US 20090063470A1

(19) **United States**

(12) **Patent Application Publication**

Peled et al.

(10) **Pub. No.: US 2009/0063470 A1**

(43) **Pub. Date: Mar. 5, 2009**

(54) **DOCUMENT MANAGEMENT USING BUSINESS OBJECTS**

(75) Inventors: **Ariel Peled**, Raanana (IL); **Gilad Savion**, Tel Aviv (IL); **Elad Reznikov**, Tel Aviv (IL); **Yizhar Regev**, Ramat Gan (IL); **Izhak Shmulewitz**, Tel Aviv (IL)

Correspondence Address:
DARBY & DARBY P.C.
P.O. BOX 770, Church Street Station
New York, NY 10008-0770 (US)

(73) Assignee: **NOGACOM LTD.**, Tel Aviv (IL)

(21) Appl. No.: **12/199,043**

(22) Filed: **Aug. 27, 2008**

Related U.S. Application Data

(60) Provisional application No. 60/968,329, filed on Aug. 28, 2007.

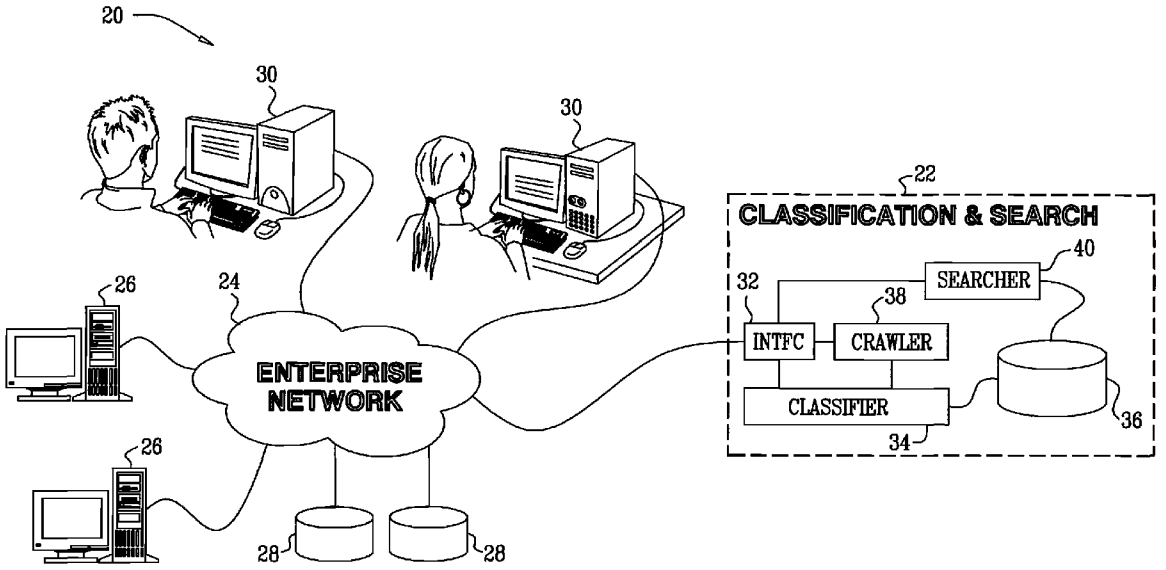
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/5**; 707/E17.014

(57) **ABSTRACT**

A computer-implemented method for processing information includes collecting data objects from one or more data repositories, the data objects having respective properties, which identify the data objects. The properties of the collected data objects are analyzed in order to derive respective identifiers corresponding to the data objects. A text string that matches one of the identifiers of a data object is identified within a context in a document. Responsively to the context, an indication that the identified text string is a valid instance of the data object is generated, and the document is processed responsively to the indication.



