

a preposition table having a set of meaningful term groups, each group having a meaningful term that is associated with a grammatical role of a verb, a meaningful term that is associated with a grammatical role of a preposition relative to the verb, and a meaningful term that is associated with a grammatical role of a verb modifier relative to the verb; and

a noun modifier table having a set of meaningful term pairs, each pair having a meaningful term that is associated with a grammatical role of a noun and a meaningful term that is associated with a grammatical role of an noun modifier relative to the noun.

173. The data processing machine of claim 172 wherein the representation of associations between the between the subject table and the object table is produced by a database join operation.

174. The data processing machine of claim 172 wherein the representation of associations between the between the subject table and the object table is a subject-object table that contains a set of meaningful term groups, each group having a meaningful term that is associated with a grammatical of a verb, a meaningful term that is associated with the grammatical role of a subject relative to the verb; and a meaningful term that is associated with the grammatical role of an object relative to the verb.

175. The data processing machine of claim 172 wherein the structured data is manipulated by the computer processor under the control of program code to query objects of a data set that are indexed as the structured data.

176. The data processing machine of claim 172 wherein the program code decomposes objects of a data set and indexes the decomposed objects in the memory as the structured data.

177. A computer-readable memory medium containing instructions for controlling a computer processor to store a normalized data structure representing a document of a data set, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

for each sentence,

determining a set of meaningful terms of the sentence and at least one grammatical role for each meaningful term; and

storing sets of grammatical relationships between a plurality of meaningful terms based upon the determined grammatical role of each meaningful term relative to a meaningful term that is being used as a governing verb, wherein, for each meaningful term that is being used as a governing verb, the normalized data structure contains a set of meaningful terms that are subjects relative to the governing verb, a set of meaningful terms that are objects relative to the governing verb, and at least one of a set of meaningful terms that are verb modifiers of prepositional phrases that contain the governing verb and a set of meaningful terms that are noun modifiers of noun phrases that relate to the governing verb.

178. The computer-readable memory medium of claim 177, the instructions further controlling the computer processor to store meaningful terms that correspond to a designated attribute.

179. The computer-readable memory medium of claim 178 wherein the designated attribute is at least one of country name, date, money, amount, number, location, person, corporate name, and organization.

180. The computer-readable memory medium of claim 177 wherein the sets of meaningful terms are stored as a plurality of tables.

181. The computer-readable memory medium of claim 180 wherein the tables comprise a subject table, an object table, a subject-object table, and at least one of a preposition table and a noun modifier table.

182. The computer-readable memory medium of claim 181 wherein the tables further comprise a sentence table that stores the text of the sentence.

183. The computer-readable memory medium of claim 181 wherein the tables further comprise an attributes table that stores meaningful terms that are associated with designated attributes.

184. The computer-readable memory medium of claim 180 wherein the preposition table contains for each meaningful term used as a verb in the sentence, a list of meaningful terms that are prepositions of the meaningful term used as the verb and at least one meaningful term that is a verb modifier associated with each preposition.

185. The computer-readable memory medium of claim 180 wherein the noun modifier table contains a list of meaningful terms that are noun modifiers of a meaningful terms that is used as a noun in the sentence.

186. The computer-readable memory medium of claim 177 wherein the tables are tables in a data base.

187. The computer-readable memory medium of claim 177 wherein the tables are stored as part of a file system.

188. The computer-readable memory medium of claim 177 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the normalized data structure.

189. The computer-readable memory medium of claim 177 wherein the document is part of a corpus of heterogeneous documents.

190. The computer-readable memory medium of claim 177 wherein the normalized data structure is used to index a corpus of documents.

191. The computer-readable memory medium of claim 177 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

192. A computer system for storing a normalized data structure representing a document of a data set, the document having a plurality of sentences, each sentence having a plurality of terms, comprising:

enhanced parsing mechanism that determines a set of meaningful terms for each sentence and at least one grammatical role for each meaningful term; and

storage mechanism structured to store sets of grammatical relationships between a plurality of the determined meaningful terms based upon the determined grammatical role of each meaningful term relative to a meaningful term that is being used as a governing verb, wherein, for each meaningful term that is being used as a governing verb, the normalized data structure contains a set of meaningful terms that are subjects relative to the governing verb, a set of meaningful terms that are objects relative to the governing verb, and at least one of a set of meaningful terms that are verb modifiers of prepositional phrases that contain the governing verb and a set of meaningful terms that are noun modifiers of noun phrases that relate to the governing verb.

193. The system of claim 192, the storage mechanism further structured to store meaningful terms that correspond to a designated attribute.

194. The system of claim 193 wherein the designated attribute is at least one of country name, date, money, amount, number, location, person, corporate name, and organization.

195. The system of claim 192 wherein the sets of meaningful terms are stored as a plurality of tables.

196. The system of claim 195 wherein the tables comprise a subject table, an object table, a subject-object table, and at least one of a preposition table and a noun modifier table.

197. The system of claim 196 wherein the tables further comprise a sentence table that stores the text of the sentence.

198. The system of claim 196 wherein the tables further comprise an attributes table that stores meaningful terms that are associated with designated attributes.

199. The system of claim 195 wherein the preposition table contains for each meaningful term used as a verb in the sentence, a list of meaningful terms that are prepositions of the meaningful term used as the verb and at least one meaningful term that is a verb modifier associated with each preposition.

200. The system of claim 195 wherein the noun modifier table contains a list of meaningful terms that are noun modifiers of a meaningful terms that is used as a noun in the sentence.

201. The system of claim 192, further comprising a database in which the tables are stored.

202. The system of claim 192, further comprising a file system, wherein the tables are stored as part of the file system.

203. The system of claim 192 wherein a plurality of grammatical relationships between meaningful terms that are identical are stored in the normalized data structure.

204. The system of claim 192 wherein the document is part of a corpus of heterogeneous documents.

205. The system of claim 192 wherein the normalized data structure is used to index a corpus of documents.

206. The system of claim 192 wherein the enhanced data representation data structure is used to execute a query against objects in a corpus of documents.

207. A method in a computer system for transforming an object of a data set into a canonical representation for use in indexing the objects of the data set and in querying the data set, the object being other than a text-only document and having a plurality of units that are specified according to an object-specific grammar, comprising:

for each object,

decomposing the object to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful units of the object from these syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful unit; and

storing in an enhanced data representation data structure a representation of each meaningful unit associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

208. The method of claim 207 wherein the objects are audio data and the units of objects are portions of audio data.

209. The method of claim 207 wherein the objects are video data and the units of objects are portions of video data.

210. The method of claim 207 wherein the objects are images and the units of objects are graphical data.

211. The method of claim 207 wherein the data set is a document that contains text and graphical data and wherein each object is one of a text sentence and a sentence created to correspond to and describe a portion of graphical data.

212. A computer-readable memory medium containing instructions for controlling a computer processor to transform an object of a data set into a canonical representation for use in indexing the objects of the data set and in querying the data set, the object being other than a text-only

document and having a plurality of units that are specified according to an object-specific grammar, by:

for each object,

decomposing the object to generate a parse structure having a plurality of syntactic elements;

determining a set of meaningful units of the object from these syntactic elements;

determining from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful unit; and

storing in an enhanced data representation data structure a representation of each meaningful unit associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

213. The computer-readable memory medium of claim 212 wherein the objects are audio data and the units of objects are portions of audio data.

214. The computer-readable memory medium of claim 212 wherein the objects are video data and the units of objects are portions of video data.

215. The computer-readable memory medium of claim 212 wherein the objects are images and the units of objects are graphical data.

216. The computer-readable memory medium of claim 212 wherein the data set is a document that contains text and graphical data and wherein each object is one of a text sentence and a sentence created to correspond to and describe a portion of graphical data.

217. A query engine in a computer system for transforming an object of a data set into a canonical representation for use in indexing the objects of the data set and in querying the data set, the object being other than a text-only document and having a plurality of units that are specified according to an object-specific grammar, comprising:

decomposition processor that is structured to decompose each object to generate a parse structure having a plurality of syntactic elements; and

postprocessor that is structured to

receive from the decomposition processor the generated parse structure;

determine a set of meaningful units of the object from these syntactic elements;

determine from the structure of the parse structure and the syntactic elements a grammatical role for each meaningful unit; and

store in an enhanced data representation data structure a representation of each meaningful unit associated with its determined grammatical role, in a manner that indicates a grammatical relationship between a plurality of the meaningful units.

218. The query engine of claim 217 wherein the objects are audio data and the units of objects are portions of audio data.

219. The query engine of claim 217 wherein the objects are video data and the units of objects are portions of video data.

220. The query engine of claim 217 wherein the objects are images and the units of objects are graphical data.

221. The query engine of claim 217 wherein the data set is a document that contains text and graphical data and wherein each object is one of a text sentence and a sentence created to correspond to and describe a portion of graphical data.

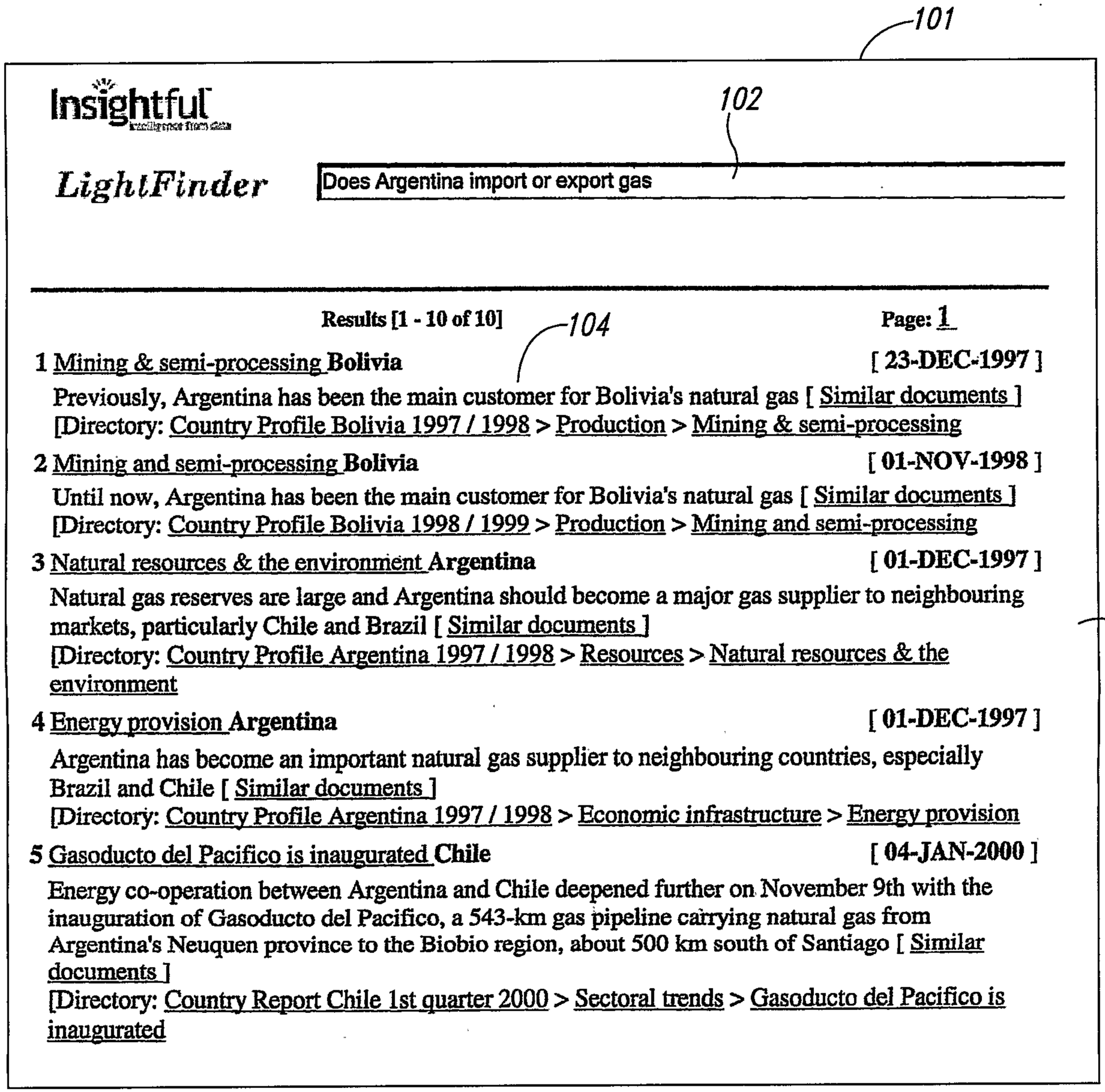

101
Insightful
intelligence from data
LightFinder
102
Does Argentina import or export gas
Results [1 - 10 of 10]
104
Page: 1
1 Mining & semi-processing Bolivia [23-DEC-1997]
Previously, Argentina has been the main customer for Bolivia's natural gas [Similar documents]
[Directory: Country Profile Bolivia 1997 / 1998 > Production > Mining & semi-processing
2 Mining and semi-processing Bolivia [01-NOV-1998]
Until now, Argentina has been the main customer for Bolivia's natural gas [Similar documents]
[Directory: Country Profile Bolivia 1998 / 1999 > Production > Mining and semi-processing
3 Natural resources & the environment Argentina [01-DEC-1997]
Natural gas reserves are large and Argentina should become a major gas supplier to neighbouring markets, particularly Chile and Brazil [Similar documents]
[Directory: Country Profile Argentina 1997 / 1998 > Resources > Natural resources & the environment
4 Energy provision Argentina [01-DEC-1997]
Argentina has become an important natural gas supplier to neighbouring countries, especially Brazil and Chile [Similar documents]
[Directory: Country Profile Argentina 1997 / 1998 > Economic infrastructure > Energy provision
5 Gasoducto del Pacifico is inaugurated Chile [04-JAN-2000]
Energy co-operation between Argentina and Chile deepened further on November 9th with the inauguration of Gasoducto del Pacifico, a 543-km gas pipeline carrying natural gas from Argentina's Neuquen province to the Biobio region, about 500 km south of Santiago [Similar documents]
[Directory: Country Report Chile 1st quarter 2000 > Sectoral trends > Gasoducto del Pacifico is inaugurated
103

Fig. 1

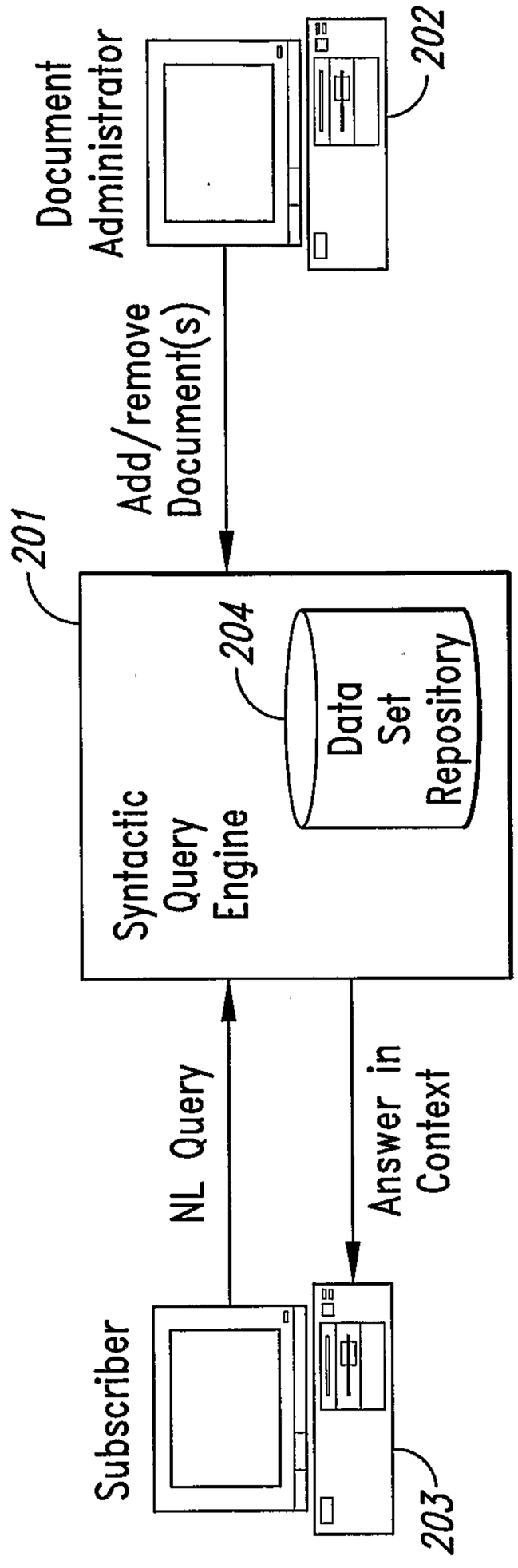

Document Administrator
202
Add/remove Document(s)
201
Syntactic Query Engine
204
Data Set Repository
NL Query
Answer in Context
Subscriber
203

