



(19) **United States**

(12) **Patent Application Publication**
Betz et al.

(10) **Pub. No.: US 2007/0150800 A1**

(43) **Pub. Date: Jun. 28, 2007**

(54) **UNSUPERVISED EXTRACTION OF FACTS**

Publication Classification

(76) Inventors: **Jonathan T. Betz**, Summit, NJ (US);
Shubin Zhao, Jersey City, NJ (US)

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/500**

Correspondence Address:

GOOGLE / FENWICK
SILICON VALLEY CENTER
801 CALIFORNIA ST.
MOUNTAIN VIEW, CA 94041 (US)

(57) **ABSTRACT**

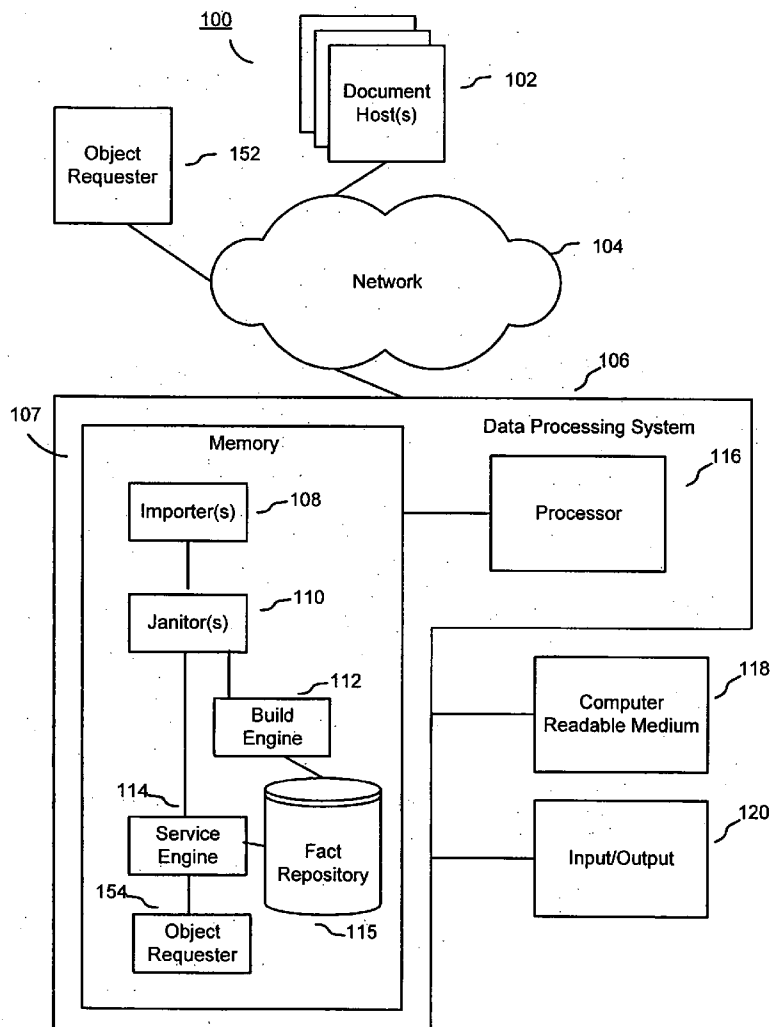
A system and method for extracting facts from documents. A fact is extracted from a first document. The attribute and value of the fact extracted from the first document are used as a seed attribute-value pair. A second document containing the seed attribute-value pair is analyzed to determine a contextual pattern used in the second document. The contextual pattern is used to extract other attribute-value pairs from the second document. The extracted attributes and values are stored as facts.

(21) Appl. No.: **11/394,414**

(22) Filed: **Mar. 31, 2006**

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/142,853, filed on May 31, 2005.



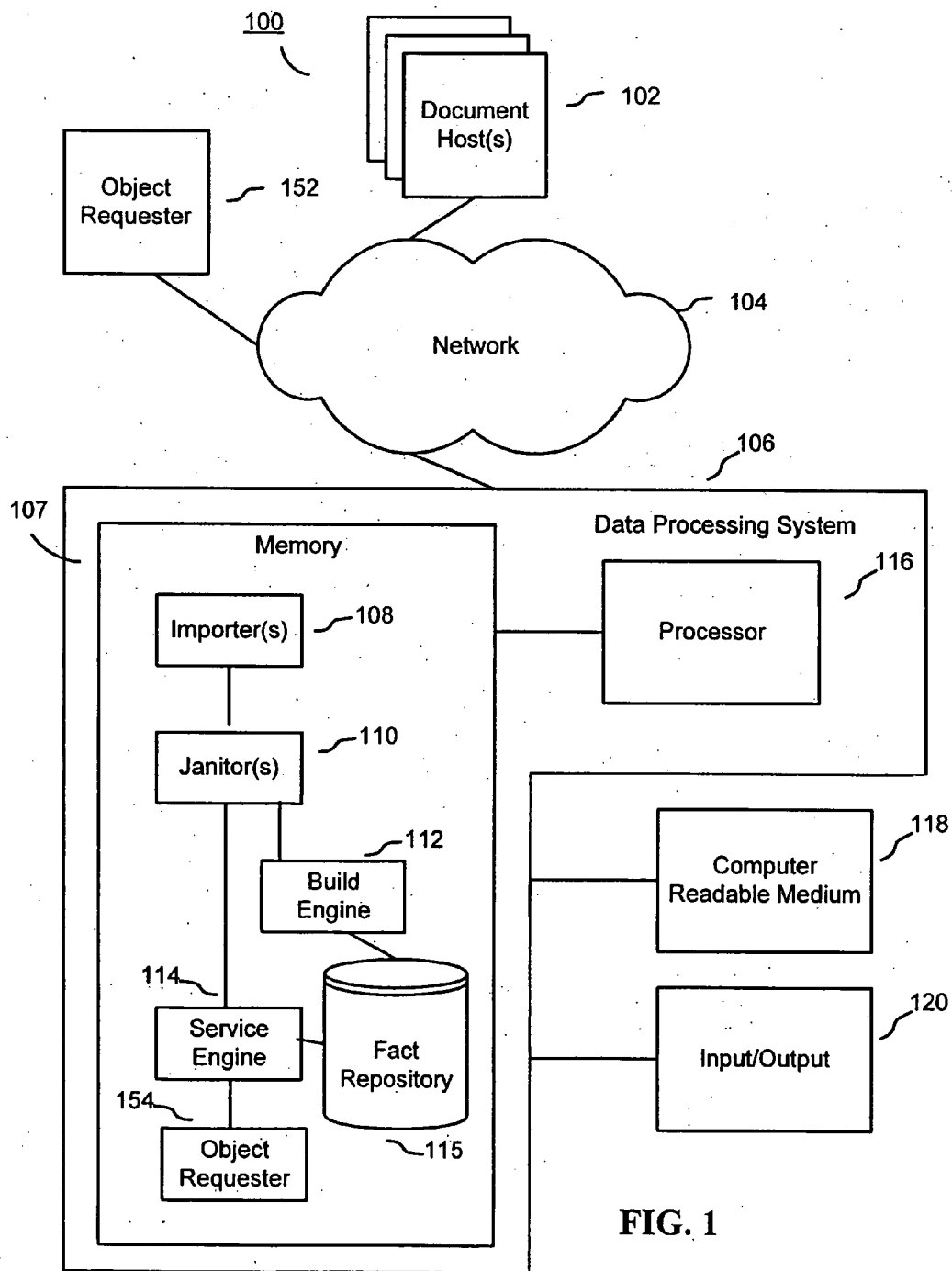


FIG. 1

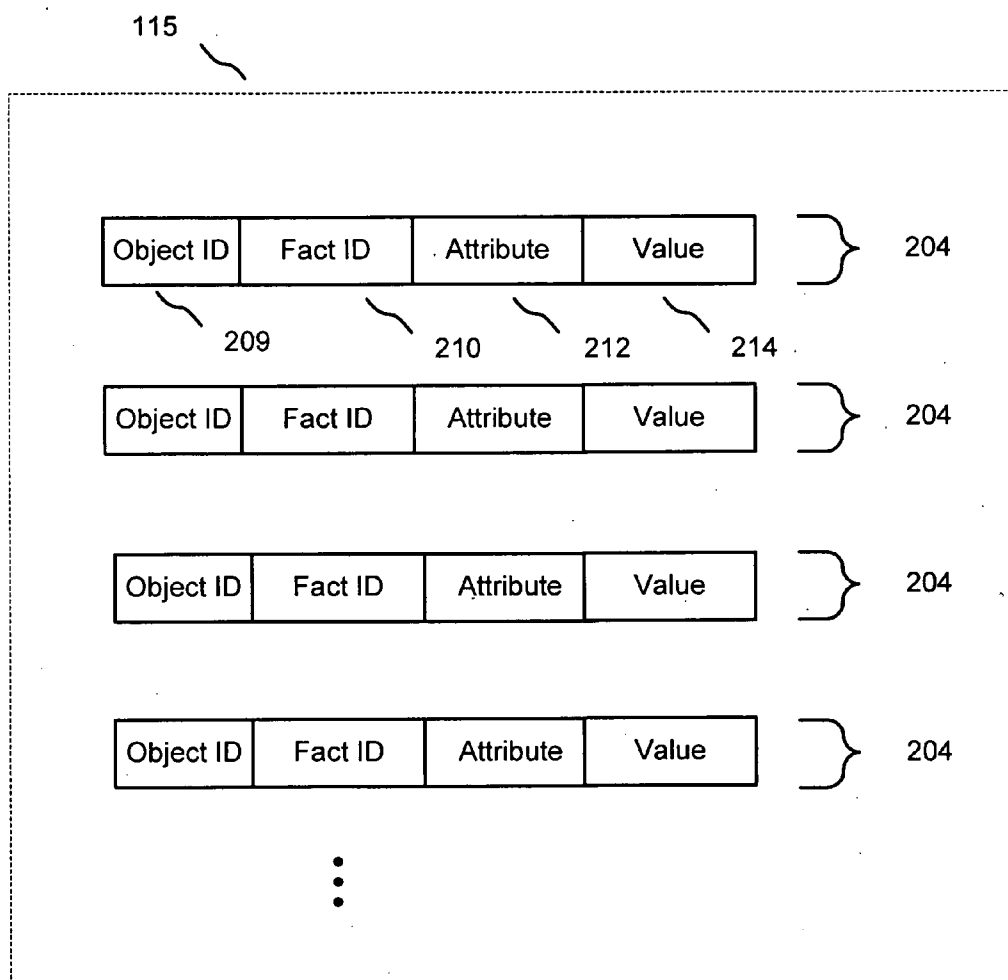


FIG. 2(a)
Example Format of Facts in Repository (each fact is associated with an object ID)