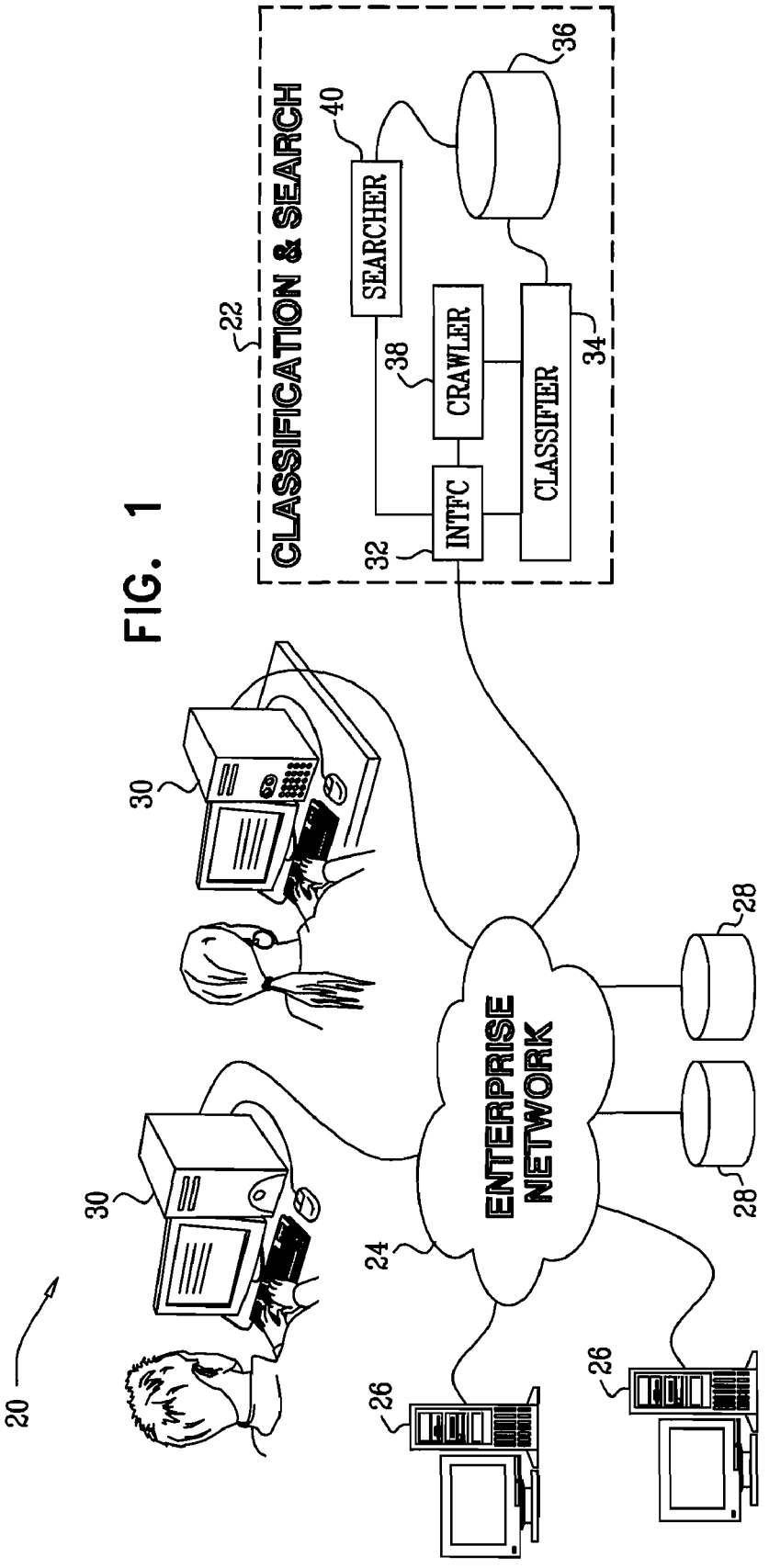

20
30
30
ENTERPRISE NETWORK
24
26
26
28
28
CLASSIFICATION & SEARCH
22
INTFC
32
CRAWLER
38
SEARCHER
40
CLASSIFIER
34
36

FIG. 1

FIG. 2

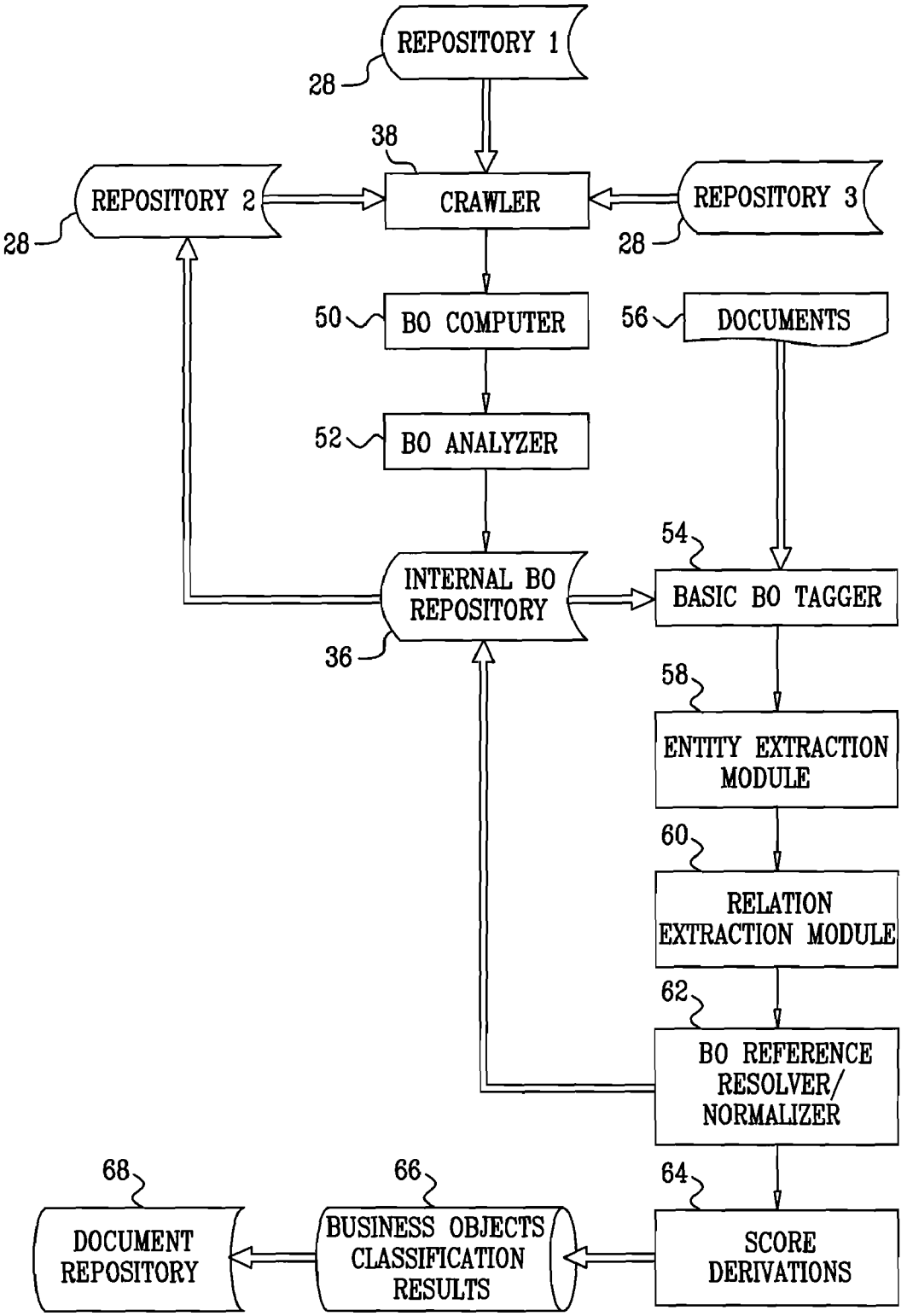

REPOSITORY 1
28
38
REPOSITORY 2
28
CRAWLER
REPOSITORY 3
28
50
BO COMPUTER
56
DOCUMENTS
52
BO ANALYZER
54
INTERNAL BO REPOSITORY
36
BASIC BO TAGGER
58
ENTITY EXTRACTION MODULE
60
RELATION EXTRACTION MODULE
62
BO REFERENCE RESOLVER/ NORMALIZER
68
DOCUMENT REPOSITORY
66
BUSINESS OBJECTS CLASSIFICATION RESULTS
64
SCORE DERIVATIONS