Fig. 2

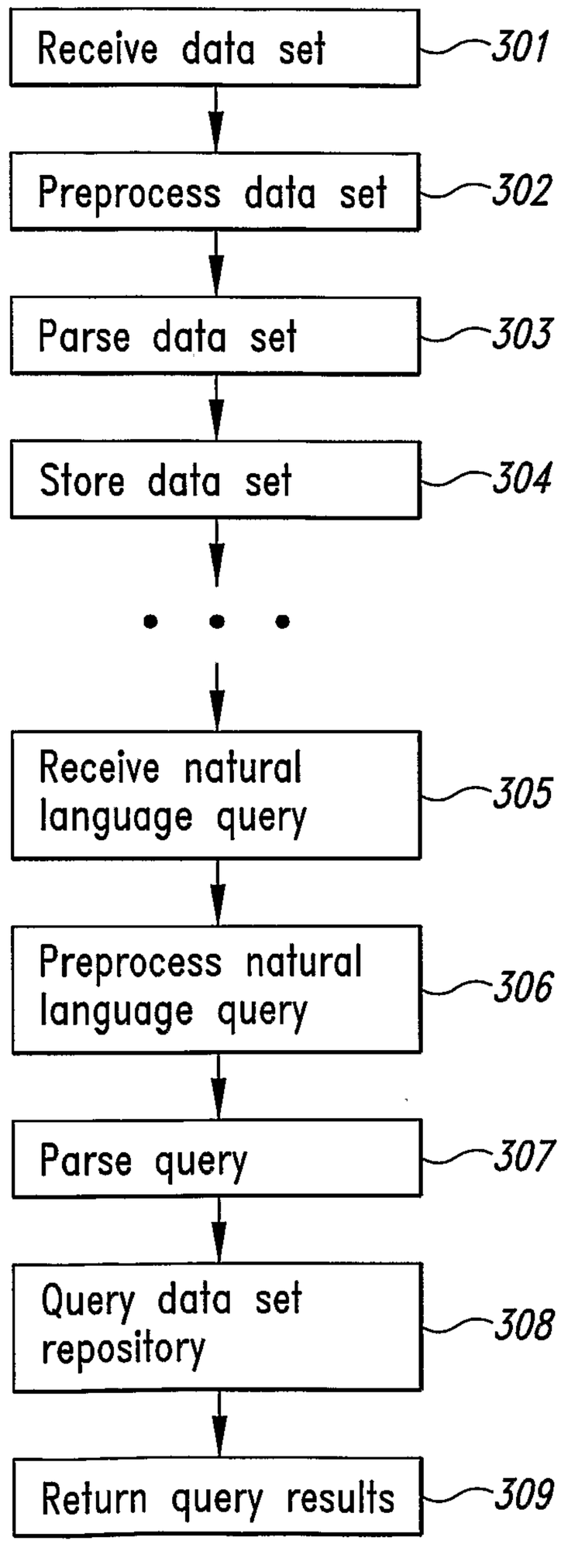

Receive data set
301
Preprocess data set
302
Parse data set
303
Store data set
304
Receive natural language query
305
Preprocess natural language query
306
Parse query
307
Query data set repository
308
Return query results
309

Fig. 3

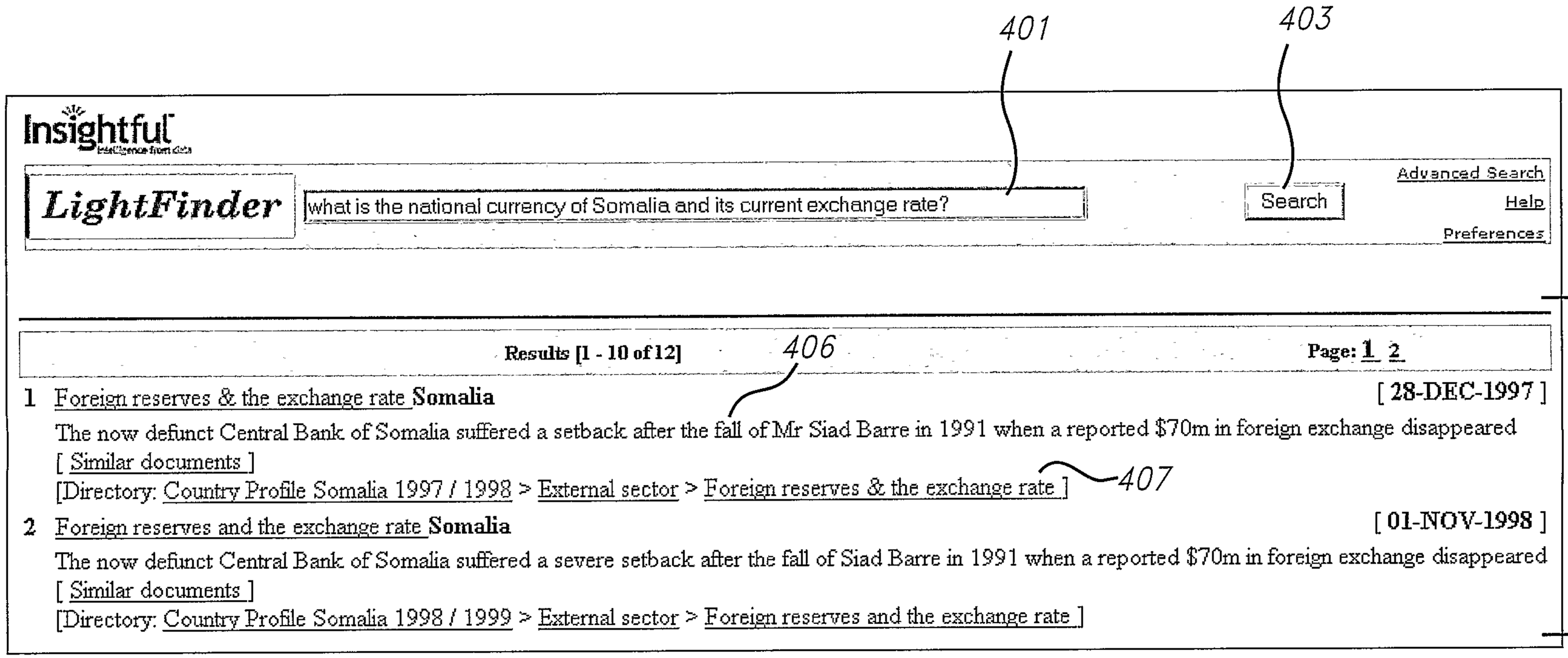

401
403
Insightful
Intelligence from data
LightFinder
what is the national currency of Somalia and its current exchange rate?
Search
Advanced Search
Help
Preferences
405
Results [1 - 10 of 12]
406
Page: 1 2
1 Foreign reserves & the exchange rate Somalia
[28-DEC-1997]
The now defunct Central Bank of Somalia suffered a setback after the fall of Mr Siad Barre in 1991 when a reported $70m in foreign exchange disappeared
[Similar documents]
[Directory: Country Profile Somalia 1997 / 1998 > External sector > Foreign reserves & the exchange rate]
407
2 Foreign reserves and the exchange rate Somalia
[01-NOV-1998]
The now defunct Central Bank of Somalia suffered a severe setback after the fall of Siad Barre in 1991 when a reported $70m in foreign exchange disappeared
[Similar documents]
[Directory: Country Profile Somalia 1998 / 1999 > External sector > Foreign reserves and the exchange rate]

Fig. 4

Insightful
Intelligence from data

LightFinder

5A04

Advanced Search
Help

Document: Foreign reserves & the exchange rate

Directory: Country Profile Somalia 1997 / 1998 > External sector > Foreign reserves & the exchange rate

Search Similar:

- Sentences — 5A01
- Paragraphs — 5A02
- Documents — 5A03

The now defunct Central Bank of Somalia suffered a setback after the fall of Mr Siad Barre in 1991 when a reported $70m in foreign exchange disappeared. There are no current data on foreign reserves.

Although the official exchange rate for the Somaliland shilling has stood at SolSh80:$1 since July 1995, its value on the parallel market has fallen. It was trading at SolSh5,000:$1 in the last months of 1996 but gained some ground to SolSh3,000:$1 through most of 1997 until a downturn to SolSh4,000:$1 in October 1997. In the south, the rate for the Somali shilling is variable from town to town and the situation is complicated by Mr Aideed's import of his own banknotes in June 1997 (see Economic policy above), but the rate in Mogadishu has fallen from SoSh3,000:$1 in 1993 to SoSh7,500:$1 in November 1997. (See Reference table 11 for historical data on average exchange rates.)

5A05

Fig. 5A

Insightful
intelligence from data

LightFinder

5B04

Advanced Search
Help

Document: Foreign reserves & the exchange rate

Directory: Country Profile Somalia 1997 / 1998 > External sector > Foreign reserves & the exchange rate

Search Similar: Sentences 5B01
Paragraphs
Documents

The now defunct Central Bank of Somalia suffered a setback after the fall of Mr Siad Barre in 1991 when a reported $70m in foreign exchange disappeared. There are no current data on foreign reserves.

Although the official exchange rate for the Somaliland shilling has stood at SolSh80:$1 since July 1995, its value on the parallel market has fallen. It was trading at SolSh5,000:$1 in the last months of 1996 but gained some ground to SolSh3,000:$1 through most of 1997 until a downturn to SolSh4,000:$1 in October 1997. In the south, the rate for the Somali shilling is variable from town to town and the situation is complicated by Mr Aideed's import of his own banknotes in June 1997 (see Economic policy above), but the rate in Mogadishu has fallen from SoSh3,000:$1 in 1993 to SoSh7,500:$1 in November 1997. (See Reference table 11 for historical data on average exchange rates.) 5B06

Fig. 5B

Insightful
Intelligence from data

LightFinder | In the south, the rate for the Somali shilling is variable from town to town and the situati | Search | Advanced Search | Help | Preferences

Results [1 - 10 of 13] **Page: 1 2**

1 Controversial new bank notes arrive -- **Somalia** **[05-JUL-1999]**

In fact, many of these traders openly support Mr Aideed's rivals, which may also account for the armed stand-offs, which are reported to have occurred at the airports upon the actual arrival of the bank-notes [Similar documents]
[Directory: Country Report Somalia 3rd quarter, 1999 > Economy > Controversial new bank notes arrive --]

2 Foreign reserves & the exchange rate **Somalia** **[28-DEC-1997]**

In the south, the rate for the Somali shilling is variable from town to town and the situation is complicated by Mr Aideed's import of his own banknotes in June 1997 (see Economic policy above), but the rate in Mogadishu has fallen from SoSh3,000:$1 in 1993 to SoSh7,500:$1 in November 1997 [Similar documents]
[Directory: Country Profile Somalia 1997 / 1998 > External sector > Foreign reserves & the exchange rate]

3 Controversial new bank notes arrive -- **Somalia** **[05-JUL-1999]**

Another delivery of new shillings was reportedly received by Mr Aideed at the Baidoa airport in late May, although a second delivery there was apparently cancelled following that city's fall to anti-Aideed forces in early June (see Political scene) [Similar documents]
[Directory: Country Report Somalia 3rd quarter, 1999 > Economy > Controversial new bank notes arrive --]

4 Economic outlook **Somalia** **[01-MAR-2001]**

However, until the government attains control of the country and is able to stop other economic agents shipping over banknotes, the prospect of the Central Bank --if it is re-established --performing a useful service during the forecast period is remote [Similar documents]
[Directory: Country Report Somalia March 2001 Main report > Outlook for 2001-02 > Economic outlook]

Fig. 5C

Insightful
intelligence from data

LightFinder Show me a map of recent conflict areas of Angola Search

Advanced Search
Help
Preferences

Results [1 - 5 of 5] **Page: 1**

1 New FAA offensive in central highlands successful -- **Angola** 608 **[19-NOV-1999]**

This map presents the Angola...(Please See Map/Chart) [Similar documents]
[Directory: Country Report Angola 4th quarter 1999 > Political scene > New FAA offensive in central highlands successful --]

2 First deepwater production is imminent **Angola** **[19-NOV-1999]**

This map presents the Angola...(Please See Map/Chart) [Similar documents]
[Directory: Country Report Angola 4th quarter 1999 > Oil & gas > First deepwater production is imminent]

3 Military situation is relatively quiet **Angola** **[01-JUN-2000]**

The area between Lucapa and Saurimo, which contains concessions held by Diamondworks and Southern Era, is also reported to have been unstable (see map, above) [Similar documents]

Fig. 6

The government's rejection of dialogue with UNITA, crackdown on legal opposition and intimidation of moderates in its own ranks provided a backdrop for a substantial military build-up. Military activity in northern Angola, followed by the widely trumpeted recapture of the southern town of Kuvango in early September, ultimately proved to be feints to distract UNITA from preparations for a major assault on its headquarters in the central highlands. A series of large FAA offensives, backed by heavy air attacks, began to unfold on September 14th, and the FAA enjoyed early successes. UNITA suffered heavy casualties when it tried unsuccessfully to defend Cangandala town against FAA forces moving southwards from Malange. They also lost the strategic Salazar bridge over the Cuanza river nearby, opening the way for FAA forces to move towards Mussende, the site of one of UNITA's most important airfields in the central highlands. At the same time, government forces based in Huambo and Kuito began to expand their perimeter of control around these towns, and simultaneously began to push northwards towards the symbolically important towns of Andulo and Bailundo, which had served as Mr Savimbi's headquarters since just before the signing of the Lusaka peace agreement in 1994. UNITA, which had significant conventional forces in the area, suffered from severe logistical problems, in particular a lack of fuel, which hampered its ability to move tanks and other heavy equipment.

Fig. 7

WO 03/017143
PCT/US02/25756
CA 02457693 2004-02-13

Insightful
intelligence from data

LightFinder | show me a chart of the GDP of China | Search | Advanced Search | Help | Preferences

Results [1 - 10 of 18]	Page: 1 2

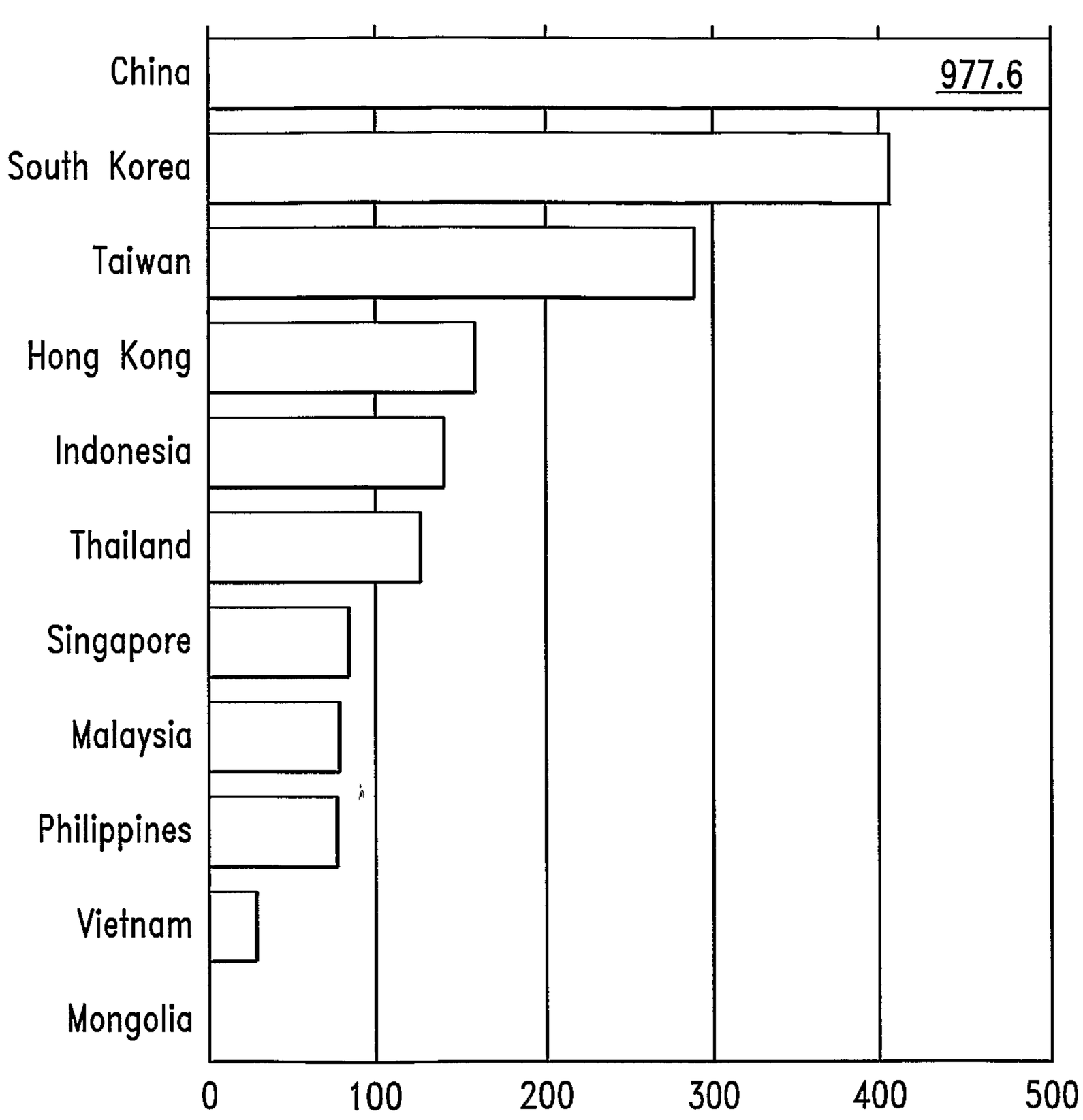
1 Related countries GDP **China** 808 [03-NOV-2000]

Related countries GDP of China is ... Please see the Table/Chart [Similar documents]
[Directory: Related countries GDP >

2 Related countries GDP per head **China** [03-NOV-2000]

Related countries GDP per head of China is ... Please see the Table/Chart [Similar documents]
[Directory: Related countries GDP per head >

Fig. 8



(a) Less than US$1bn.
Sources: EIU estimates; national sources.

Fig. 9

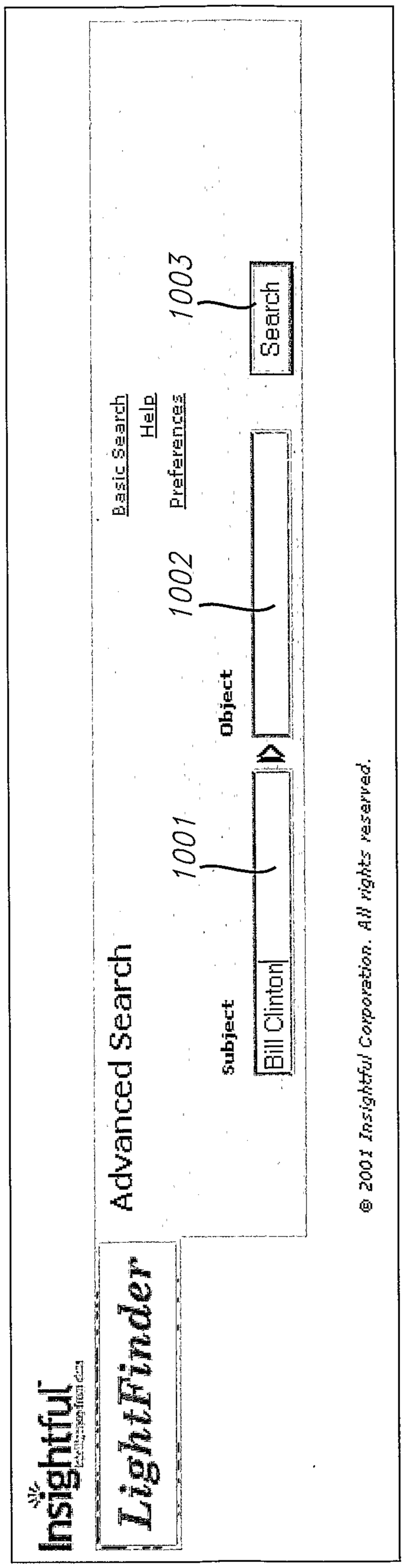

Insightful
LightFinder
Advanced Search
Subject
Bill Clinton
1001
Object
1002
Basic Search
Help
Preferences
1003
Search
© 2001 Insightful Corporation. All rights reserved.