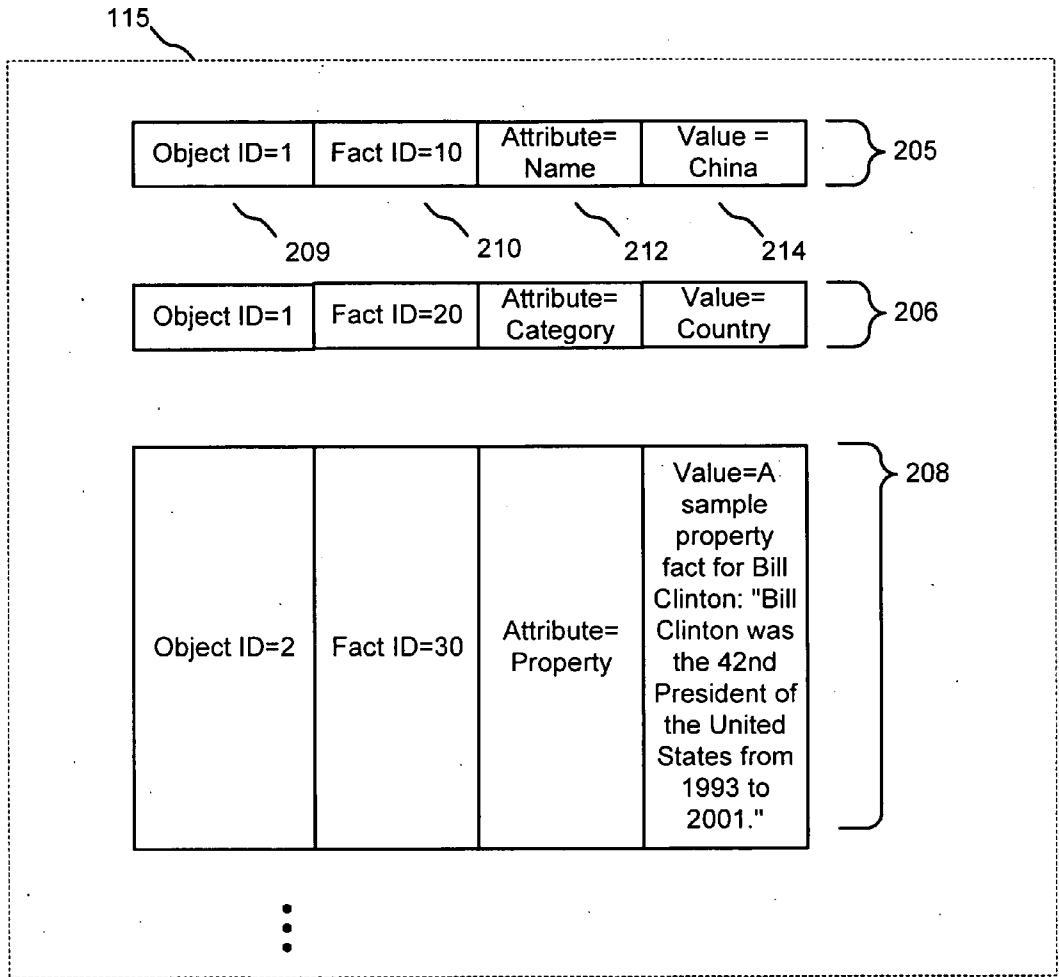


FIG. 2(b)
Example Facts in Repository (each fact is associated with an object ID)

Object ID=1	Fact ID=10
Object ID=1	Fact ID=20
Object ID=1	Fact ID=30
Object ID=2	Fact ID=40

⋮

210

FIG. 2(c)
Example Object
Reference Table

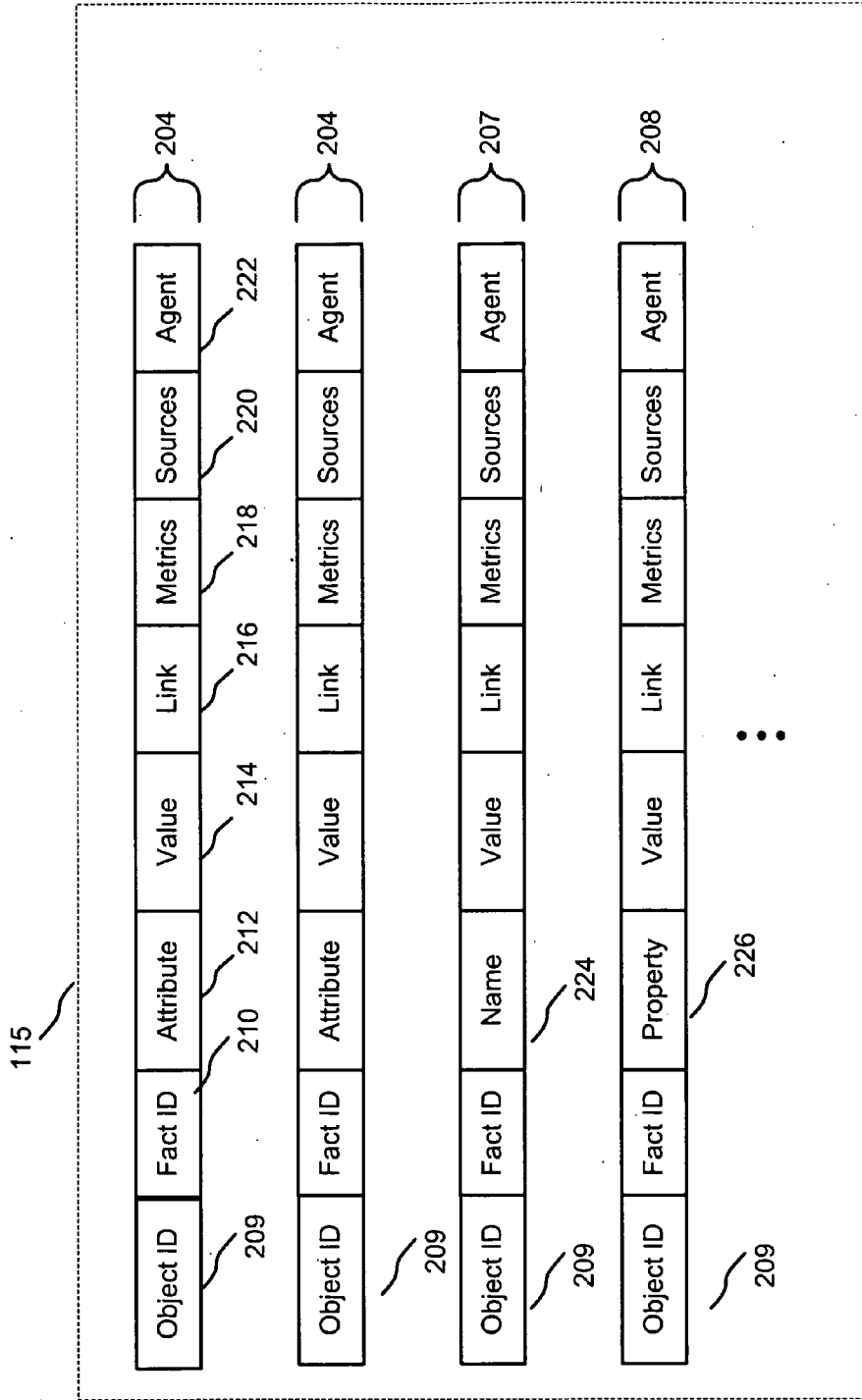


FIG. 2(d)
Example Format of Facts in
Repository (each fact is associated
with an object ID)