FIG. 3

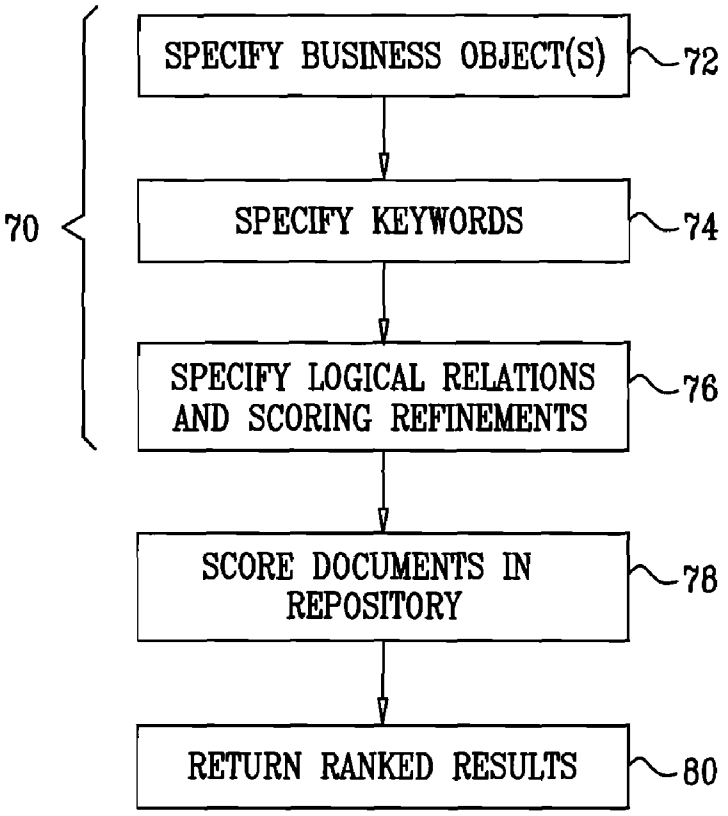

SPECIFY BUSINESS OBJECT(S)
72
70
SPECIFY KEYWORDS
74
SPECIFY LOGICAL RELATIONS AND SCORING REFINEMENTS
76
SCORE DOCUMENTS IN REPOSITORY
78
RETURN RANKED RESULTS
80