Fig. 10

1101

Relationship: BILL CLINTON

be[33]	sign[19]	visit[16]	make[12]	say[9]
renew[6]	agree[4]	announce[4]	approve[4]	fly[4]
issue[4]	lead[4]	pay[4]	seek[4]	apologise[3]
appear[3]	ask[3]	call on[3]	confirm[3]	force[3]
form[3]	go[3]	have[3]	hold[3]	lead to[3]
mark[3]	meet[3]	propose[3]	reward[3]	take[3]
appoint[2]	block[2]	choose[2]	continue[2]	decide[2]
end[2]	follow[2]	give[2]	halt[2]	hope[2]
indicate[2]	invite[2]	praise[2]	raise[2]	recognise[2]
refer to[2]	reiterate[2]	respond[2]	see[2]	tell[2]
thank[2]	urge[2]	veto[2]	want[2]	aid[1]
allege[1]	appeal to[1]	apply[1]	arrive[1]	associate[1]
attend[1]	authorise[1]	award[1]	back[1]	become[1]
begin[1]	believe[1]	bin[1]	blame[1]	bolster[1]
break[1]	bring about[1]	call for[1]	cap[1]	cite[1]
claim[1]	come out[1]	commit[1]	convince[1]	declare[1]
decline[1]	delay[1]	deny[1]	devote[1]	dispatch[1]
divert[1]	draw[1]	echo[1]	embark[1]	emphasise[1]
encourage[1]	enhance[1]	enter[1]	express[1]	extend[1]
fail[1]	fight[1]	gain[1]	help[1]	hint[1]

Fig. 11

1200

Insightful
intelligence from data

LightFinder bill clinton visit Search

Advanced Search
Help
Preferences

Results [1 - 10 of 16] **Page: 1 2** 1205

1 President Clinton visits Guatemala -- **Guatemala** **[23-APR-1999]**

On March 10th and 11th the US president, Bill Clinton, visited Guatemala, as part of his four-day visit to Central America [Similar documents]
[Directory: Country Report Guatemala 2nd quarter, 1999 > Political scene > President Clinton visits Guatemala --]

2 Pres Clinton apologises for not stopping genocide -- **Rwanda** **[09-MAY-1998]**

President Bill Clinton of the US visited Rwanda for three hours on March 25th during his African tour [Similar documents]
[Directory: Country Report Rwanda 2nd Quarter 1998 > Political scene > Pres Clinton apologises for not stopping genocide --]

3 International relations and defence **Ghana** **[08-JUN-2000]**

The PNDC's socialist contacts caused tension with the US, but since the beginning of the 1990s relations have improved dramatically and the US president, Bill Clinton, visited Ghana in March 1998 on the first leg of his six-country tour of Africa [Similar documents]
[Directory: Country Profile Ghana 2000/2001 > Political background > International relations and defence]

4 International relations and defence **Ghana** **[12-JUL-1999]**

The PNDC's socialist contacts caused tension with the US, but since the beginning of the 1990s relations have improved dramatically and the US president, Bill Clinton, visited Ghana in March 1998, on the first leg of his six-country tour of Africa [Similar documents]
[Directory: Country Profile Ghana 1999/2000 > Political background > International relations and defence]

5 At a glance--2001-02--North Korea November 2000 **North Korea** **[09-NOV-2000]**

The US president, Bill Clinton, may visit North Korea in return for a missile deal [Similar documents]
[Directory: Country Report North Korea November 2000 Main report > At a glance--2001-02--North Korea November 2000

6 US troops may stay longer **Bosnia and Hercegovina** **[30-JAN-1998]**

The US president, Bill Clinton, visited American troops in Bosnia in December [Similar documents]

Fig. 12

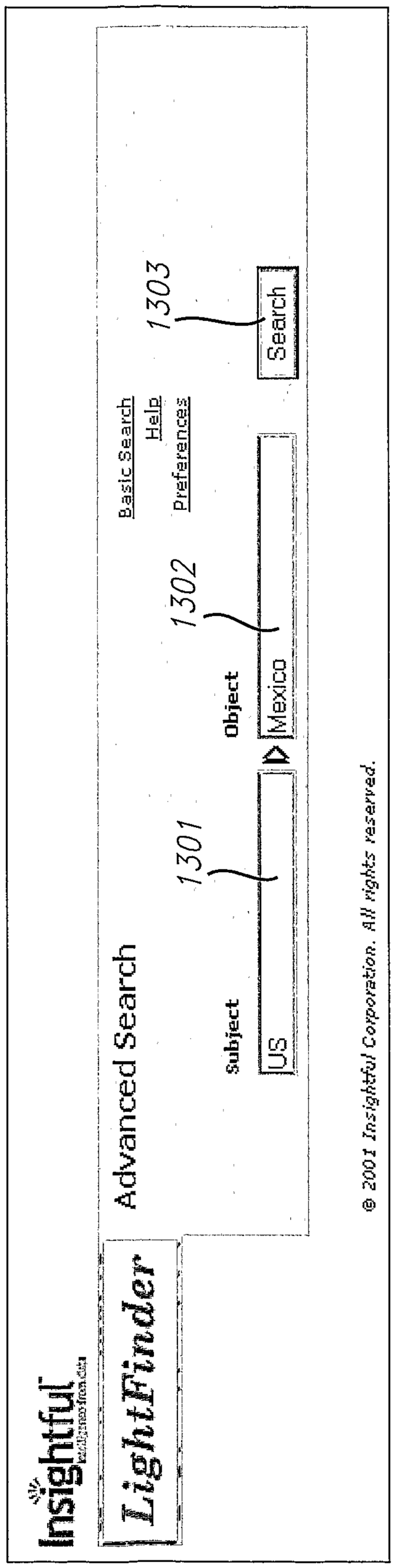

Insightful
LightFinder
Advanced Search
Subject
US
1301
Object
Mexico
1302
Basic Search
Help
Preferances
Search
1303
© 2001 Insightful Corporation. All rights reserved.

Fig. 13

1406 1405

Relationship: US ▷ MEXICO

accord[3]	account for[1]	accuse[3]	approve[1]	be[31]
benefit[1]	certify[5]	define[3]	dismiss[1]	ensure[1]
favour[1]	generate[2]	grant[1]	have[1]	hit[1]
improve[2]	initiate[1]	issue[1]	market[1]	reserve[1]
retain[1]	see[1]	sign[1]	speed[3]	state[1]
step up[1]	test[1]	visit[1]		

1407

Reverse Relationship: MEXICO ▷ US

0 >

be[12]	become[4]	bid[1]	close[1]	come from[5]
complement[1]	compromise[1]	deepen[1]	displace[1]	dump[1]
encounter[2]	enjoy[2]	export[1]	fear[1]	have[10]
impose[1]	improve[1]	increase[2]	lead to[1]	make[1]
pay[1]	privilege[2]	rise[2]	run into[5]	sign[1]
suffer[1]	take[1]	tariff[6]	test[1]	widen[1]

Fig. 14

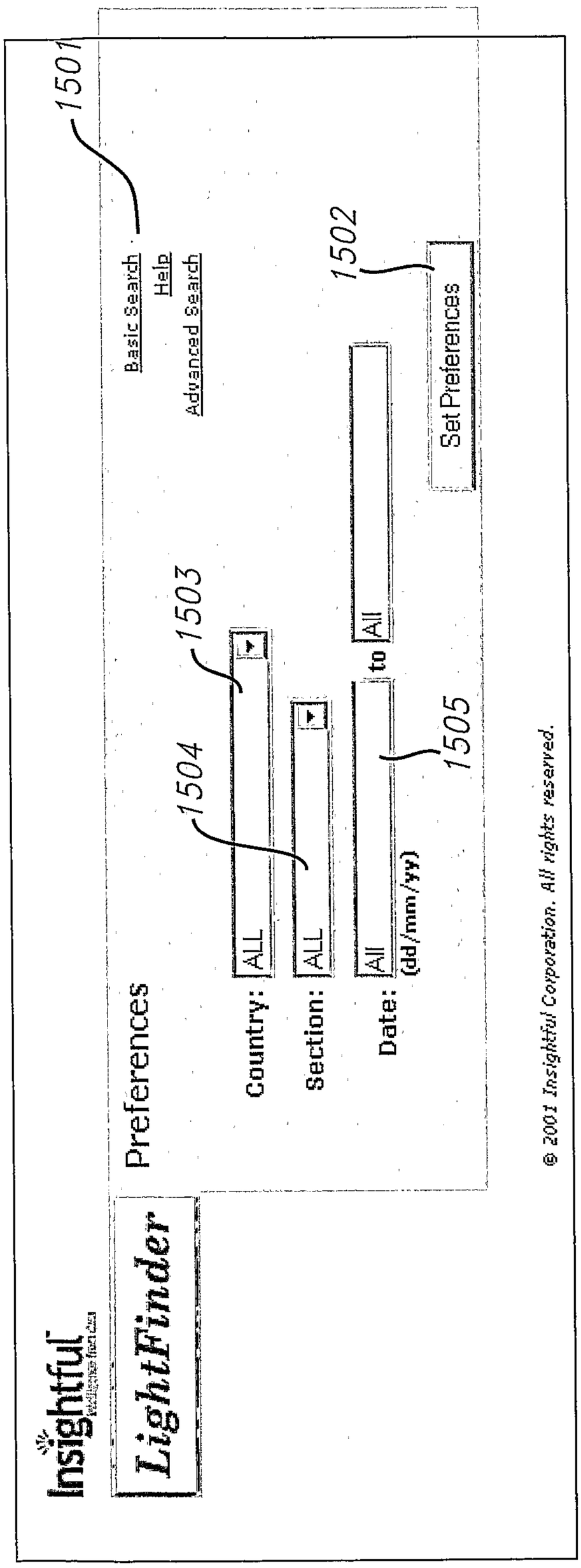

1501
Basic Search
Help
Advanced Search
1503
1502
to
All
Set Preferences
Preferences
1504
Country:
ALL
Section:
ALL
Date:
All
(dd/mm/yy)
to
1505
© 2001 Insightful Corporation. All rights reserved.
Insightful
LightFinder

Fig. 15

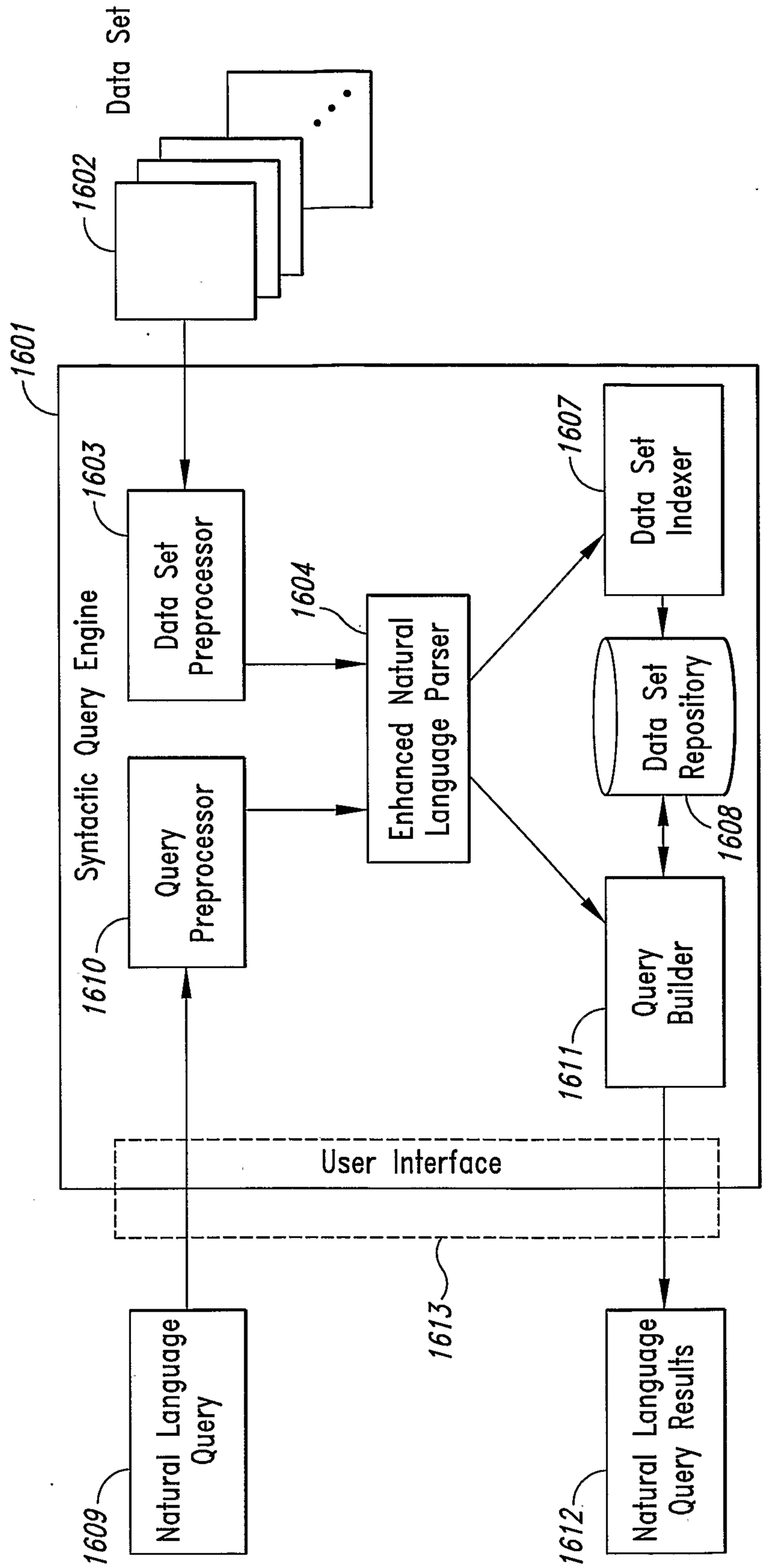

Data Set
1602
1601
Syntactic Query Engine
1603
Data Set Preprocessor
1604
Enhanced Natural Language Parser
1607
Data Set Indexer
Data Set Repository
1608
Query Preprocessor
1610
Query Builder
1611
User Interface
1609
Natural Language Query
1613
1612
Natural Language Query Results

Fig. 16

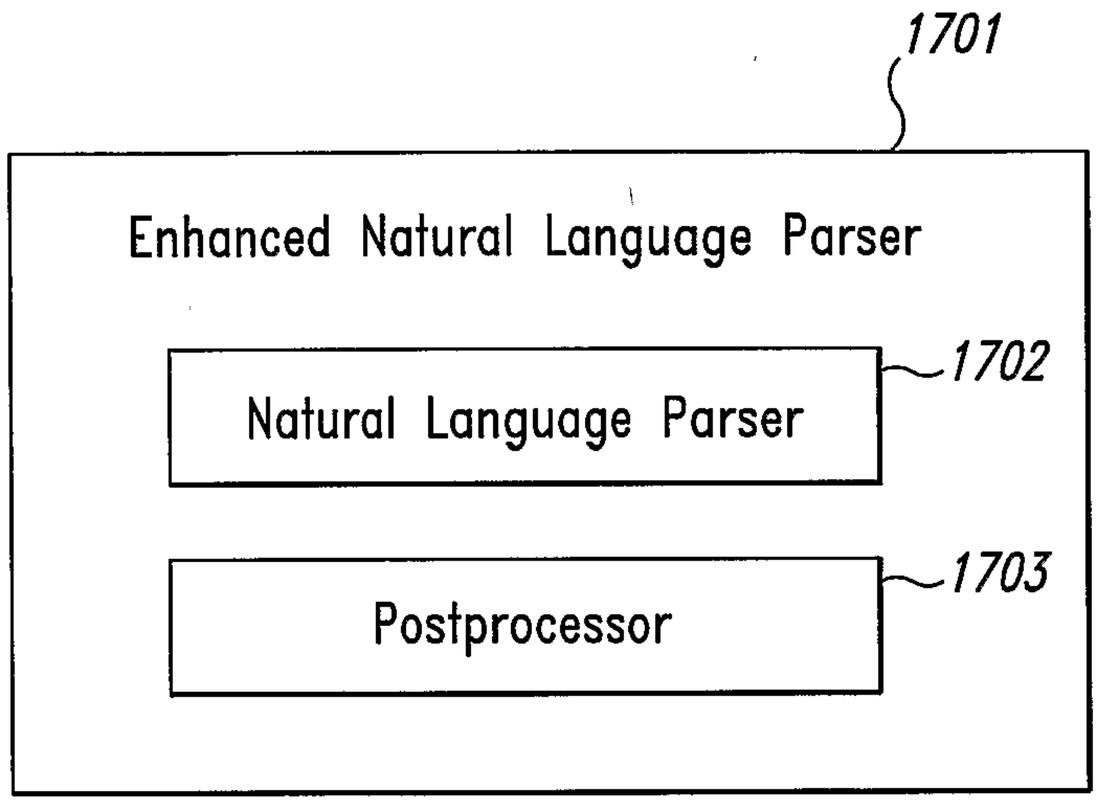

1701
Enhanced Natural Language Parser
1702
Natural Language Parser
1703
Postprocessor

Fig. 17

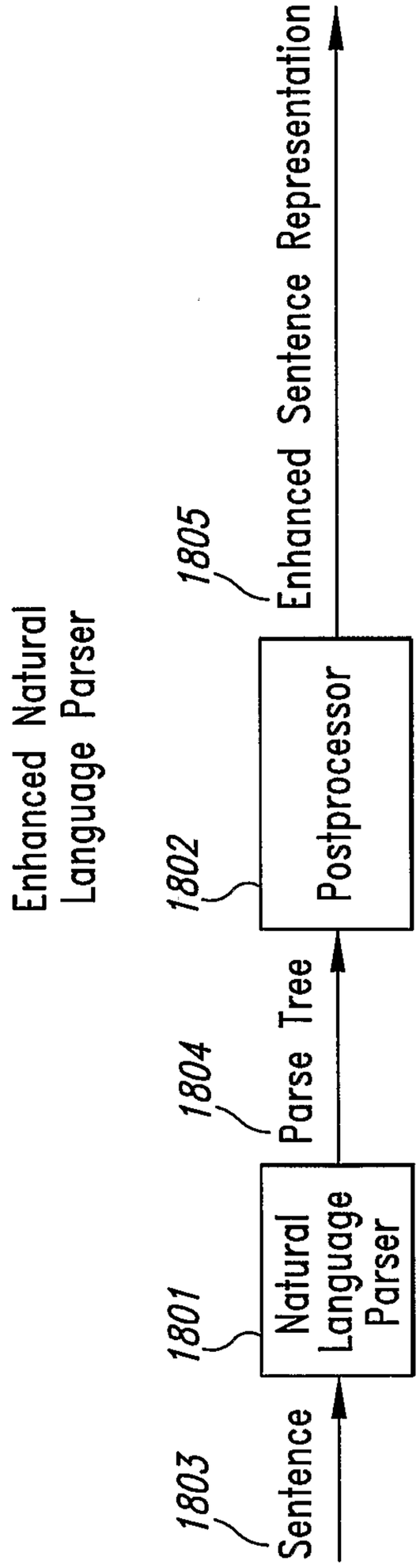

Enhanced Natural Language Parser
1803
Sentence
1801
Natural Language Parser
1804
Parse Tree
1802
Postprocessor
1805
Enhanced Sentence Representation