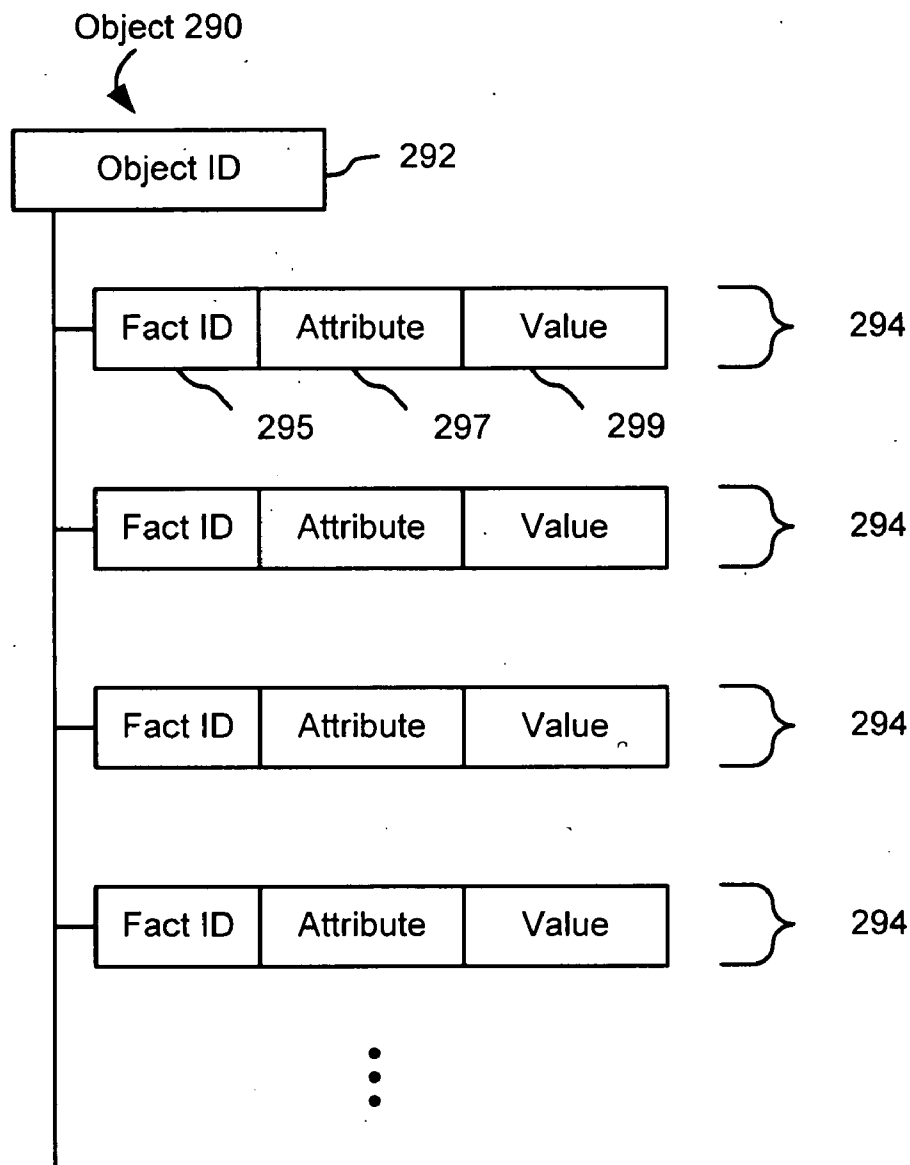


FIG. 2(e)
Example Objects

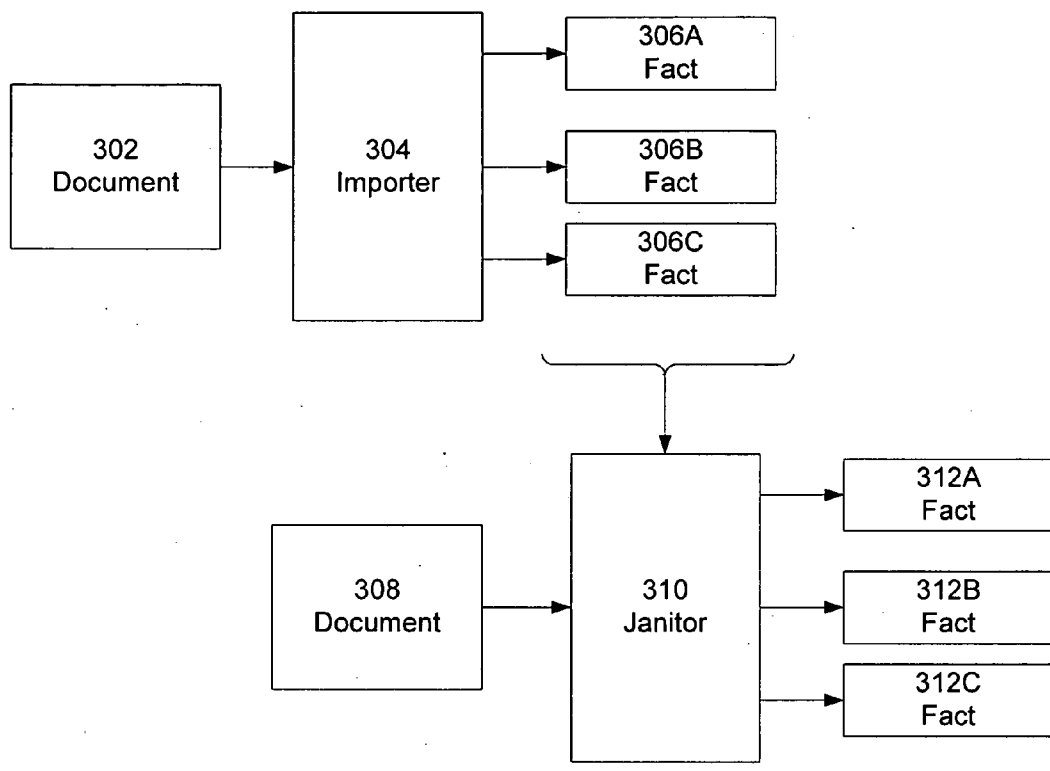


FIG. 3(a)

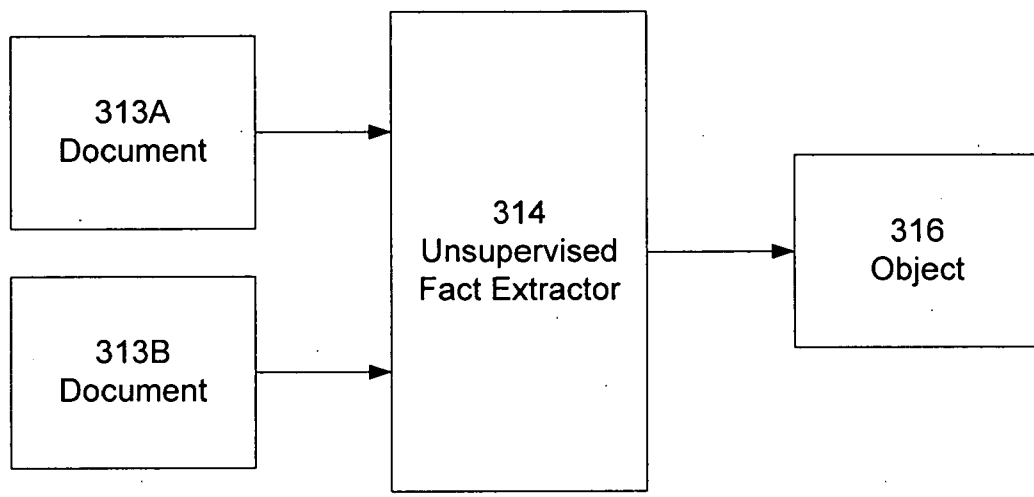


FIG. 3(b)

ADVERTISEMENT

Britney Spears pictures, picture gallery, celebrity greeting cards, photos, pics, snaps, high quality pictures, rare pictures.

Profile
check whether you know these

Biography
a detailed story of Britney Spears

Photo Gallery
check out the multiple picture galleries

Related Sites

PROFILE

NAME:	BRITNEY SPEARS
PROFESSION:	ACTRESS, SINGER
DATE OF BIRTH:	DECEMBER 2, 1981
PLACE OF BIRTH:	KENTWOOD, LA
SIGN:	SAGITTARIUS
EYE COLOR:	BROWN
HAIR COLOR:	BROWN