FIG. 4

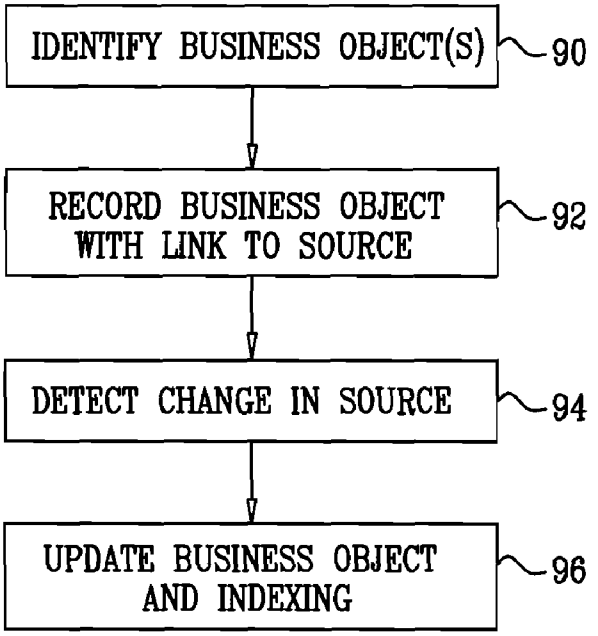
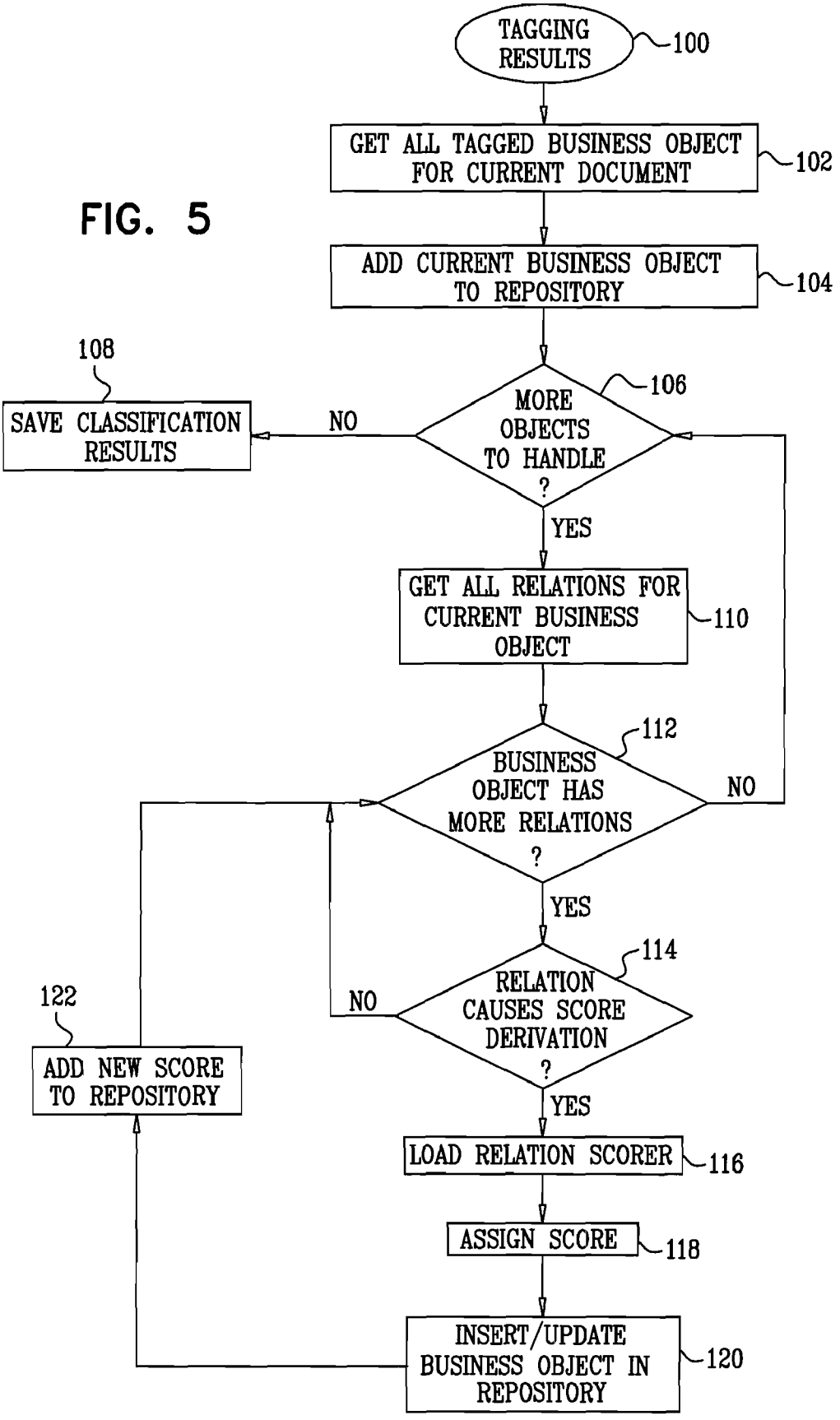

IDENTIFY BUSINESS OBJECT(S)
90
RECORD BUSINESS OBJECT WITH LINK TO SOURCE
92
DETECT CHANGE IN SOURCE
94
UPDATE BUSINESS OBJECT AND INDEXING
96