Fig. 18

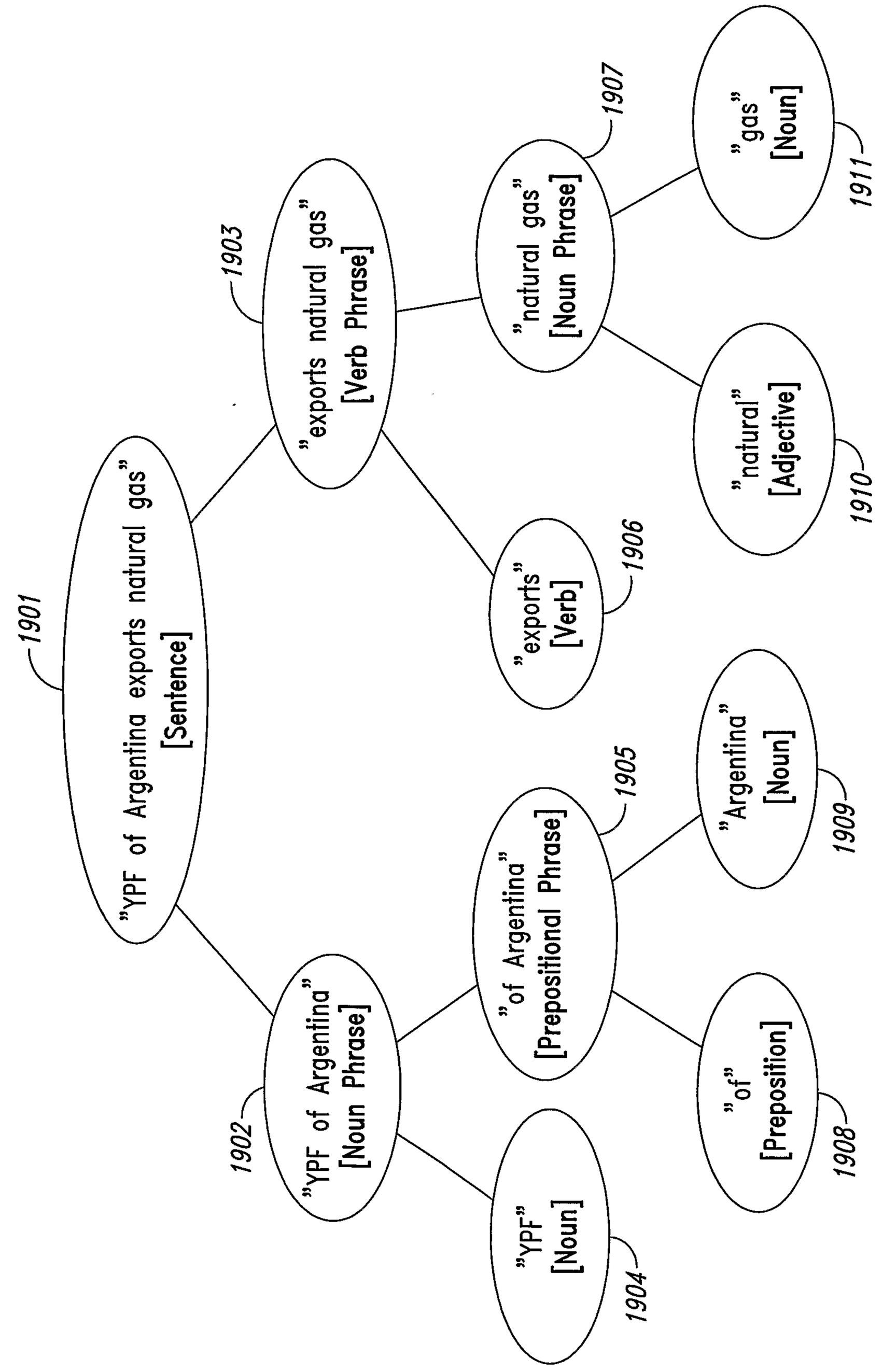

"YPF of Argentina exports natural gas"
[Sentence]
1901
"YPF of Argentina"
[Noun Phrase]
1902
"YPF"
[Noun]
1904
"exports natural gas"
[Verb Phrase]
1903
"exports"
[Verb]
1906
"of Argentina"
[Prepositional Phrase]
1905
"of"
[Preposition]
1908
"Argentina"
[Noun]
1909
"natural gas"
[Noun Phrase]
1907
"natural"
[Adjective]
1910
"gas"
[Noun]
1911

Fig. 19

	Subject	Verb	Object	Preposition	Verb Modifier	Noun Modifier
2001	Argentina					YPF
2002	YPF		natural gas			
2003	YPF		gas			
2004	Argentina		natural gas			
2005	Argentina		gas			
2006	YPF	export				
2007	Argentina	export				
2008		export	natural gas			
2009		export	gas			

Fig. 20

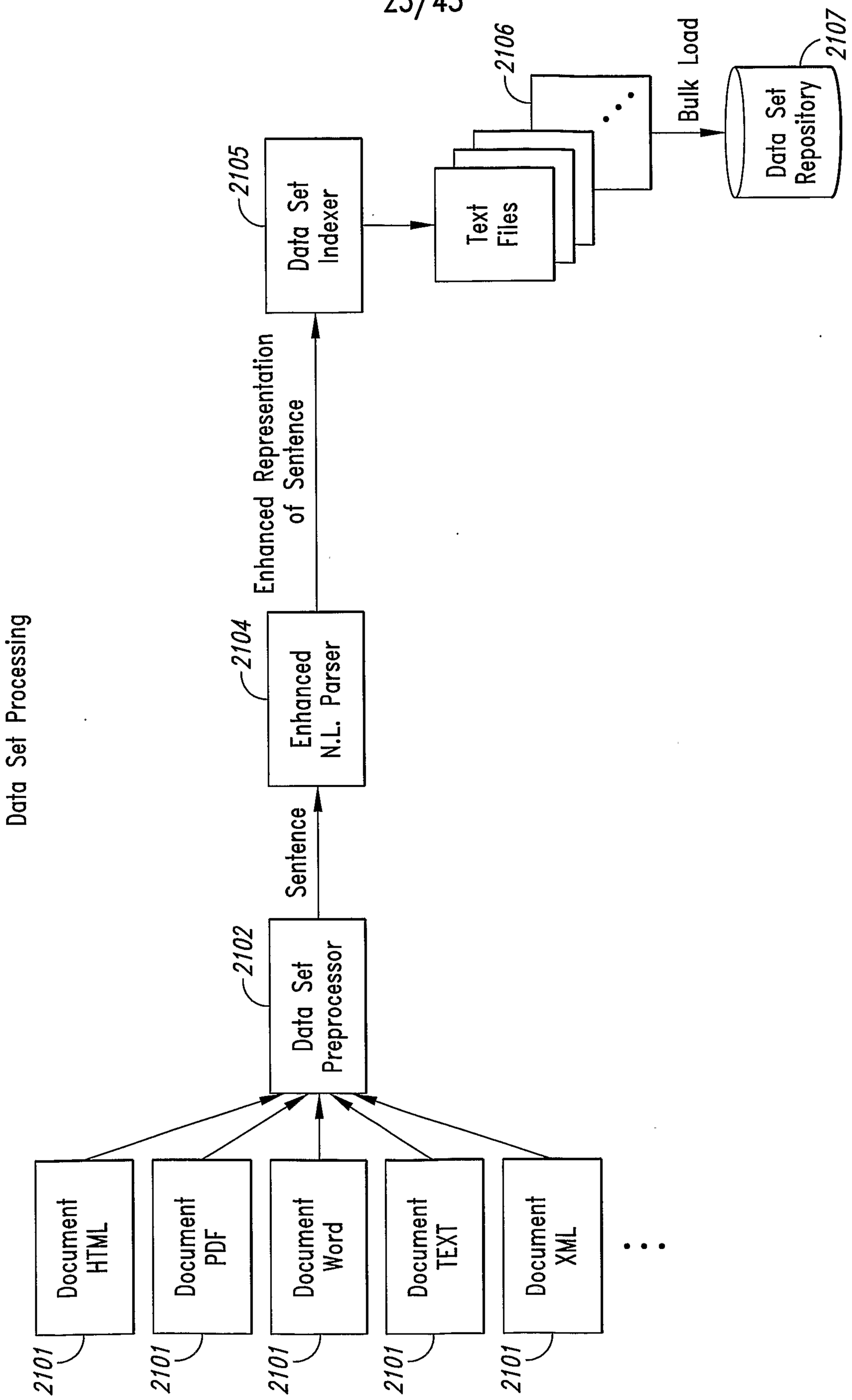

Data Set Processing
2101
Document HTML
Document PDF
Document Word
Document TEXT
Document XML
2102
Data Set Preprocessor
Sentence
2104
Enhanced N.L. Parser
Enhanced Representation of Sentence
2105
Data Set Indexer
2106
Text Files
Bulk Load
2107
Data Set Repository

Fig. 21

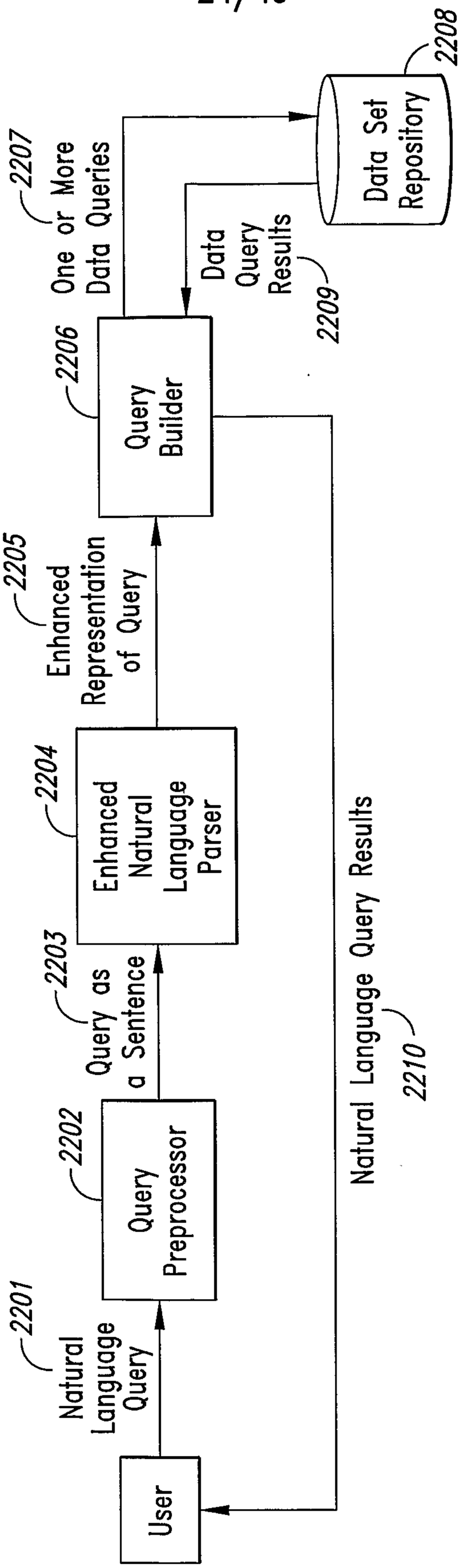

Natural Language Query Processing
2201
Natural Language Query
2202
Query Preprocessor
2203
Query as a Sentence
2204
Enhanced Natural Language Parser
2205
Enhanced Representation of Query
2206
Query Builder
2207
One or More Data Queries
2208
Data Set Repository
2209
Data Query Results
2210
Natural Language Query Results
User

Fig. 22

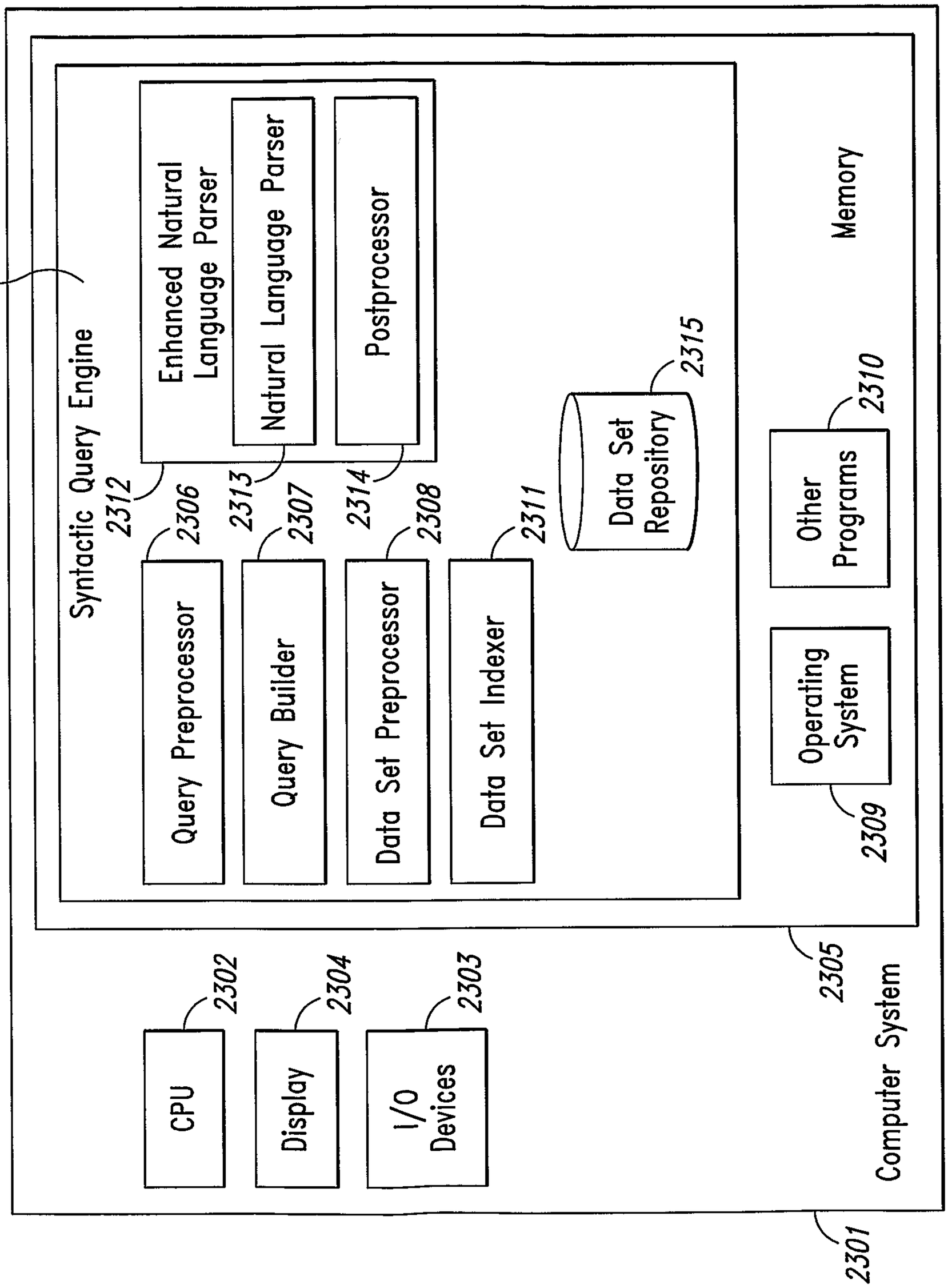

2301
Computer System
2305
Memory
2320
Syntactic Query Engine
Enhanced Natural Language Parser
Natural Language Parser
Postprocessor
Query Preprocessor
Query Builder
Data Set Preprocessor
Data Set Indexer
2312
2313
2314
2306
2307
2308
2311
Data Set Repository
2315
Other Programs
2310
Operating System
2309
CPU
2302
Display
2304
I/O Devices
2303