BIOGRAPHY

To say that Britney Spears's life has changed over the course of the past

402



404

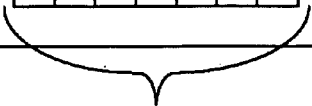


FIG. 4

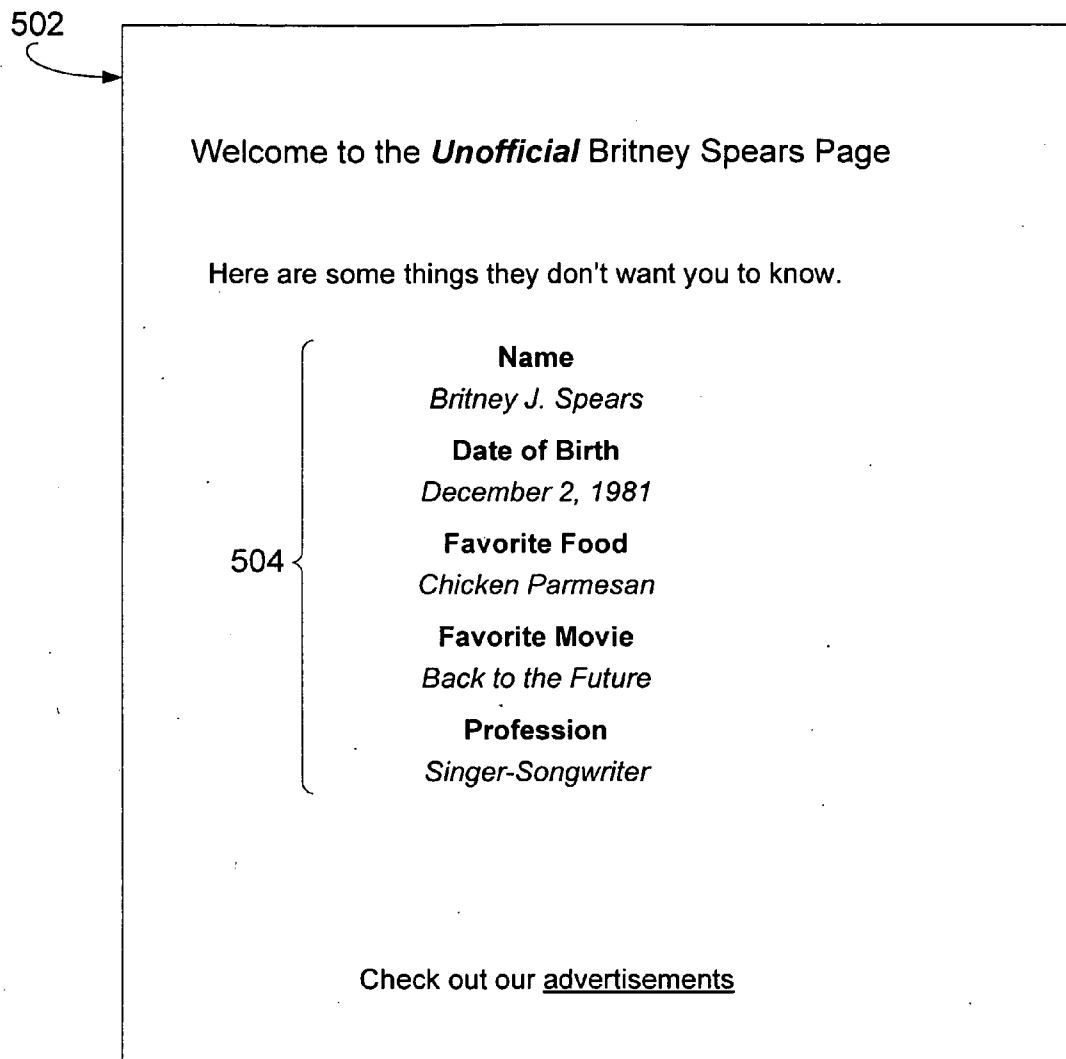


FIG. 5

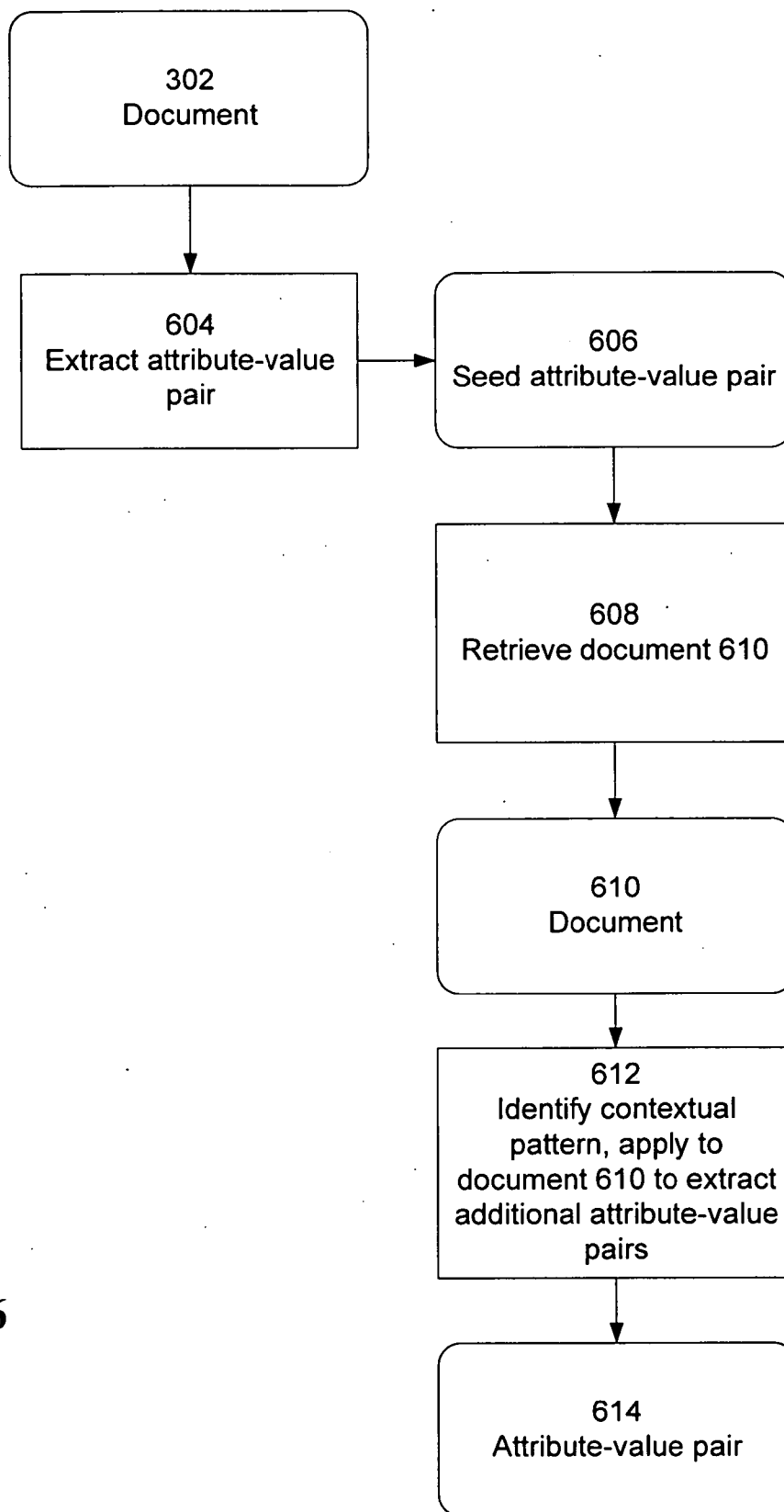


FIG. 6

UNSUPERVISED EXTRACTION OF FACTS

RELATED APPLICATIONS

[0001] This application is a continuation-in-part of U.S. application Ser. No. 11/142,853, entitled, "Learning Facts from Semi-Structured Text", by Shubin Zhao and Jonathan Betz, filed on May 31, 2005, which is hereby incorporated by reference.

[0002] This application is related to the following applications, all of which are hereby incorporated by reference:

[0003] U.S. application Ser. No. 11/024,784, entitled, "Supplementing Search Results with Information of Interest", by Jonathan Betz, filed on Dec. 30, 2004;

[0004] U.S. application Ser. No. 11/142,765, entitled, "Identifying the Unifying Subject of a Set of Facts", by Jonathan Betz, filed on May 31, 2005;

[0005] U.S. application Ser. No. 11/097,588, entitled, "Corroborating Facts Extracted from Multiple Sources", by Jonathan Betz, filed on Mar. 31, 2005;

[0006] U.S. application Ser. No. 11/366,162, entitled "Generating Structured Information," filed Mar. 1, 2006, by Egon Pasztor and Daniel Egnor, Attorney Docket number 24207-11149;

[0007] U.S. application Ser. No. 11/357,748, entitled "Support for Object Search", filed Feb. 17, 2006, by Alex Kehlenbeck, Andrew W. Hogue, Attorney Docket No. 24207-10945;

[0008] U.S. application Ser. No. 11/342,290, entitled "Data Object Visualization", filed on Jan. 27, 2006, by Andrew W. Hogue, David Vespe, Alex Kehlenbeck, Mike Gordon, Jeffrey C. Reynar, David Alpert;

[0009] U.S. application Ser. No. 11/342,293, entitled "Data Object Visualization Using Maps", filed on Jan. 27, 2006, by Andrew W. Hogue, David Vespe, Alex Kehlenbeck, Mike Gordon, Jeffrey C. Reynar, David Alpert;

[0010] U.S. application Ser. No. 11/356,679, entitled "Query Language", filed Feb. 17, 2006, by Andrew W. Hogue, Doug Rohde, Attorney Docket No. 24207-10948;

[0011] U.S. application Ser. No. 11/356,837, entitled "Automatic Object Reference Identification and Linking in a Browseable Fact Repository", filed Feb. 17, 2006, by Andrew W. Hogue, Attorney Docket No. 24207-10961;

[0012] U.S. application Ser. No. 11/356,851, entitled "Browseable Fact Repository", filed Feb. 17, 2006, by Andrew W. Hogue, Jonathan T. Betz, Attorney Docket No. 24207-10949;

[0013] U.S. application Ser. No. 11/356,842, entitled "ID Persistence Through Normalization", filed Feb. 17, 2006, by Jonathan T. Betz, Andrew W. Hogue, Attorney Docket No. 24207-10950;

[0014] U.S. application Ser. No. 11/356,728, entitled "Annotation Framework", filed Feb. 17, 2006, by Tom Richford, Jonathan T. Betz, Attorney Docket No. 24207-10951;

[0015] U.S. application Ser. No. 11/341,069, entitled "Object Categorization for Information Extraction", filed on Jan. 27, 2006, by Jonathan T. Betz, Attorney Docket No. 24207-10952;

[0016] U.S. application Ser. No. 11/356,838, entitled "Modular Architecture for Entity Normalization", filed Feb. 17, 2006, by Jonathan T. Betz, Farhan Shamsi, Attorney Docket No. 24207-10953;

[0017] U.S. application Ser. No. 11/356,765, entitled "Attribute Entropy as a Signal in Object Normalization", filed Feb. 17, 2006, by Jonathan T. Betz, Vivek Menezes, Attorney Docket No. 24207-10954;