FIG. 5


TAGGING RESULTS
100
GET ALL TAGGED BUSINESS OBJECT FOR CURRENT DOCUMENT
102
ADD CURRENT BUSINESS OBJECT TO REPOSITORY
104
MORE OBJECTS TO HANDLE ?
106
NO
SAVE CLASSIFICATION RESULTS
108
YES
GET ALL RELATIONS FOR CURRENT BUSINESS OBJECT
110
BUSINESS OBJECT HAS MORE RELATIONS ?
112
NO
YES
RELATION CAUSES SCORE DERIVATION ?
114
NO
YES
ADD NEW SCORE TO REPOSITORY
122
LOAD RELATION SCORER
116
ASSIGN SCORE
118
INSERT/UPDATE BUSINESS OBJECT IN REPOSITORY
120

DOCUMENT MANAGEMENT USING BUSINESS OBJECTS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application 60/968,329, filed Aug. 28, 2007, whose disclosure is incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The present invention relates generally to information processing, and specifically to methods and systems for indexing and searching documents.

BACKGROUND OF THE INVENTION

[0003] Organizations, such as business enterprises, typically accumulate vast amounts of data, including both structured data, such as database and spreadsheet records, and unstructured data, in the form of natural language text (also referred to as "free text"). Structured data can be efficiently indexed, addressed and searched using well-known tools, such as structured query language (SQL). Search tools for natural language documents, however, are limited for the most part to keyword-based techniques. As a result, searching a corpus of textual documents for a particular occurrence of a certain data object is frequently inefficient and time-consuming and may miss relevant occurrences of an object of interest, such as a person, company or product.

SUMMARY OF THE INVENTION

[0004] Embodiments of the present invention provide improved methods and systems for analyzing a set of data objects in a data repository of an organization, and using these data objects in tagging, classifying and then searching a corpus of data.

[0005] There is therefore provided, in accordance with an embodiment of the present invention, a computer-implemented method for processing information includes collecting data objects from one or more data repositories, the data objects having respective properties, which identify the data objects. The properties of the collected data objects are analyzed in order to derive respective identifiers corresponding to the data objects. A text string that matches one of the identifiers of a data object is identified within a context in a document. Responsively to the context, an indication that the identified text string is a valid instance of the data object is generated, and the document is processed responsively to the indication.

[0006] In another embodiment of the present invention, a computer-implemented method for processing information includes collecting data objects from one or more data repositories and identifying a respective record in the repositories corresponding to each of the data objects. One or more documents are processed so as to generate a listing of occurrences of the data objects in the documents. Upon detecting a change in the respective record corresponding to one of the data objects, the listing is automatically updated with respect to the one of the data objects. The documents are processed responsively to the listing.

[0007] Other embodiments of the present invention provide apparatus and computer software products for processing information.

[0008] The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. **1** is a block diagram that schematically illustrates a system for exchange and management of data, in accordance with an embodiment of the present invention;

[0010] FIG. **2** is a flow chart that schematically illustrates a method for classifying and tagging documents according to business objects and indexing the documents according to the tagging results, in accordance with an embodiment of the present invention;

[0011] FIG. **3** is a flow chart that schematically illustrates a method for searching a set of documents, in accordance with an embodiment of the present invention;

[0012] FIG. **4** is a flow chart that schematically illustrates a method for updating a business object, in accordance with an embodiment of the present invention; and

[0013] FIG. **5** is a flow chart that schematically illustrates a method for scoring business objects, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

Overview

[0014] Embodiments of the present invention that are described hereinbelow provide apparatus, methods and software for document and knowledge management within an organization. The methods focus on analyzing a set of data objects in the organization's data repositories, which are then used in tagging, classifying, indexing and searching a corpus of documents that is maintained by the organization. In a business, for example, the data objects may refer to entities of importance to the business, such as employees, products, customers and suppliers of the business, departments or units of the organization, or geographical areas. For convenience and clarity, the description that follows will relate to these sorts of data objects, which will be referred to hereinafter as "business objects." The principles of the present invention, however, are similarly applicable to organizations and data objects of other types.

[0015] Each business object is identified by properties that typically include one or more names, which may comprise multiple words. In addition, each business object typically has additional properties, such as synonyms, e-mail address, physical address, job title with organization name and/or affiliation, telephone number, and ID numbers, which may be useful in identifying occurrences of the business object in documents. There may also be links and dependencies between the business objects. (For example, a person business object might be an employee of an organization business object, such as a customer or supplier.) Business objects are also dynamic, and their properties may change during their life cycle.