Fig. 23

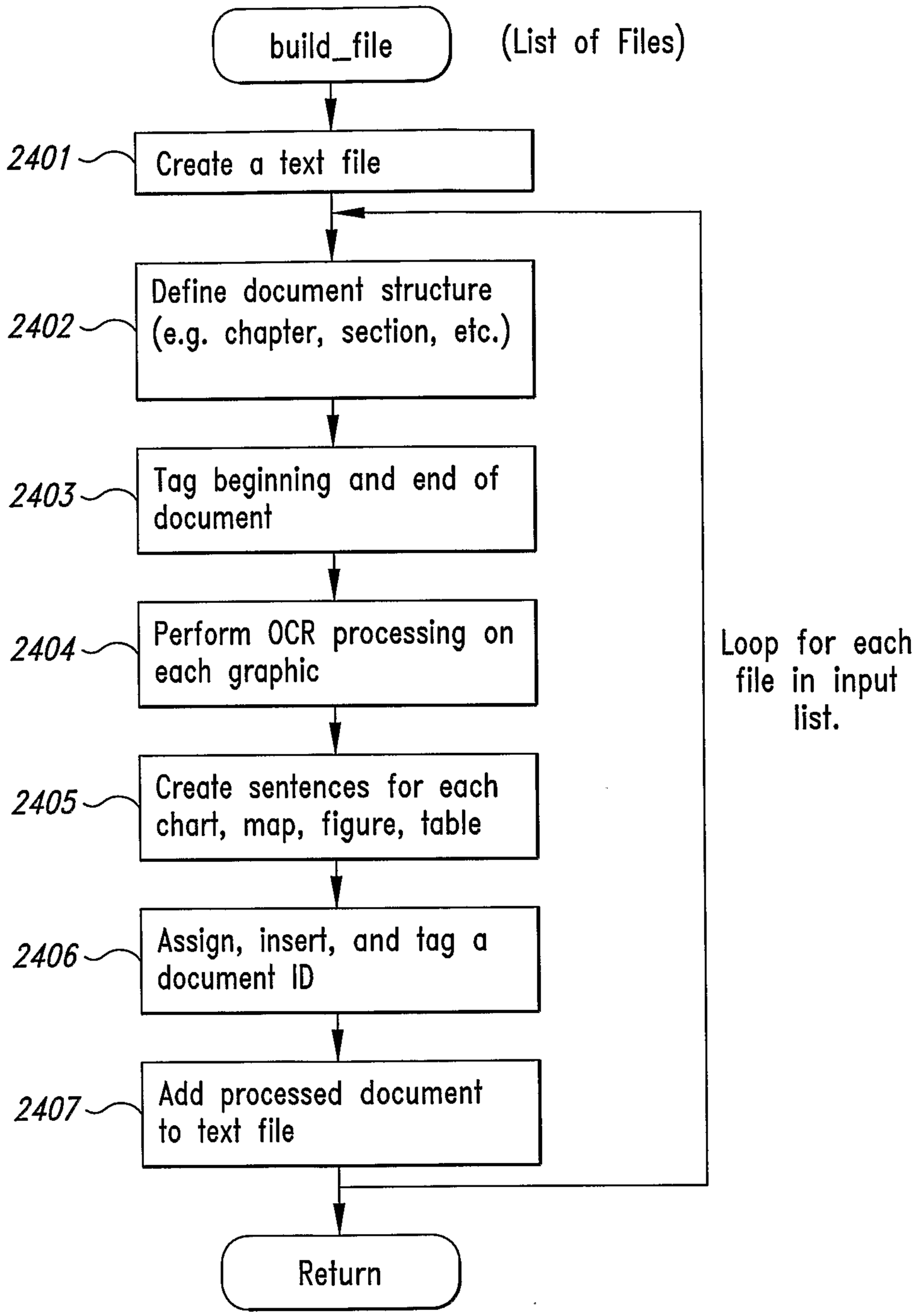

build_file
(List of Files)
2401
Create a text file
2402
Define document structure (e.g. chapter, section, etc.)
2403
Tag beginning and end of document
2404
Perform OCR processing on each graphic
2405
Create sentences for each chart, map, figure, table
2406
Assign, insert, and tag a document ID
2407
Add processed document to text file
Loop for each file in input list.
Return

Fig. 24

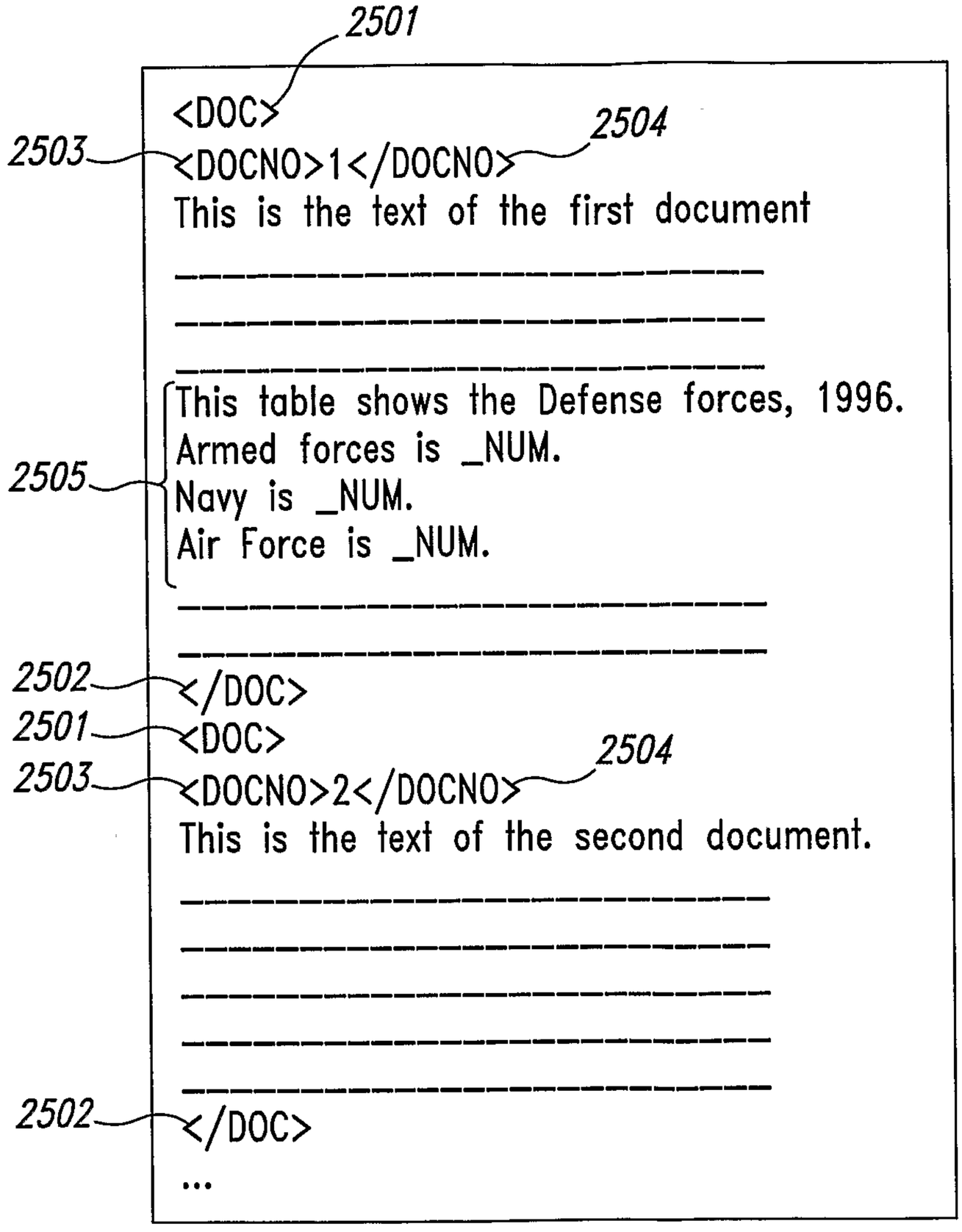

2501
<DOC>
2503
<DOCNO>1</DOCNO>
2504
This is the text of the first document
This table shows the Defense forces, 1996.
Armed forces is _NUM.
2505
Navy is _NUM.
Air Force is _NUM.
2502
</DOC>
2501
<DOC>
2503
<DOCNO>2</DOCNO>
2504
This is the text of the second document.
2502
</DOC>
...

Fig. 25

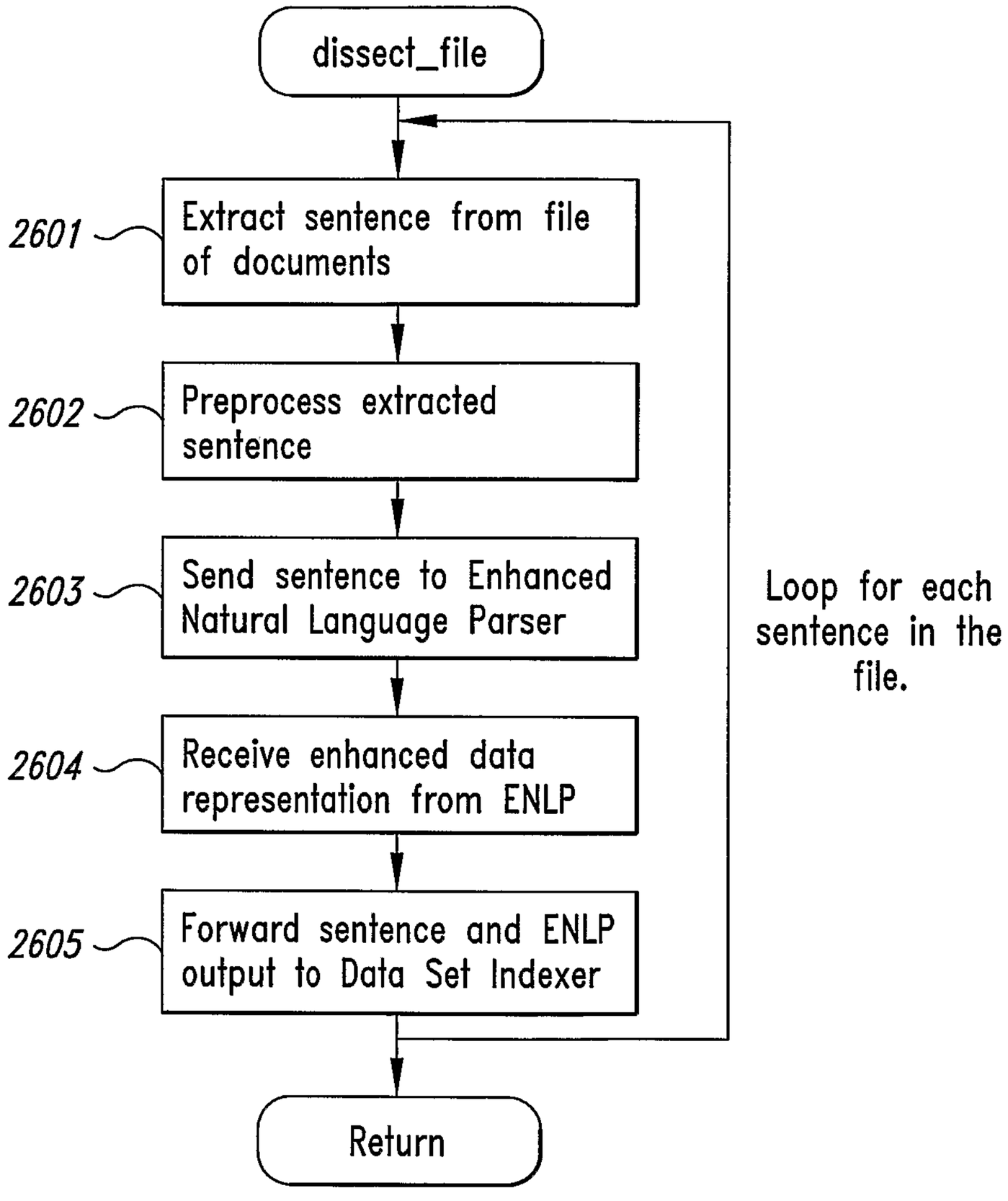

dissect_file
2601
Extract sentence from file of documents
2602
Preprocess extracted sentence
2603
Send sentence to Enhanced Natural Language Parser
2604
Receive enhanced data representation from ENLP
2605
Forward sentence and ENLP output to Data Set Indexer
Return
Loop for each sentence in the file.

Fig. 26

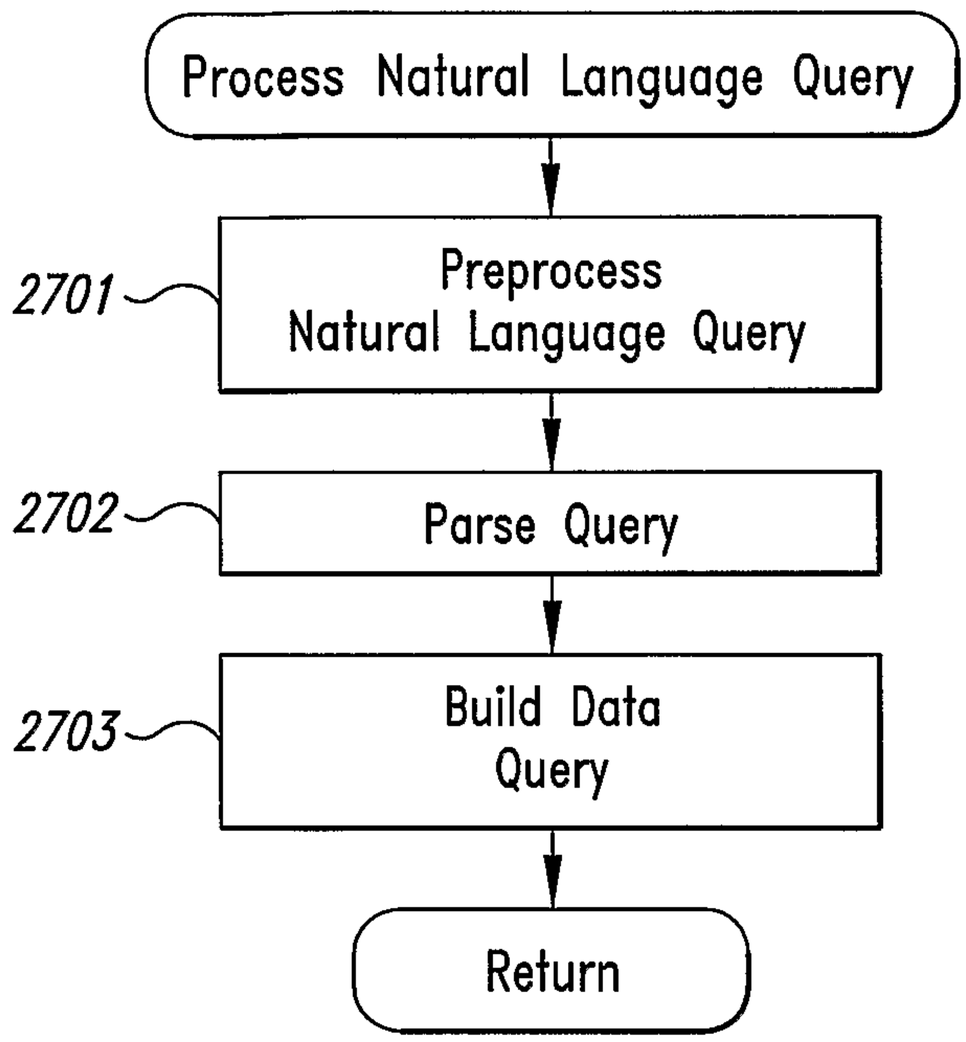

Process Natural Language Query
2701
Preprocess Natural Language Query
2702
Parse Query
2703
Build Data Query
Return

Fig. 27

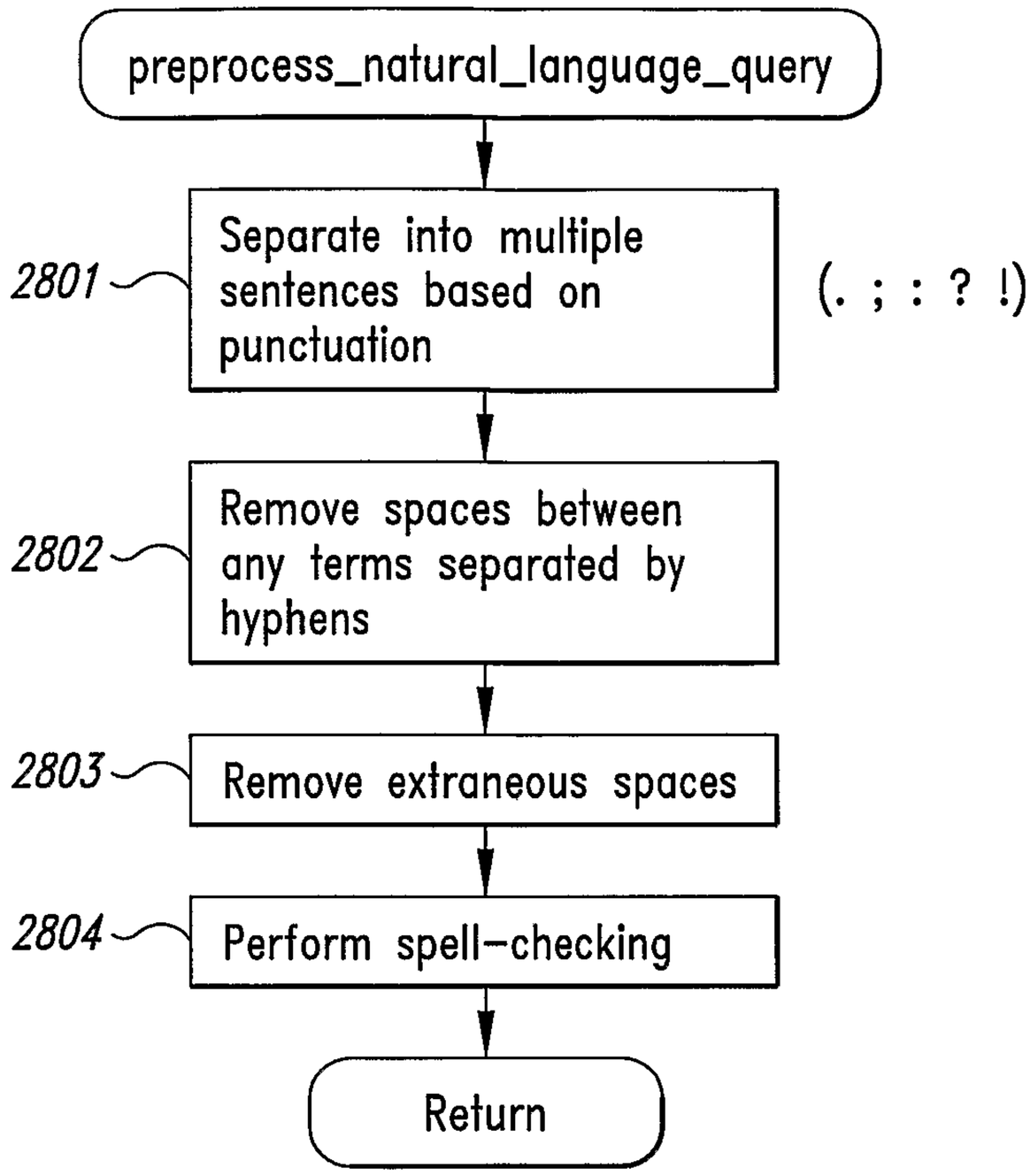

preprocess_natural_language_query
2801
Separate into multiple sentences based on punctuation
(. ; : ? !)
2802
Remove spaces between any terms separated by hyphens
2803
Remove extraneous spaces
2804
Perform spell-checking
Return