[0018] U.S. application Ser. No. 11/341,907, entitled "Designating Data Objects for Analysis", filed on Jan. 27, 2006, by Andrew W. Hogue, David Vespe, Alex Kehlenbeck, Mike Gordon, Jeffrey C. Reynar, David Alpert;

[0019] U.S. application Ser. No. 11/342,277, entitled "Data Object Visualization Using Graphs", filed on Jan. 27, 2006, by Andrew W. Hogue, David Vespe, Alex Kehlenbeck, Mike Gordon, Jeffrey C. Reynar, David Alpert;

[0020] U.S. application Ser. No. _____, entitled "Entity Normalization Via Name Normalization", filed on Mar. 31, 2006, by Jonathan T. Betz, Attorney Docket No. 24207-11047;

[0021] U.S. application Ser. No. _____, entitled "Determining Document Subject by Using Title and Anchor Text of Related Documents", filed on Mar. 31, 2006, by Shubin Zhao, Attorney Docket No. 24207-11049;

[0022] U.S. application Ser. No. _____, entitled "Anchor Text Summarization for Corroboration", filed on Mar. 31, 2006, by Jonathan T. Betz and Shubin Zhao, Attorney Docket No. 24207-11046;

[0023] U.S. application Ser. No. _____, entitled "Mechanism for Inferring Facts from a Fact Repository", filed on Mar. 31, 2006, by Andrew Hogue and Jonathan Betz, Attorney Docket No. 24207-11048.

TECHNICAL FIELD

[0024] The disclosed embodiments relate generally to fact databases. More particularly, the disclosed embodiments relate to extracting facts from documents.

BACKGROUND

[0025] The internet provides access to a wealth of information. Documents created by authors all over the world are freely available for reading, indexing, and extraction of information. This incredible diversity of fact and opinion that make the internet the ultimate information source.

[0026] However, this same diversity of information creates a considerable challenge when extracting information. Information may be presented in a variety of formats, languages, and layouts. A human user may (or may not) be able to decipher individual documents to gather the information contained therein, but these differences may confuse or mislead an automated extraction system, resulting in information of little or no value. Extracting information from documents of various formats poses a formidable challenge to efforts to create an automated extraction system.

SUMMARY

[0027] A system and method for extracting facts from documents. A fact is extracted from a first document. The

attribute and value of the fact extracted from the first document is used as a seed attribute-value pair. A second document containing the seed attribute-value pair is analyzed to determine a contextual pattern used in the second document. The contextual pattern is used to extract other attribute-value pairs from the second document. The extracted attributes and values are stored as facts.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1 shows a network, in accordance with a preferred embodiment of the invention.

[0029] FIGS. 2(a)-2(d) are block diagrams illustrating a data structure for facts within a repository of FIG. 1 in accordance with preferred embodiments of the invention.

[0030] FIG. 2(e) is a block diagram illustrating an alternate data structure for facts and objects in accordance with preferred embodiments of the invention.

[0031] FIG. 3(a) is a block diagram illustrating the extraction of facts from a plurality of documents, according to one embodiment of the present invention.

[0032] FIG. 3(b) is a block diagram illustrating the extraction of facts from a plurality of documents to produce an object, according to one embodiment of the present invention.

[0033] FIG. 4 is an example of a document which can be processed using predefined patterns, according to one embodiment of the present invention.

[0034] FIG. 5 is an example of a document which can be processed using contextual patterns, according to one embodiment of the present invention.

[0035] FIG. 6 is a flow chart illustrating a method for extracting facts, according to one embodiment of the present invention.

DESCRIPTION OF EMBODIMENTS

[0036] Embodiments of the present invention are now described with reference to the figures where like reference numbers indicate identical or functionally similar elements.

[0037] FIG. 1 shows a system architecture 100 adapted to support one embodiment of the invention. FIG. 1 shows components used to add facts into, and retrieve facts from a repository 115. The system architecture 100 includes a network 104, through which any number of document hosts 102 communicate with a data processing system 106, along with any number of object requesters 152, 154.

[0038] Document hosts 102 store documents and provide access to documents. A document is comprised of any machine-readable data including any combination of text, graphics, multimedia content, etc. A document may be encoded in a markup language, such as Hypertext Markup Language (HTML), i.e., a web page, in an interpreted language (e.g., JavaScript) or in any other computer readable or executable format. A document can include one or more hyperlinks to other documents. A typical document will include one or more facts within its content. A document stored in a document host 102 may be located and/or identified by a Uniform Resource Locator (URL), or Web address, or any other appropriate form of identification and/or location. A document host 102 is implemented by a

computer system, and typically includes a server adapted to communicate over the network 104 via networking protocols (e.g., TCP/IP), as well as application and presentation protocols (e.g., HTTP, HTML, SOAP, D-HTML, Java). The documents stored by a host 102 are typically held in a file directory, a database, or other data repository. A host 102 can be implemented in any computing device (e.g., from a PDA or personal computer, a workstation, mini-computer, or mainframe, to a cluster or grid of computers), as well as in any processor architecture or operating system.

[0039] FIG. 1 shows components used to manage facts in a fact repository 115. Data processing system 106 includes one or more importers 108, one or more janitors 110, a build engine 112, a service engine 114, and a fact repository 115 (also called simply a "repository"). Each of the foregoing are implemented, in one embodiment, as software modules (or programs) executed by processor 116. Importers 108 operate to process documents received from the document hosts, read the data content of documents, and extract facts (as operationally and programmatically defined within the data processing system 106) from such documents. The importers 108 also determine the subject or subjects with which the facts are associated, and extract such facts into individual items of data, for storage in the fact repository 115. In one embodiment, there are different types of importers 108 for different types of documents, for example, dependent on the format or document type.

[0040] Janitors 110 operate to process facts extracted by importer 108. This processing can include but is not limited to, data cleansing, object merging, and fact induction. In one embodiment, there are a number of different janitors 110 that perform different types of data management operations on the facts. For example, one janitor 110 may traverse some set of facts in the repository 115 to find duplicate facts (that is, facts that convey the same factual information) and merge them. Another janitor 110 may also normalize facts into standard formats. Another janitor 110 may also remove unwanted facts from repository 115, such as facts related to pornographic content. Other types of janitors 110 may be implemented, depending on the types of data management functions desired, such as translation, compression, spelling or grammar correction, and the like.

[0041] Various janitors 110 act on facts to normalize attribute names, and values and delete duplicate and near-duplicate facts so an object does not have redundant information. For example, we might find on one page that Britney Spears' birthday is "Dec. 2, 1981" while on another page that her date of birth is "Dec. 2, 1981." Birthday and Date of Birth might both be rewritten as Birthdate by one janitor and then another janitor might notice that Dec. 2, 1981 and Dec. 2, 1981 are different forms of the same date. It would choose the preferred form, remove the other fact and combine the source lists for the two facts. As a result when you look at the source pages for this fact, on some you'll find an exact match of the fact and on others text that is considered to be synonymous with the fact.

[0042] Build engine 112 builds and manages the repository 115. Service engine 114 is an interface for querying the repository 115. Service engine 114's main function is to process queries, score matching objects, and return them to the caller but it is also used by janitor 110.

[0043] Repository 115 stores factual information extracted from a plurality of documents that are located on document

hosts **102**. A document from which a particular fact may be extracted is a source document (or “source”) of that particular fact. In other words, a source of a fact includes that fact (or a synonymous fact) within its contents.

[**0044**] Repository **115** contains one or more facts. In one embodiment, each fact is associated with exactly one object. One implementation for this association includes in each fact an object ID that uniquely identifies the object of the association. In this manner, any number of facts may be associated with an individual object, by including the object ID for that object in the facts. In one embodiment, objects themselves are not physically stored in the repository **115**, but rather are defined by the set or group of facts with the same associated object ID, as described below. Further details about facts in repository **115** are described below, in relation to FIGS. **2(a)**-**2(d)**.

[**0045**] It should be appreciated that in practice at least some of the components of the data processing system **106** will be distributed over multiple computers, communicating over a network. For example, repository **115** may be deployed over multiple servers. As another example, the janitors **110** may be located on any number of different computers. For convenience of explanation, however, the components of the data processing system **106** are discussed as though they were implemented on a single computer.

[**0046**] In another embodiment, some or all of document hosts **102** are located on data processing system **106** instead of being coupled to data processing system **106** by a network. For example, importer **108** may import facts from a database that is a part of or associated with data processing system **106**.

[**0047**] FIG. **1** also includes components to access repository **115** on behalf of one or more object requesters **152**, **154**. Object requesters are entities that request objects from repository **115**. Object requesters **152**, **154** may be understood as clients of the system **106**, and can be implemented in any computer device or architecture. As shown in FIG. **1**, a first object requester **152** is located remotely from system **106**, while a second object requester **154** is located in data processing system **106**. For example, in a computer system hosting a blog, the blog may include a reference to an object whose facts are in repository **115**. An object requester **152**, such as a browser displaying the blog will access data processing system **106** so that the information of the facts associated with the object can be displayed as part of the blog web page. As a second example, janitor **110** or other entity considered to be part of data processing system **106** can function as object requester **154**, requesting the facts of objects from repository **115**.

[**0048**] FIG. **1** shows that data processing system **106** includes a memory **107** and one or more processors **116**. Memory **107** includes importers **108**, janitors **110**, build engine **112**, service engine **114**, and requester **154**, each of which are preferably implemented as instructions stored in memory **107** and executable by processor **116**. Memory **107** also includes repository **115**. Repository **115** can be stored in a memory of one or more computer systems or in a type of memory such as a disk. FIG. **1** also includes a computer readable medium **118** containing, for example, at least one of importers **108**, janitors **110**, build engine **112**, service engine **114**, requester **154**, and at least some portions of repository **115**. FIG. **1** also includes one or more input/

output devices **120** that allow data to be input and output to and from data processing system **106**. It will be understood that data processing system **106** preferably also includes standard software components such as operating systems and the like and further preferably includes standard hardware components not shown in the figure for clarity of example.

[**0049**] FIG. **2(a)** shows an example format of a data structure for facts within repository **115**, according to some embodiments of the invention. As described above, the repository **115** includes facts **204**. Each fact **204** includes a unique identifier for that fact, such as a fact ID **210**. Each fact **204** includes at least an attribute **212** and a value **214**. For example, a fact associated with an object representing George Washington may include an attribute of “date of birth” and a value of “Feb. 22, 1732.” In one embodiment, all facts are stored as alphanumeric characters since they are extracted from web pages. In another embodiment, facts also can store binary data values. Other embodiments, however, may store fact values as mixed types, or in encoded formats.

[**0050**] As described above, each fact is associated with an object ID **209** that identifies the object that the fact describes. Thus, each fact that is associated with a same entity (such as George Washington), will have the same object ID **209**. In one embodiment, objects are not stored as separate data entities in memory. In this embodiment, the facts associated with an object contain the same object ID, but no physical object exists. In another embodiment, objects are stored as data entities in memory, and include references (for example, pointers or IDs) to the facts associated with the object. The logical data structure of a fact can take various forms; in general, a fact is represented by a tuple that includes a fact ID, an attribute, a value, and an object ID. The storage implementation of a fact can be in any underlying physical data structure.

[**0051**] FIG. **2(b)** shows an example of facts having respective fact IDs of **10**, **20**, and **30** in repository **115**. Facts **10** and **20** are associated with an object identified by object ID “1.” Fact **10** has an attribute of “Name” and a value of “China.” Fact **20** has an attribute of “Category” and a value of “Country.” Thus, the object identified by object ID “1” has a name fact **205** with a value of “China” and a category fact **206** with a value of “Country.” Fact **30208** has an attribute of “Property” and a value of ““Bill Clinton was the 42nd President of the United States from 1993 to 2001.”” Thus, the object identified by object ID “2” has a property fact with a fact ID of **30** and a value of “Bill Clinton was the 42nd President of the United States from 1993 to 2001.” In the illustrated embodiment, each fact has one attribute and one value. The number of facts associated with an object is not limited; thus while only two facts are shown for the “China” object, in practice there may be dozens, even hundreds of facts associated with a given object. Also, the value fields of a fact need not be limited in size or content. For example, a fact about the economy of “China” with an attribute of “Economy” would have a value including several paragraphs of text, numbers, perhaps even tables of figures. This content can be formatted, for example, in a markup language. For example, a fact having an attribute “original html” might have a value of the original html text taken from the source web page.

[**0052**] Also, while the illustration of FIG. **2(b)** shows the explicit coding of object ID, fact ID, attribute, and value, in

practice the content of the fact can be implicitly coded as well (e.g., the first field being the object ID, the second field being the fact ID, the third field being the attribute, and the fourth field being the value). Other fields include but are not limited to: the language used to state the fact (English, etc.), how important the fact is, the source of the fact, a confidence value for the fact, and so on.

[0053] FIG. 2(c) shows an example object reference table 210 that is used in some embodiments. Not all embodiments include an object reference table. The object reference table 210 functions to efficiently maintain the associations between object IDs and fact IDs. In the absence of an object reference table 210, it is also possible to find all facts for a given object ID by querying the repository to find all facts with a particular object ID. While FIGS. 2(b) and 2(c) illustrate the object reference table 210 with explicit coding of object and fact IDs, the table also may contain just the ID values themselves in column or pair-wise arrangements.

[0054] FIG. 2(d) shows an example of a data structure for facts within repository 115, according to some embodiments of the invention showing an extended format of facts. In this example, the fields include an object reference link 216 to another object. The object reference link 216 can be an object ID of another object in the repository 115, or a reference to the location (e.g., table row) for the object in the object reference table 210. The object reference link 216 allows facts to have as values other objects. For example, for an object “United States,” there may be a fact with the attribute of “president” and the value of “George W. Bush,” with “George W. Bush” being an object having its own facts in repository 115. In some embodiments, the value field 214 stores the name of the linked object and the link 216 stores the object identifier of the linked object. Thus, this “president” fact would include the value 214 of “George W. Bush”, and object reference link 216 that contains the object ID for the “George W. Bush” object. In some other embodiments, facts 204 do not include a link field 216 because the value 214 of a fact 204 may store a link to another object.

[0055] Each fact 204 also may include one or more metrics 218. A metric provides an indication of the some quality of the fact. In some embodiments, the metrics include a confidence level and an importance level. The confidence level indicates the likelihood that the fact is correct. The importance level indicates the relevance of the fact to the object, compared to other facts for the same object. The importance level may optionally be viewed as a measure of how vital a fact is to an understanding of the entity or concept represented by the object.

[0056] Each fact 204 includes a list of one or more sources 220 that include the fact and from which the fact was extracted. Each source may be identified by a Uniform Resource Locator (URL), or Web address, or any other appropriate form of identification and/or location, such as a unique document identifier.

[0057] The facts illustrated in FIG. 2(d) include an agent field 222 that identifies the importer 108 that extracted the fact. For example, the importer 108 may be a specialized importer that extracts facts from a specific source (e.g., the pages of a particular web site, or family of web sites) or type of source (e.g., web pages that present factual information in tabular form), or an importer 108 that extracts facts from free text in documents throughout the Web, and so forth.

[0058] Some embodiments include one or more specialized facts, such as a name fact 207 and a property fact 208. A name fact 207 is a fact that conveys a name for the entity or concept represented by the object ID. A name fact 207 includes an attribute 224 of “name” and a value, which is the name of the object. For example, for an object representing the country Spain, a name fact would have the value “Spain.” A name fact 207, being a special instance of a general fact 204, includes the same fields as any other fact 204; it has an attribute, a value, a fact ID, metrics, sources, etc. The attribute 224 of a name fact 207 indicates that the fact is a name fact, and the value is the actual name. The name may be a string of characters. An object ID may have one or more associated name facts, as many entities or concepts can have more than one name. For example, an object ID representing Spain may have associated name facts conveying the country’s common name “Spain” and the official name “Kingdom of Spain.” As another example, an object ID representing the U.S. Patent and Trademark Office may have associated name facts conveying the agency’s acronyms “PTO” and “USPTO” as well as the official name “United States Patent and Trademark Office.” If an object does have more than one associated name fact, one of the name facts may be designated as a primary name and other name facts may be designated as secondary names, either implicitly or explicitly.

[0059] A property fact 208 is a fact that conveys a statement about the entity or concept represented by the object ID. Property facts are generally used for summary information about an object. A property fact 208, being a special instance of a general fact 204, also includes the same parameters (such as attribute, value, fact ID, etc.) as other facts 204. The attribute field 226 of a property fact 208 indicates that the fact is a property fact (e.g., attribute is “property”) and the value is a string of text that conveys the statement of interest. For example, for the object ID representing Bill Clinton, the value of a property fact may be the text string “Bill Clinton was the 42nd President of the United States from 1993 to 2001.” “Some object IDs may have one or more associated property facts while other objects may have no associated property facts. It should be appreciated that the data structures shown in FIGS. 2(a)-2(d) and described above are merely exemplary. The data structure of the repository 115 may take on other forms. Other fields may be included in facts and some of the fields described above may be omitted. Additionally, each object ID may have additional special facts aside from name facts and property facts, such as facts conveying a type or category (for example, person, place, movie, actor, organization, etc.) for categorizing the entity or concept represented by the object ID. In some embodiments, an object’s name(s) and/or properties may be represented by special records that have a different format than the general facts records 204.

[0060] As described previously, a collection of facts is associated with an object ID of an object. An object may become a null or empty object when facts are disassociated from the object. A null object can arise in a number of different ways. One type of null object is an object that has had all of its facts (including name facts) removed, leaving no facts associated with its object ID. Another type of null object is an object that has all of its associated facts other than name facts removed, leaving only its name fact(s). Alternatively, the object may be a null object only if all of its associated name facts are removed. A null object repre-

sents an entity or concept for which the data processing system 106 has no factual information and, as far as the data processing system 106 is concerned, does not exist. In some embodiments, facts of a null object may be left in the repository 115, but have their object ID values cleared (or have their importance to a negative value). However, the facts of the null object are treated as if they were removed from the repository 115. In some other embodiments, facts of null objects are physically removed from repository 115.

[0061] FIG. 2(e) is a block diagram illustrating an alternate data structure 290 for facts and objects in accordance with preferred embodiments of the invention. In this data structure, an object 290 contains an object ID 292 and references or points to facts 294. Each fact includes a fact ID 295, an attribute 297, and a value 299. In this embodiment, an object 290 actually exists in memory 107.

[0062] FIG. 3(a) is a block diagram illustrating the extraction of facts from a plurality of documents, according to one embodiment of the present invention. Document 302 and document 308 are analogous to the documents described herein with reference to FIG. 1. According to one embodiment of the present invention, the document 302 and the document 308 are stored in a document repository (not shown).