Fig. 28

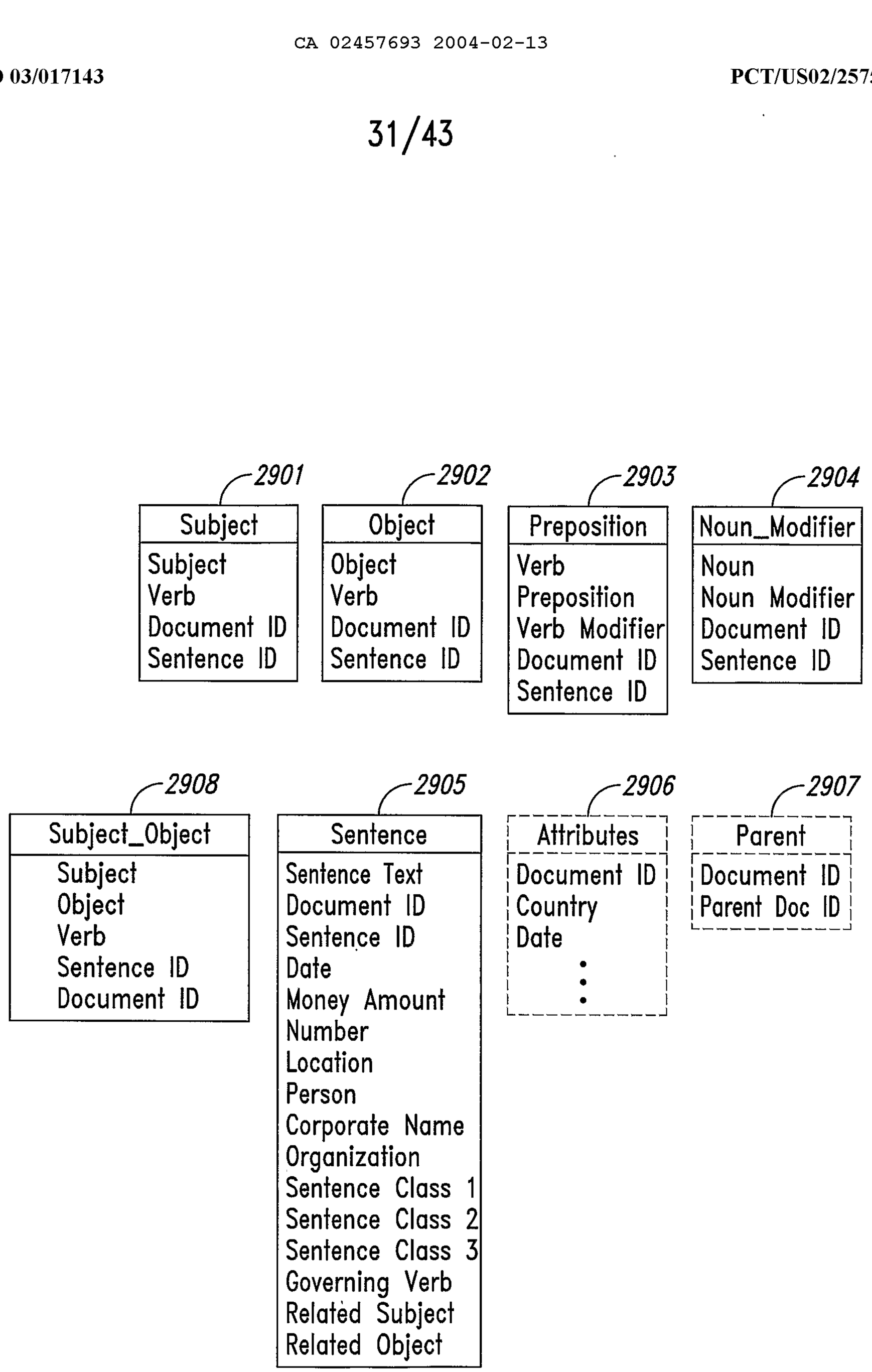

2901
Subject
Subject
Verb
Document ID
Sentence ID
2902
Object
Object
Verb
Document ID
Sentence ID
2903
Preposition
Verb
Preposition
Verb Modifier
Document ID
Sentence ID
2904
Noun_Modifier
Noun
Noun Modifier
Document ID
Sentence ID
2908
Subject_Object
Subject
Object
Verb
Sentence ID
Document ID
2905
Sentence
Sentence Text
Document ID
Sentence ID
Date
Money Amount
Number
Location
Person
Corporate Name
Organization
Sentence Class 1
Sentence Class 2
Sentence Class 3
Governing Verb
Related Subject
Related Object
2906
Attributes
Document ID
Country
Date
2907
Parent
Document ID
Parent Doc ID

Fig. 29

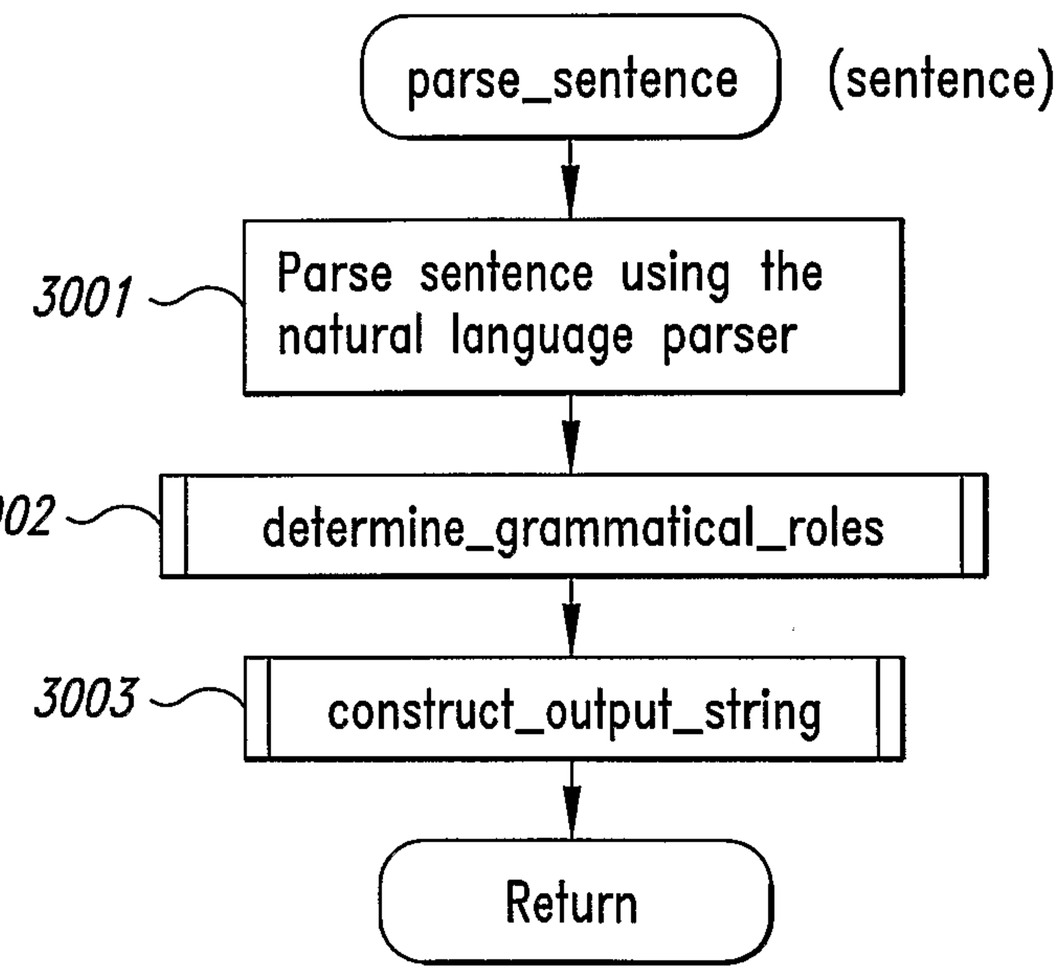

parse_sentence
(sentence)
3001
Parse sentence using the natural language parser
3002
determine_grammatical_roles
3003
construct_output_string
Return

Fig. 30

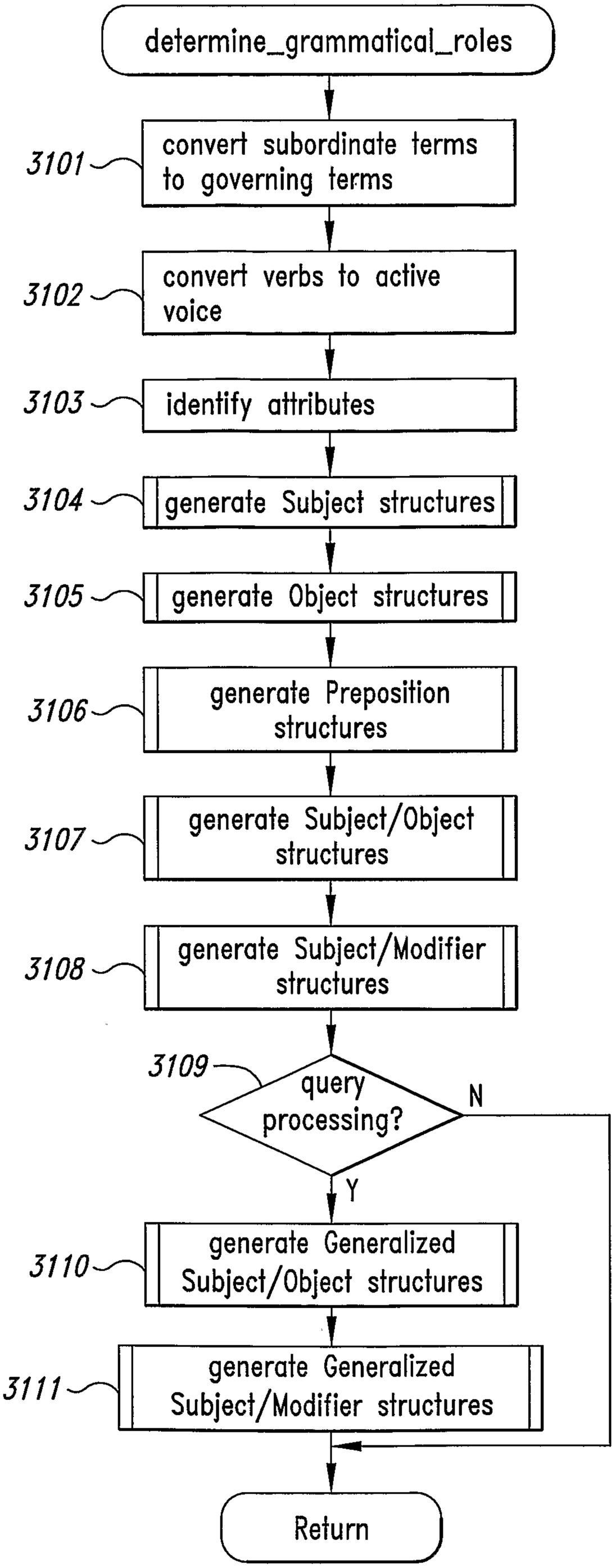

determine_grammatical_roles
3101
convert subordinate terms to governing terms
3102
convert verbs to active voice
3103
identify attributes
3104
generate Subject structures
3105
generate Object structures
3106
generate Preposition structures
3107
generate Subject/Object structures
3108
generate Subject/Modifier structures
3109
query processing?
N
Y
3110
generate Generalized Subject/Object structures
3111
generate Generalized Subject/Modifier structures
Return

Fig. 31

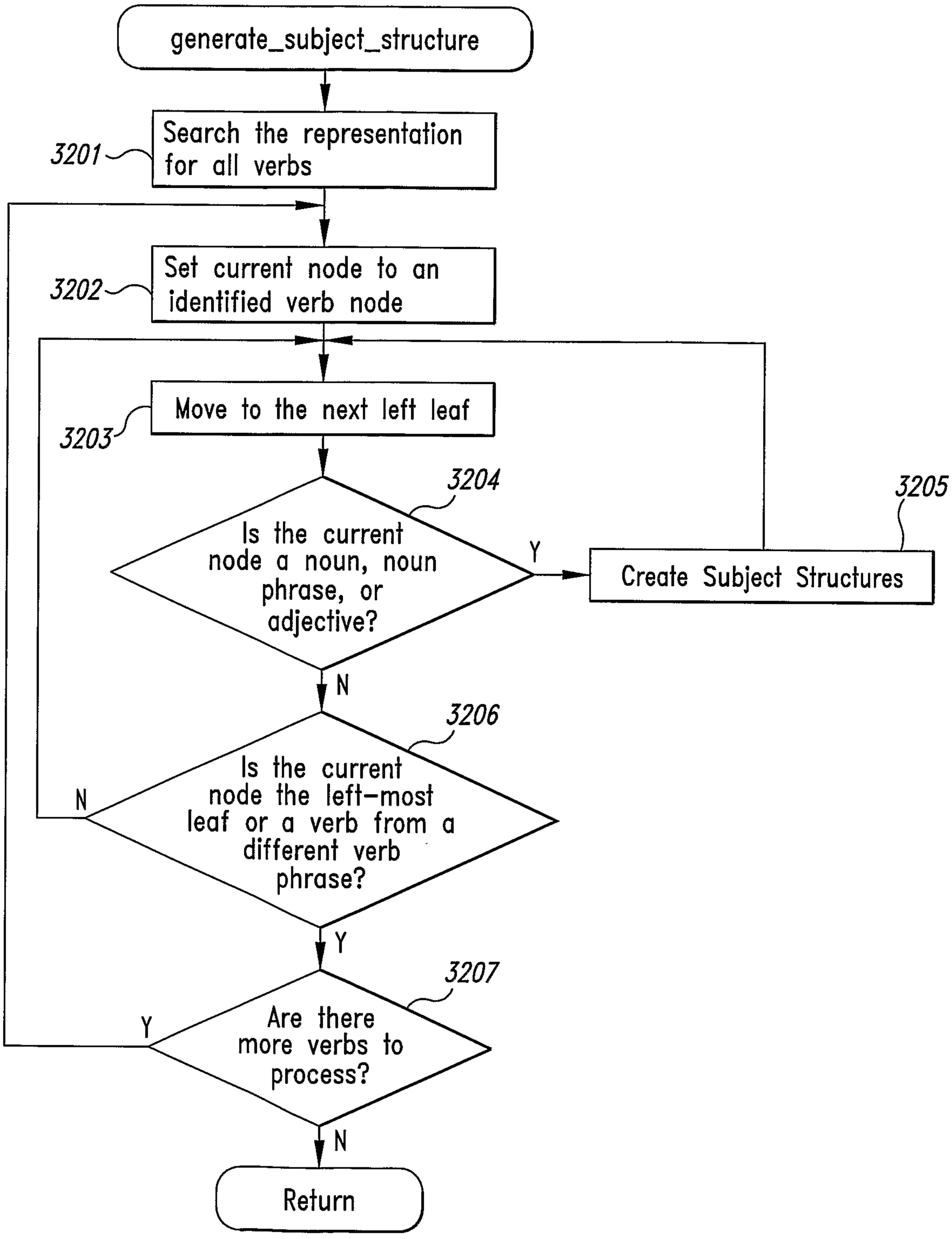

generate_subject_structure
Search the representation for all verbs
3201
Set current node to an identified verb node
3202
Move to the next left leaf
3203
Is the current node a noun, noun phrase, or adjective?
3204
Y
Create Subject Structures
3205
N
Is the current node the left-most leaf or a verb from a different verb phrase?
3206
N
Y
Are there more verbs to process?
3207
Y
N
Return

Fig. 32

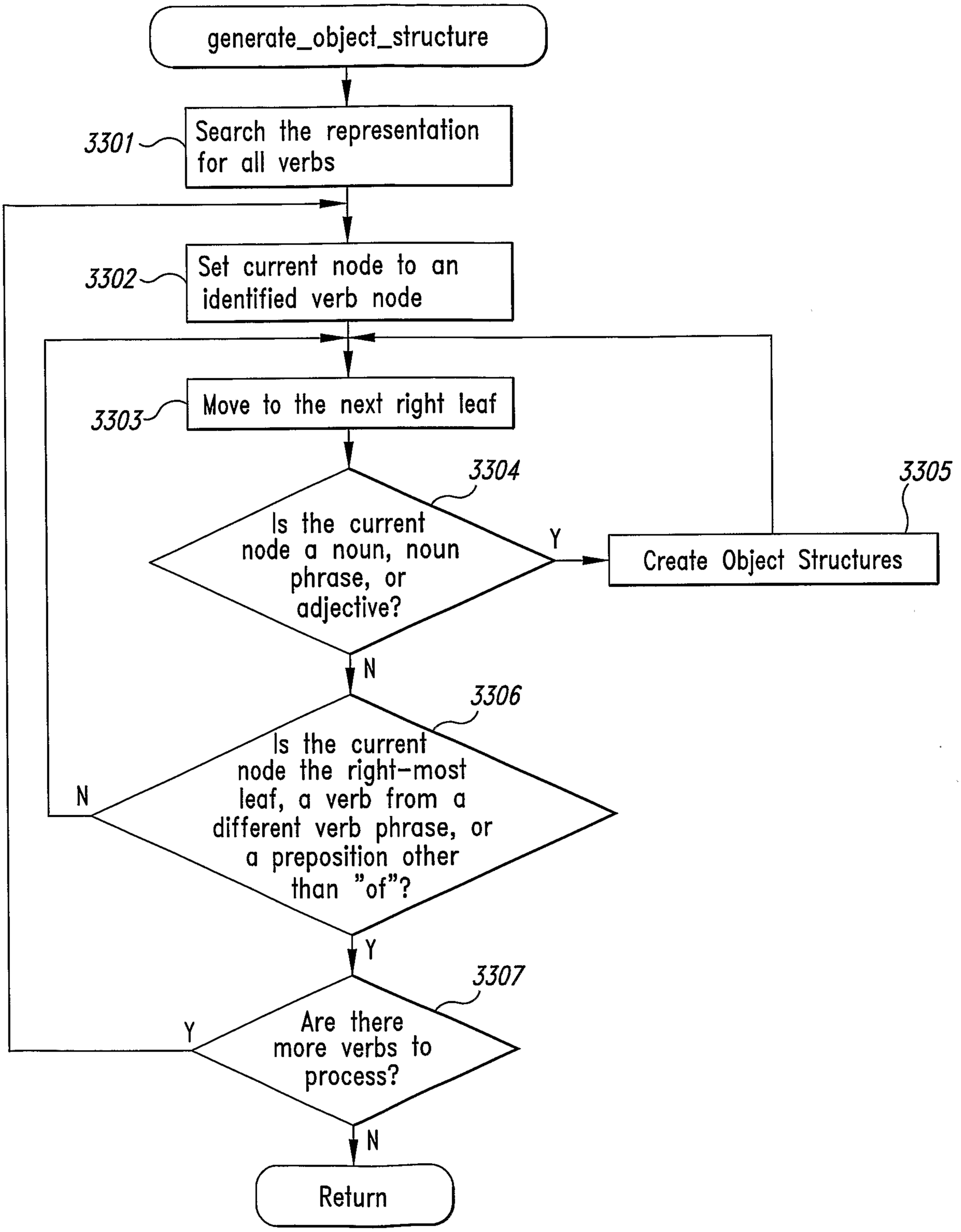

generate_object_structure
3301
Search the representation for all verbs
3302
Set current node to an identified verb node
3303
Move to the next right leaf
3304
Is the current node a noun, noun phrase, or adjective?
Y
3305
Create Object Structures
N
3306
Is the current node the right-most leaf, a verb from a different verb phrase, or a preposition other than "of"?
N
Y
3307
Are there more verbs to process?
Y
N
Return