[0063] The importer 304 processes the document 302 and extracts facts 306. The importer 304 may employ any of a variety of methods for extracting the facts 306 from the document 302, such as one of those described in "Supplementing Search Results with Information of Interest" or in the other incorporated applications. For the purposes of illustration, a single document 302 is shown in the figure. In practice, importer 304 can process a plurality of documents 302 to extract the facts 306.

[0064] According to one embodiment of the present invention, the importer 304 identifies a predefined pattern in the document 302 and applies the predefined pattern to extract attribute-value pairs. The extracted attribute-value pairs are then stored as facts 306. As described in "Supplementing Search Results with Information of Interest", a predefined pattern defines specific, predetermined sections of the document which are expected to contain attributes and values. For example, in an HTML document, the presence of a text block such as "
*. *
" (where "*" can be any string) may indicate that the document contains an attribute-value pair organized according to the pattern "
(attribute text):(value text)
". Such a pattern is predefined in the sense that it is one of a known list of patterns to be identified and applied for extraction in documents. Of course, not every predefined pattern will necessarily be found in every document; identifying the patterns contained in a document determines which (if any) of the predefined patterns may be used for extraction on that document with a reasonable expectation of producing valid attribute-value pairs. The extracted attribute-value pairs are stored in the facts 306.

[0065] An attribute-value pair is composed of an attribute and its associated value. An attribute-value pair may be stored as a fact, for example, by storing the attribute in the attribute field of the fact and the value in the value field of the fact. Extracting a fact is synonymous with extracting at least an attribute-value pair and storing the attribute and value as a fact.

[0066] In the example illustrated, document 302 contains at least some attribute-value pairs organized according to

one of the predefined patterns recognizable by the importer 304. An example of a document containing attribute-value pairs organized according to one of the predefined patterns recognizable by the importer 304 is described herein with reference to FIG. 4. Applying predefined patterns to documents containing attribute-value pairs organized according to those patterns beneficially extracts valuable information without the need for human supervision.

[0067] However, the document 302 may contain other attribute-value pairs organized differently, such that applying one of the predefined patterns recognizable by the importer 304 produces incomplete, inconsistent, or erroneous results. Similarly, a document such as the document 308 may contain attribute-value pairs organized in a manner different from those prescribed by the various predefined patterns. It is possible that, the importer 304 were applied to the document 308, none of the predefined patterns recognizable by the importer 304 would be identified in the document 308.

[0068] Advantageously, one embodiment of the present invention facilitates the extraction of attribute-value pairs organized according to a pattern not itself recognizable by the importer 304. According to one embodiment of the present invention, a janitor 310 receives the facts 306 and the document 308. If the document 308 contains the same (or similar) attribute-value pairs as at least some of the facts 306, the facts 306 may be used to identify a contextual pattern in the document 308. A contextual pattern is a pattern that is inferred on the basis of the context in which known attribute-value pairs appear in a document. An example of a contextual pattern in a document is described herein with reference to FIG. 5. The janitor 310 applies the contextual pattern to the document 308 to extract additional attribute-value pairs. These attribute-value pairs are then stored as the facts 312. Several exemplary methods for identifying a contextual pattern and using it to extract attribute-value pairs are described in "Learning Facts from Semi-Structured Text."

[0069] According to one embodiment of the present invention the janitor 310 additionally corroborates the facts 306 using a corroborating document (not shown). For example, as a result of improperly applied predefined patterns (or the document 302 itself), some of the facts 306 may contain errors, inconsistent information, or other factual anomalies. If the attribute-value pair of the fact 306A cannot be found in any corroborating document, the janitor 310 may reduce the confidence score of the fact 306A. Alternatively, if the attribute-value pair of the fact 306A is identified in a corroborating document, the confidence score of the fact 306A can be increased, and a reference to the corroborating document can be added to the list of sources for that fact. Several exemplary methods for corroborating facts can be found in "Corroborating Facts Extracted from Multiple Sources."

[0070] According to one embodiment of the present invention, a plurality of documents are used to import and corroborate a group of facts. From this group of imported facts, those associated with a common name may be aggregated to form the facts 306. The facts 306 may be normalized, merged and/or corroborated, and their confidence score may be adjusted accordingly (for example, by the janitor 310, or by another janitor). According to one embodiment of

the present invention, only facts 306 having a confidence score above a threshold are used for identification of contextual patterns by the janitor 310. Corroborating facts beneficially improves the consistency of extracted facts, and can reduce the influence of improperly applied predefined patterns on the quality of the fact database.

[0071] The facts 306 and facts 312 may be associated with a common object. For example, the facts 306 may be extracted from the document 302 and stored as an object in an object repository. According to one embodiment of the present invention, the facts 306 may be associated with an object name. An exemplary method for associating an object name with an object is described in "Identifying a Unifying Subject of a Set of Facts". According to one embodiment of the present invention, the object name (or another property associated with the facts 302) are used to retrieve the document 308. Using the object name to retrieve the document 308 is one example of a method for finding a document potentially containing attribute-value pairs common with the document 302. As another example, the corroboration janitor 306 could query a search engine for documents containing one of the attribute-value pairs of the facts 306. Other methods will be apparent to one of skill in the art without departing from the scope of the present invention.

[0072] According to one embodiment of the present invention, the facts 312 are further processed by a janitor (either the janitor 310 or another janitor). For example, the facts 312 can be merged with another set of facts (for example, the facts 306), normalized, corroborated, and/or given a confidence score. According to one embodiment of the present invention, facts 312 having a confidence score above a threshold are added to a fact repository.

[0073] FIG. 3(b) is a block diagram illustrating the extraction of facts from a plurality of documents to produce an object, according to one embodiment of the present invention. The documents 313 contain at least one attribute-value pair in common, although this attribute-value pair may be organized according to different patterns in the various documents. Document 313A and document 313B may or may not describe a common subject.

[0074] The unsupervised fact extractor 314 identifies in document 313A a predefined pattern and applies that pattern to extract a "seed" attribute-value pair. The unsupervised fact extractor 314 uses the seed attribute-value pair to identify a contextual pattern, in either or both of the documents 313, and applies the contextual pattern to extract additional attribute-value pairs. A method used by the unsupervised fact extractor 314, according to one embodiment of the present invention, is described herein with reference to FIG. 6. The unsupervised fact extractor 314 may be composed of any number of sub-components, for example, the importer 304 and janitor 310 described herein with reference to FIG. 3(a).

[0075] The unsupervised fact extractor 314 organizes the extracted attribute-value pairs into an object 316. The unsupervised fact extractor 314 may also employ techniques for normalization, corroboration, confidence rating, and others such as those described in the applications incorporated by reference above. Other methods for processing the extracted facts to produce an object will be apparent to one of skill in the art without departing from the scope of the present invention. Furthermore, the unsupervised fact extractor 314

has been shown as receiving two documents and producing one object for the purposes of illustration only. In practice, the unsupervised fact extractor 314 may operate on any number of documents, to extract a plurality of facts to be organized into any number of objects.

[0076] By identifying both predefined and contextual patterns in the documents 313, the unsupervised fact extractor 314 is able to build objects containing more information than extractors relying on predefined patterns alone, and without the need for document-specific human tailoring or intervention.

[0077] FIG. 4 is an example of a document containing attribute-value pairs organized according to a predefined pattern. According to one embodiment of the present invention, document 402 may be analogous to the document 302 described herein with reference to FIG. 3. Document 402 includes information about Britney Spears organized according to a two column table 404. According to one embodiment of the present invention, the two column table is a predefined pattern recognizable by the unsupervised fact extractor 314. The pattern specifies that attributes will be in the left column and that corresponding values will be in the right column. Thus the unsupervised fact extractor 314 may extract from the document 402 the following attribute-value pairs using the predefined pattern: (name; Britney Spears), (profession; actress, singer), (date of birth; Dec. 2, 1981), (place of birth; Kentwood, La.), (sign; Sagittarius), (eye color; brown), and (hair color; brown). These attribute-value pairs can then be stored as facts, associated with an object, used as seed attribute-value pairs, and so on.

[0078] FIG. 5 is an example of a document 502 from which a contextual pattern can be identified using a seed fact, according to one embodiment of the present invention. Document 502 may be analogous to the document 308 described herein with reference to FIG. 3. Document 502 includes information about Britney Spears. Document 502 illustrates a list 504 organized according to a pattern that for the purposes of illustration could be considered whimsical. Attributes are in bold, and values associated with those attributes are listed immediately below in italics. Such a pattern might be intuitive to a human user, but if that particular pattern is not recognizable to an extractor as a predefined pattern, using predefined patterns exclusively could result in the incorrect or failed extraction of the attribute-value pairs.

[0079] However, the document 502 has several attribute-value pairs in common with the document 402. Specifically, the (name; Britney Spears) and (date of birth; Dec. 2, 1981) pairs are contained in both documents. The unsupervised fact extractor 314 can use one (or both) of these pairs as a seed attribute-value pair to identify a contextual pattern of other attribute-value pairs. For example, the (name; Britney Spears) pair might be contained in a context such as the following:

[0080]
Name
<I>Britney Spears</I>

[0081] Thus, using the information extracted from the document 402, the unsupervised fact extractor 314 might identify in document 502 a contextual pattern for attribute-value pairs organized as:

[0082]
(attribute)
<I>(value)</I>

[0083] The common pair comprised of (date of birth; Dec. 2, 1981) may be used to confirm this contextual pattern, since this pair might also be contained in a context such as:

[0084]
Date of Birth
<I>Dec. 2, 1981</I>

[0085] Once the unsupervised fact extractor 314 has identified a contextual pattern, the unsupervised fact extractor 314 uses the contextual pattern to extract additional facts from the document 502. Thus the unsupervised fact extractor 314 may extract from the document 502 the following attribute-value pairs using the predefined pattern: (Favorite Food; Chicken Parmesan), (Favorite Movie; Back to the Future), and (Profession; Singer-Songwriter).