Fig. 33

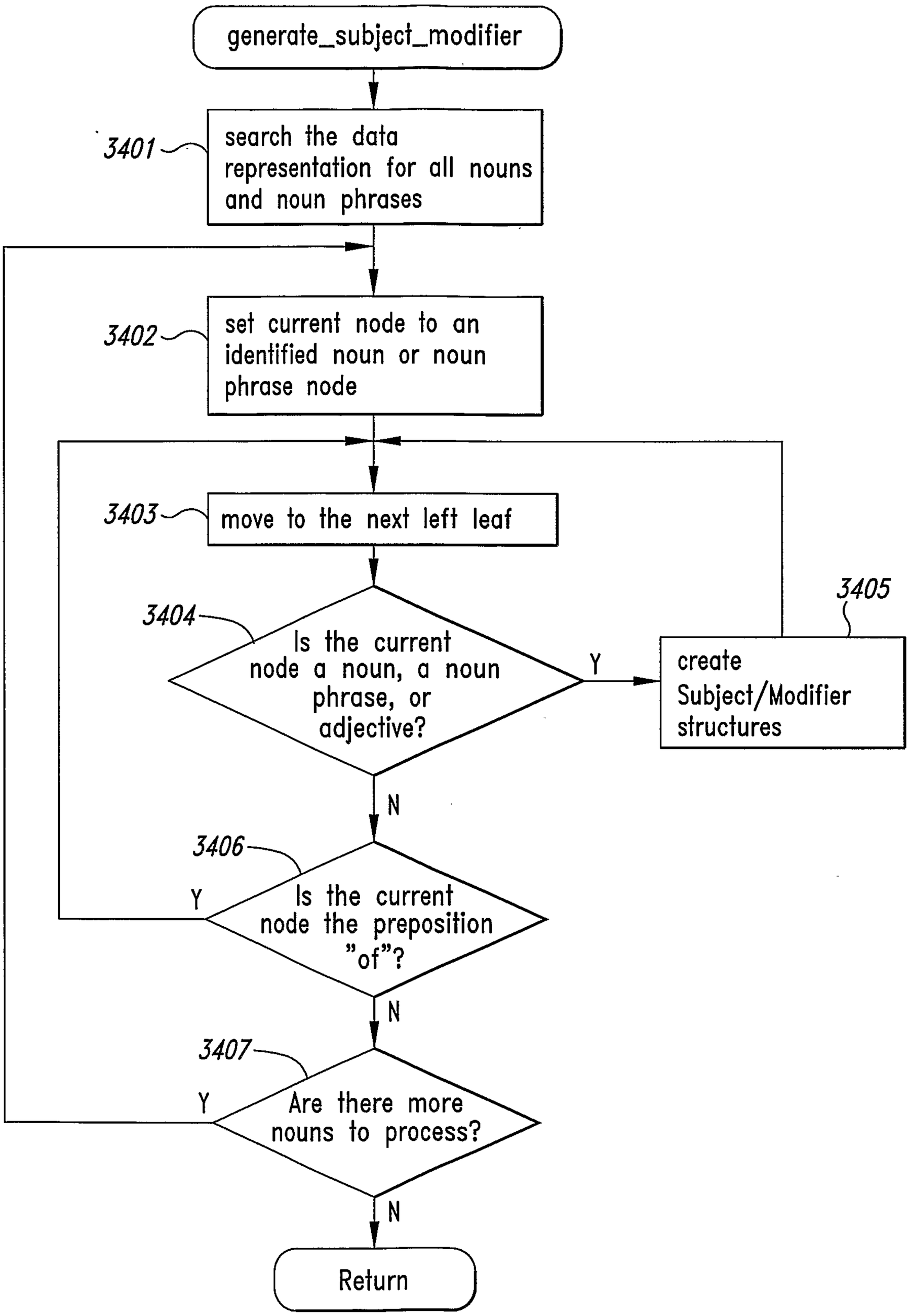

generate_subject_modifier
3401
search the data representation for all nouns and noun phrases
3402
set current node to an identified noun or noun phrase node
3403
move to the next left leaf
3404
Is the current node a noun, a noun phrase, or adjective?
Y
3405
create Subject/Modifier structures
N
3406
Y
Is the current node the preposition "of"?
N
3407
Y
Are there more nouns to process?
N
Return

Fig. 34

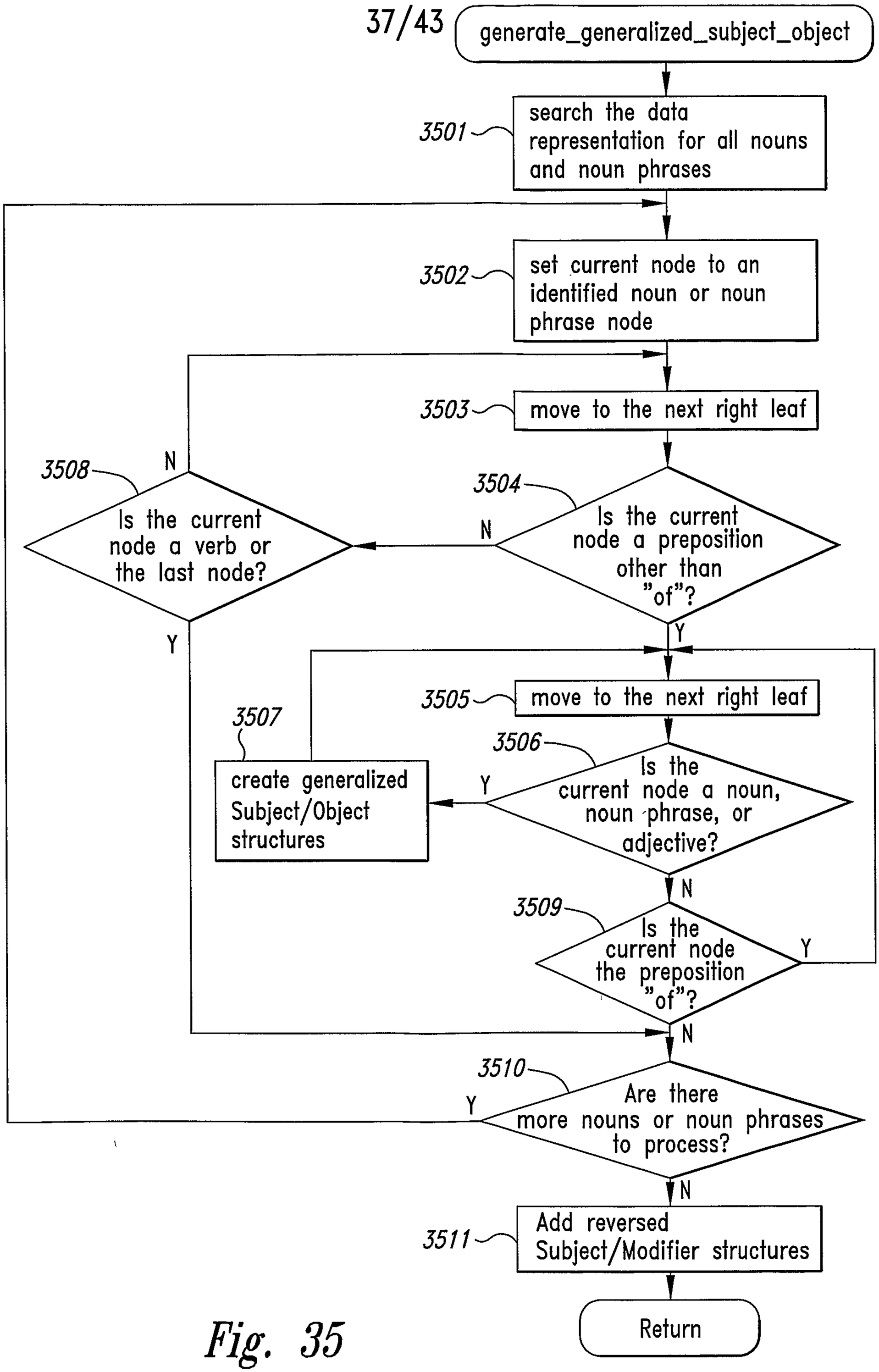

generate_generalized_subject_object
3501
search the data representation for all nouns and noun phrases
3502
set current node to an identified noun or noun phrase node
3503
move to the next right leaf
3504
Is the current node a preposition other than "of"?
N
Y
3508
Is the current node a verb or the last node?
N
Y
3505
move to the next right leaf
3506
Is the current node a noun, noun phrase, or adjective?
Y
N
3507
create generalized Subject/Object structures
3509
Is the current node the preposition "of"?
Y
N
3510
Are there more nouns or noun phrases to process?
Y
N
3511
Add reversed Subject/Modifier structures
Return

Fig. 35

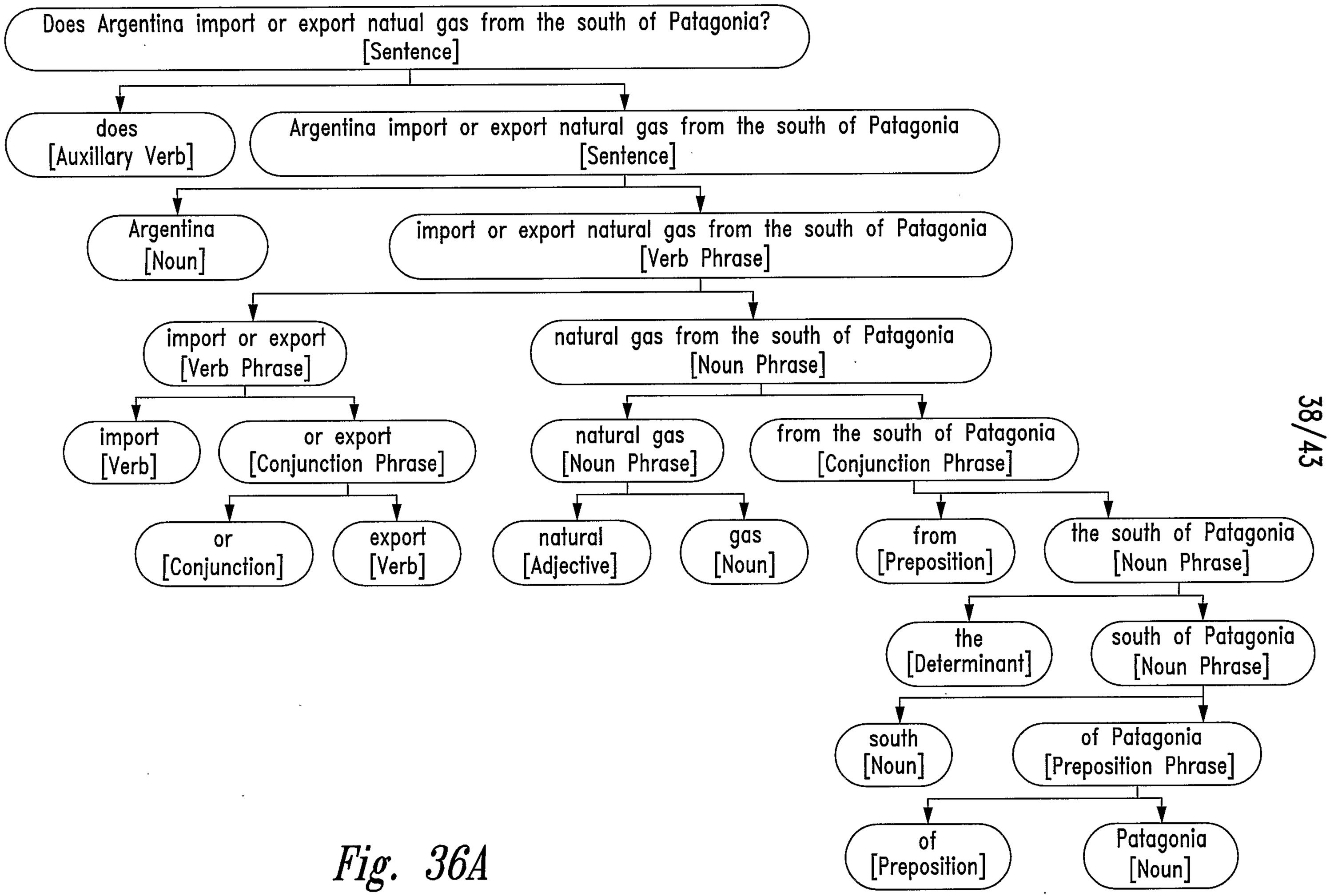

Does Argentina import or export natual gas from the south of Patagonia? [Sentence]
does [Auxillary Verb]
Argentina import or export natural gas from the south of Patagonia [Sentence]
Argentina [Noun]
import or export natural gas from the south of Patagonia [Verb Phrase]
import or export [Verb Phrase]
natural gas from the south of Patagonia [Noun Phrase]
import [Verb]
or export [Conjunction Phrase]
natural gas [Noun Phrase]
from the south of Patagonia [Conjunction Phrase]
or [Conjunction]
export [Verb]
natural [Adjective]
gas [Noun]
from [Preposition]
the south of Patagonia [Noun Phrase]
the [Determinant]
south of Patagonia [Noun Phrase]
south [Noun]
of Patagonia [Preposition Phrase]
of [Preposition]
Patagonia [Noun]

Fig. 36A

Does Argentina import or export natural gas from the south of Patagonia?

Row Number	Subject	Verb	Object	Preposition	Modifier
1	Argentina	import			
2	Argentina	export			
3		import	natural gas		
4		import	gas		
5		export	natural gas		
6		export	gas		
7		import		from	south
8		import		from	Patagonia
9		export		from	south
10		export		from	Patagonia
11	Argentina		natural gas		
12	Argentina		gas		
13	Argentina		Patagonia		
14	Argentina		south		
15	Patagonia				south
16	natural gas		south		
17	gas		south		
18	natural gas		Patagonia		
19	gas		Patagonia		
20	south		Patagonia		
21	natural gas				Argentina
22	gas				Argentina
23	Patagonia				Argentina
24	south				Argentina
25	south				natural gas
26	south				gas
27	Patagonia				natural gas
28	Patagonia				gas

Fig. 36B

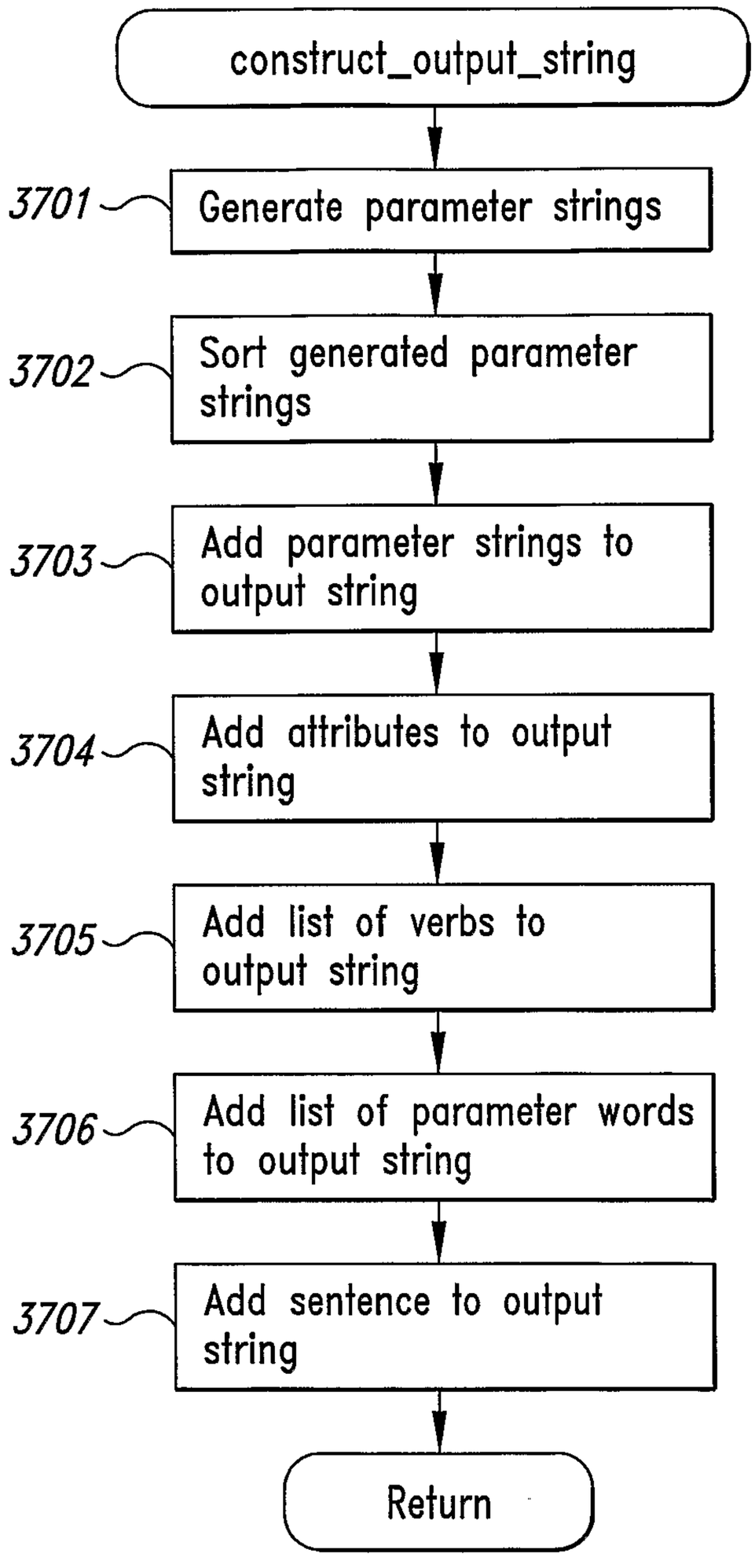

construct_output_string
3701
Generate parameter strings
3702
Sort generated parameter strings
3703
Add parameter strings to output string
3704
Add attributes to output string
3705
Add list of verbs to output string
3706
Add list of parameter words to output string
3707
Add sentence to output string
Return