[0086] For the purposes of illustration, the document 502 shows attribute-value pairs organized according to a single contextual pattern. Documents may contain multiple and various contextual patterns, or a mix of predefined patterns and contextual patterns. Furthermore, the examples of predefined patterns and contextual patterns illustrated herein as been selected for the purposes of illustration only. In some cases the attribute-value pattern used by document 502 may be recognizable as a predefined pattern, and conversely, in some cases the attribute-value pattern used by document 402 may not be recognizable as a predefined pattern. Given the scope and diversity of the internet, however, there will always be some documents containing attribute-value pairs not organized by a recognizable predefined pattern, and the ability to identify contextual patterns beneficially facilitates the extraction of at least some of these pairs.

[0087] FIG. 6 is a flow chart illustrating a method for extracting facts, according to one embodiment of the present invention. According to one embodiment of the present invention, the method is performed by the unsupervised fact extractor 314.

[0088] The method begins with a document 302. The document 302 contains an attribute-value pair organized according to a predefined pattern. The unsupervised fact extractor 314 extracts 604 an attribute-value pair from the document 302, producing a seed attribute-value pair 606. According to one embodiment of the present invention, the unsupervised fact extractor 314 can extract 604 the attribute and value from the document by applying a predefined pattern; other methods for extracting 604 the attribute and value will be apparent to one of skill in the art without departing from the scope of the present invention. Additionally, the unsupervised fact extractor 314 may store the seed attribute-value pair 606 in a fact (not shown). According to one embodiment of the present invention, the fact in which the seed attribute-value pair 606 is stored is associated with an object.

[0089] The unsupervised fact extractor 314 retrieves 608 a document 610 that contains the seed attribute-value pair 606 organized according to a contextual pattern.

[0090] According to one embodiment of the present invention, the unsupervised fact extractor 314 retrieves 608 the document 610 by searching (for example, on document hosts or in a document repository) for documents containing the attribute and value of the seed attribute-value pair. According to another embodiment of the present invention,

the seed attribute-value pair is stored as a fact associated with an object. This object may have a name, and the unsupervised fact extractor 314 may retrieve 608 a document 610 by searching in a document repository for documents containing the object name. Other methods for retrieving 608 a document 610 will be apparent to one of skill in the art without departing from the scope of the present invention.

[0091] The unsupervised fact extractor 314 identifies 612 a contextual pattern associated with the seed attribute-value pair 606 and uses the pattern to extract an attribute-value pair 614 from the document 610. The attribute-value pair 614 may then be stored as a fact and processed by further janitors, importers, and object retrievers as appropriate. According to one embodiment of the present invention, the fact in which the attribute-value pair 614 is stored is associated with an object. The fact containing attribute-value pair 614 may be associated with the same object as the fact containing seed attribute-value pair 606, or it may be associated with a different object.

[0092] By extracting attributes and value using both predefined and contextual patterns, the unsupervised fact extractor 314 is able to collect a larger amount of information into facts than an extractor relying on either approach alone. Advantageously, information may be extracted into facts efficiently, accurately, and without need for human supervision.

[0093] Additionally, the unsupervised fact extractor 314 may also use the contextual pattern to extract another attribute-value pair from a third document. According to one embodiment of the present invention, the unsupervised fact extractor 314 determines if the third document is similar to the document 610, for example, by comparing the domain hosting the document 610 to the domain hosting the third document. Using the contextual pattern to extract another attribute-value pair from a third document may be responsive to the determination that the third document is similar to the document 610. Using the contextual pattern to extract another attribute-value pair from a third document advantageously facilitates the extracting of attribute-value pairs organized according to patterns not recognizable as predefined patterns, even from documents not containing a seed attribute-value pair.

[0094] While a method for extracting facts has been shown for the purposes of illustration as extracting a single seed attribute-value pair 606 and a single attribute-value pair 614, it will be apparent to one of skill in the art that in practice the unsupervised fact extractor 314 may extract 604 a plurality of attribute-value pairs and extract 612 a plurality of attribute-value pairs 614. When a plurality of attribute-value pairs are extracted 604, any number of that plurality may be used as seed attribute-value pairs 606. According to one embodiment of the present invention, extracting 612 additional attribute-value pairs from the document 610 is responsive to the number of seed-attribute-value pairs 606 contained in the document 610. According to another embodiment of the present invention, a first seed attribute-value pair 606 may be used to identify 612 a contextual pattern and a second seed attribute-value pair 606 may be used to verify that contextual pattern, for example, by determining if the second seed attribute-value pair 606 is organized in the document 610 according to the contextual

pattern. By using a plurality of seed attribute-value pairs 606, the efficiency and accuracy of the unsupervised fact extractor 314 may be improved.

[0095] Reference in the specification to “one embodiment” or to “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

[0096] Some portions of the above are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps (instructions) leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic or optical signals capable of being stored, transferred, combined, compared and otherwise manipulated. It is convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. Furthermore, it is also convenient at times, to refer to certain arrangements of steps requiring physical manipulations of physical quantities as modules or code devices, without loss of generality.

[0097] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as “processing” or “computing” or “calculating” or “determining” or “displaying” or “determining” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0098] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention can be embodied in software, firmware or hardware, and when embodied in software, can be downloaded to reside on and be operated from different platforms used by a variety of operating systems.

[0099] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs,

EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

[0100] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references below to specific languages are provided for disclosure of enablement and best mode of the present invention.

[0101] While the invention has been particularly shown and described with reference to a preferred embodiment and several alternate embodiments, it will be understood by persons skilled in the relevant art that various changes in form and details can be made therein without departing from the spirit and scope of the invention.

[0102] Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

What is claimed is:

1. A method for extracting facts, the method comprising:
 - extracting a first fact having an attribute and a value from a first document;
 - retrieving a second document that contains the attribute and the value of the first fact;
 - identifying in the second document a contextual pattern associated with the attribute and value of the first fact; and
 - extracting a second fact from the second document using the contextual pattern.
2. The method of claim 1, further comprising:
 - extracting a third fact from a third document using the contextual pattern.
3. The method of claim 2, wherein the second document is hosted on a first domain, and wherein the third document is hosted on the first domain.
4. The method of claim 1, further comprising:
 - associating the first fact with a first object.
5. The method of claim 4, wherein the first object is associated with an object name, and wherein retrieving the second document comprises searching a repository of documents for a document containing the object name.

6. The method of claim 4, further comprising associating the second fact with the first object.

7. The method of claim 1, wherein extracting a first fact from a first document comprises:

extracting a first plurality of facts from the first document, each fact having an attribute and a value.

8. The method of claim 7, wherein identifying in the second document a contextual pattern associated with the attribute and value of the first fact comprises:

identifying in the second document a contextual pattern associated with the attributes and the values of a number of the first plurality of facts.

9. The method of claim 8, wherein said extracting said second fact is responsive to the number of the first plurality of facts having attributes and values associated with the contextual pattern.

10. The method of claim 7, wherein said first plurality of facts includes a third fact, the method further comprising:

determining if the third fact is organized in the second document according to the contextual pattern.

11. The method of claim 1, further comprising:

identifying a predefined pattern in the first document, and wherein said extracting said first fact comprises extracting said first fact from the first document using the predefined pattern.

12. The method of claim 1, wherein said first document is different from said second document.

13. The method of claim 1, wherein retrieving the second document comprises querying a search engine for a document containing the attribute and the value of the first fact.

14. A system for extracting facts comprising:

an importer, configured to extract a first fact from a first document; and

a janitor, configured to receive said first fact, identify a contextual pattern of said first fact in a second document, and extract a second fact from the second document using the contextual pattern.

15. The system of claim 14, further comprising:

a document repository, wherein the first document and the second document are stored in said document repository.

16. The system of claim 14, further comprising:

a fact repository configured to store the first fact and the second fact.

17. A computer program product, the computer program product comprising a computer-readable medium, for extracting facts, the computer-readable medium comprising:

program code for extracting a first fact having an attribute and a value from a first document;

program code for retrieving a second document that contains the attribute and the value of the first fact;

program code for identifying in the second document a contextual pattern associated with the attribute and value of the first fact; and

program code for extracting a second fact from the second document using the contextual pattern.

18. The computer program product of claim 17, the computer-readable medium further comprising:

program code for extracting a third fact from a third document using the contextual pattern.

19. The computer program product of claim 17, the computer-readable medium further comprising:

program code for associating the first fact with a first object.

20. The computer program product of claim 19, wherein the first object is associated with an object name, and wherein the program code for retrieving the second document comprises program code for searching a repository of documents for a document containing the object name.

21. The computer program product of claim 19, further comprising program code for associating the second fact with the first object.

22. The computer program product of claim 17, wherein the computer code for retrieving the second document comprises computer code for querying a search engine for a document containing the attribute and the value of the first fact.

23. A method for extracting facts, the method comprising:

extracting a first fact having an attribute and a value from a first document;

retrieving a second document;

if the second document corroborates the first fact:

retrieving a third document that contains the attribute and the value of the first fact;

identifying in the third document a contextual pattern associated with the attribute and value of the first fact; and

extracting a second fact from the third document using the contextual pattern.

24. The method of claim 24, further comprising:

retrieving a fourth document;

if the fourth document corroborates the second fact, storing the second fact in a fact repository.

* * * * *