Fig. 37

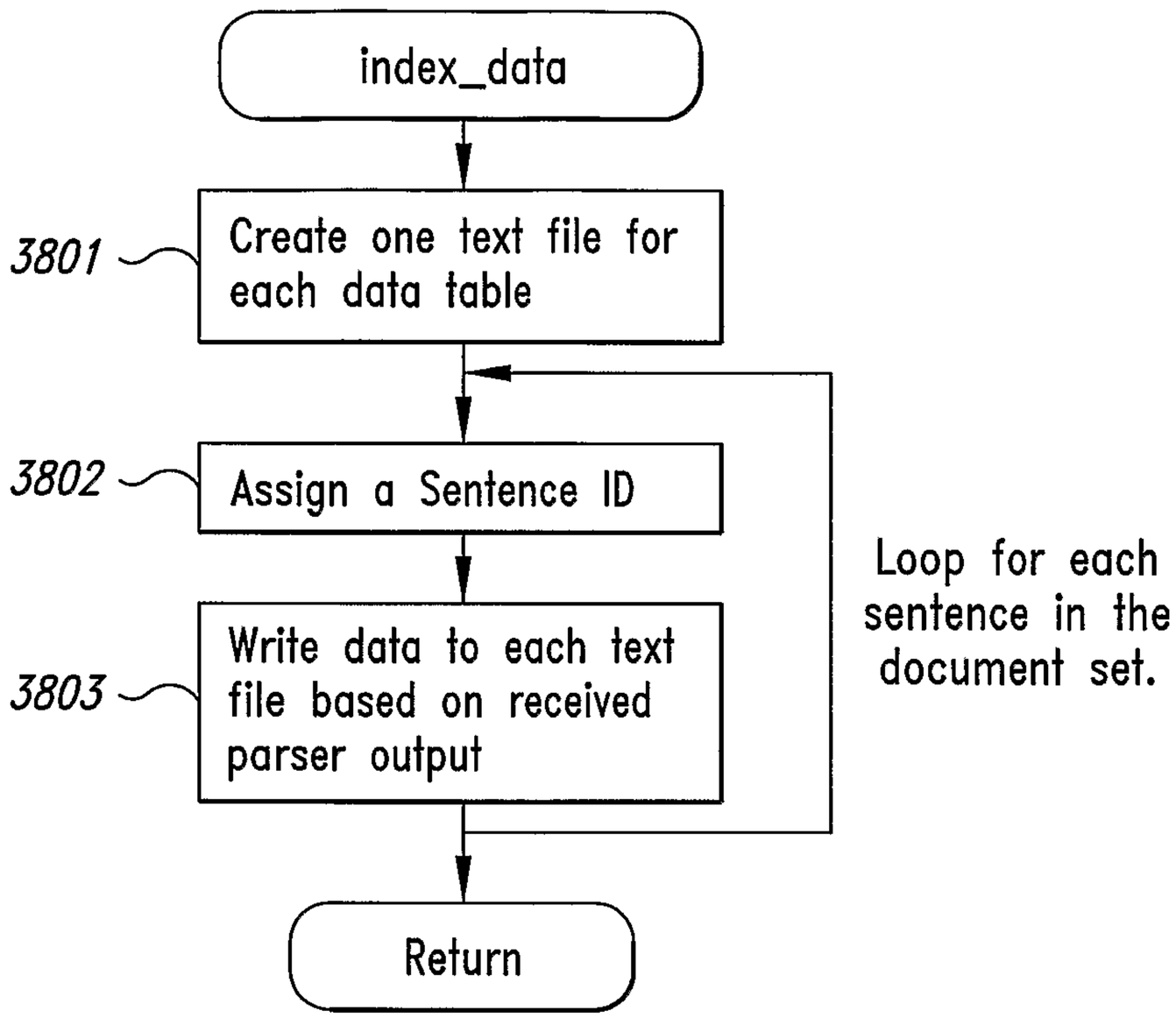

index_data
3801
Create one text file for each data table
3802
Assign a Sentence ID
3803
Write data to each text file based on received parser output
Return
Loop for each sentence in the document set.

Fig. 38

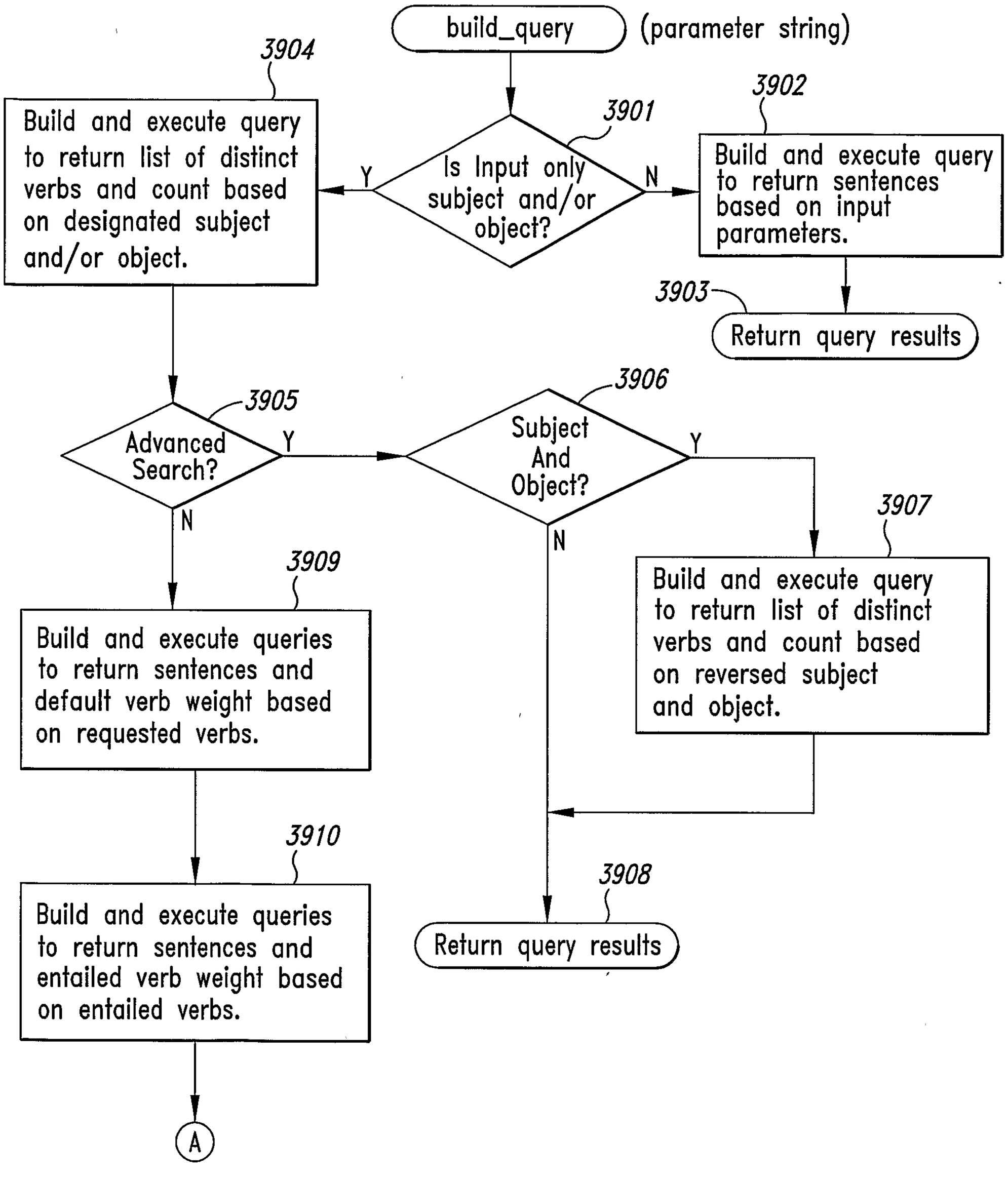

build_query
(parameter string)
3901
Is Input only subject and/or object?
Y
N
3904
Build and execute query to return list of distinct verbs and count based on designated subject and/or object.
3902
Build and execute query to return sentences based on input parameters.
3903
Return query results
3905
Advanced Search?
Y
N
3906
Subject And Object?
Y
N
3907
Build and execute query to return list of distinct verbs and count based on reversed subject and object.
3909
Build and execute queries to return sentences and default verb weight based on requested verbs.
3910
Build and execute queries to return sentences and entailed verb weight based on entailed verbs.
3908
Return query results
A

Fig. 39A

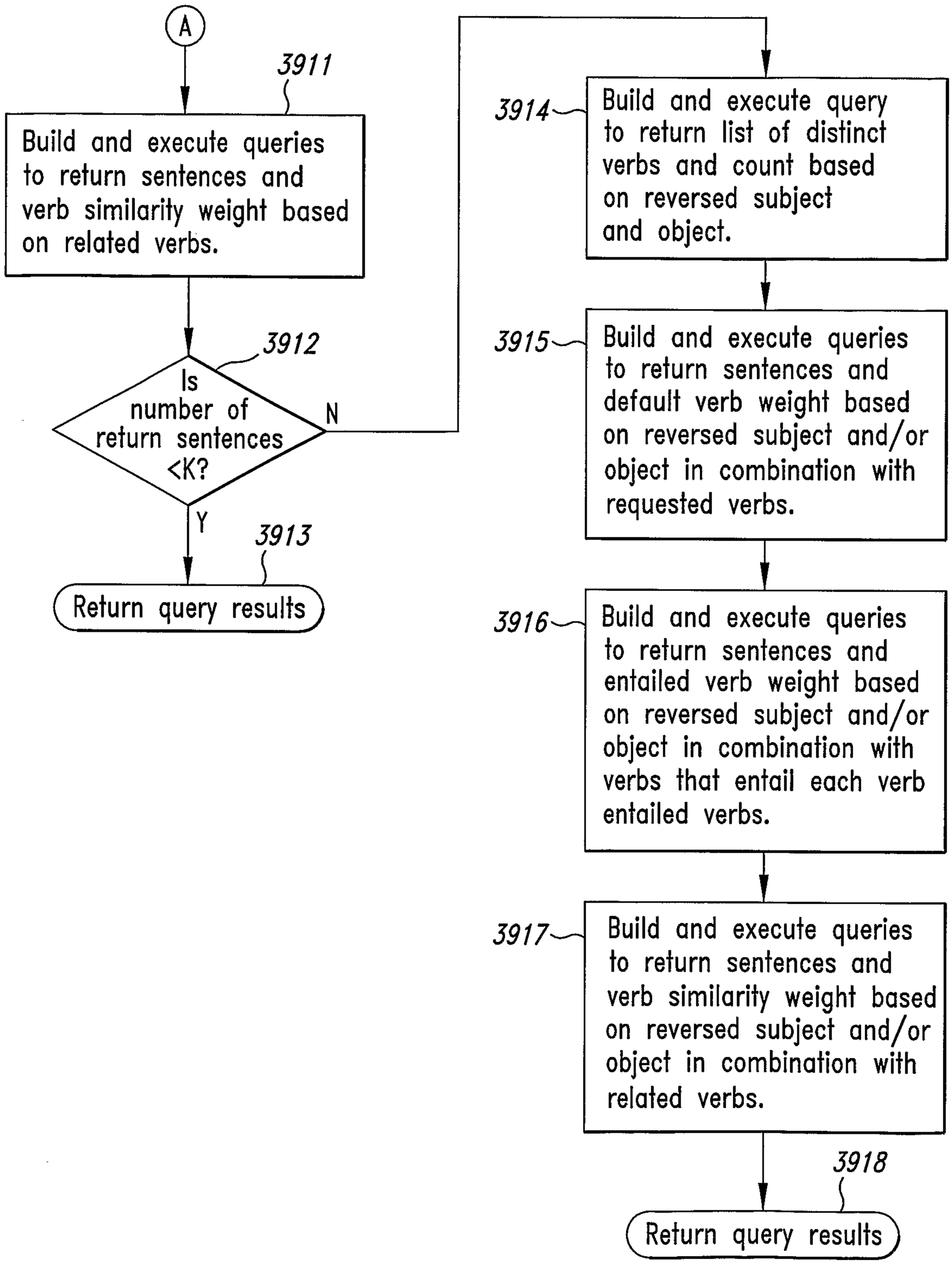
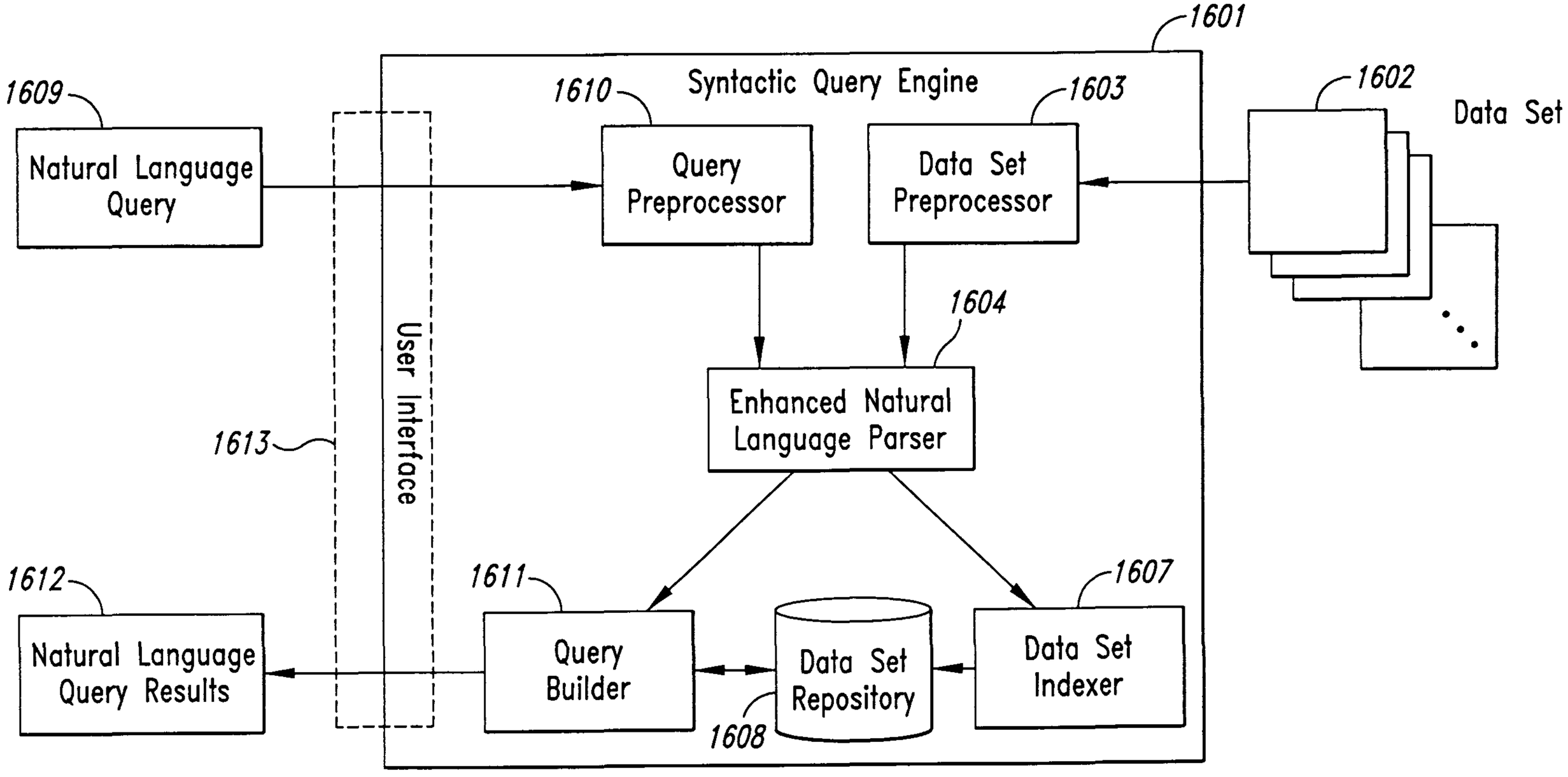

A
3911
Build and execute queries to return sentences and verb similarity weight based on related verbs.
3912
Is number of return sentences <K?
N
Y
3913
Return query results
3914
Build and execute query to return list of distinct verbs and count based on reversed subject and object.
3915
Build and execute queries to return sentences and default verb weight based on reversed subject and/or object in combination with requested verbs.
3916
Build and execute queries to return sentences and entailed verb weight based on reversed subject and/or object in combination with verbs that entail each verb entailed verbs.
3917
Build and execute queries to return sentences and verb similarity weight based on reversed subject and/or object in combination with related verbs.
3918
Return query results

Fig. 39B


1601
Syntactic Query Engine
1609
Natural Language Query
1610
Query Preprocessor
1603
Data Set Preprocessor
1602
Data Set
1604
Enhanced Natural Language Parser
1613
User Interface
1612
Natural Language Query Results
1611
Query Builder
1608
Data Set Repository
1607
Data Set Indexer