[0016] Some business objects properties may be numerical or fixed strings, but many properties, such as business object names, are open, literal, natural language strings, and thus are more complex and error-prone. In addition, people tend to automatically shorten natural language names and/or to create from them synonyms and other variants, according to various lexical and semantic rules. Frequently the different names or parts of a name of a given business object may be used separately and independently. For example, business

objects may be referred to in documents by partial names, such as the first name or nickname of a person, an abbreviation of a product, or an organization name without the usual prefix or suffix.

[0017] Thus, in the context of a document (and particularly a natural language document), it may not be clear whether a certain text fragment, such as a word, that is known to be related to the name of a given business object actually represents an instance of the business object. In some cases a text string may be found in a document that matches a name or a variant of a name of a business object, while the actual semantic meaning of the string in the specific context of the document does not refer to the business object. (For example, the phrase "nice systems," without further information, could either refer to the company NICE Systems or describe the qualities of certain products unrelated to the particular company.) Some embodiments of the present invention address this problem by using natural language processing to ascertain automatically, based on the context (and without human intervention in most cases), whether or not the text string in question actually refers to a certain data object.

[0018] In embodiments of the present invention, business objects are identified automatically by processing data repositories of the organization. Typically, the business objects are identified in sources of structured data, such as databases, CRM (Customer Relation Management) systems, or other similar organizational systems and spreadsheets. In some embodiments, business objects are also extracted from documents containing unstructured data, in which case the "record" with which the business object is associated is the document or portion of the document from which the business object was extracted. Furthermore, in many organizations, different types of business objects may be managed in different systems, which may include duplicates and errors. Therefore, in the disclosed embodiments, data are collected and compared from various repositories, and are then analyzed to create a unified listing of the business objects across the organization. This analysis uses natural language tools, such as lexical, linguistic and semantic analysis, to find identifiers, including variants that are different from the actual names of the business objects, that may identify the business objects in a document. As explained in detail hereinbelow, these variants may be based either on the object names or on other object properties.

[0019] For purposes of management and updating of the centralized listing of business objects, each business object may be associated with one or more corresponding source records in the structured data. (A business object may be present in several organizational systems. For example, an employee may have a record both in Windows Active Directory and in the organization HR system. For each business object, each such source record (with its corresponding ID) is listed. When a change occurs in a source record or when a new source record is found to correspond to an existing business object, the business object is then updated accordingly, without any need for human intervention. If required, the relevant documents are re-tagged, so that subsequent searches use the most up-to-date information regarding all the business objects in the set.

System Description

[0020] FIG. **1** is a block diagram that schematically illustrates a system **20** for exchange and management of data, in accordance with an embodiment of the present invention. System **20** is typically maintained by an organization, such as a business, for purposes of exchanging, storing and recalling data used by the organization. A data classification and search server **22** identifies business objects and builds a listing, such as an index, for use in searching the data, as described in detail hereinbelow.

[0021] System **20** is typically built around an enterprise network **24**, which may comprise any suitable type or types of data communication network, and may, for example, include both intranet and extranet segments. A variety of servers **26** may be connected to the network, including mail and other application servers, for instance. Storage repositories **28** are also connected to the network and typically contain both structured and unstructured data. The structured data may include a variety of databases, such as product databases, human resources (HR) databases containing records of personnel of the organization, and customer relations management (CRM) databases containing records of customers of the organization, as well as their orders and payment records. Additionally or alternatively, structured data may be organized and stored in other forms and formats that are known in the art, such as spreadsheets. Servers **26** and repositories **28** are accessible to client computers **30** via network **24**.

[0022] Server **22** connects to network **24** via a suitable network interface **32**. The server typically comprises one or more general-purpose computer processors, which are programmed in software to carry out the functions that are described herein. This software may be downloaded to server **22** in electronic form, over a network, for example. Alternatively or additionally, the software may be provided on tangible storage media, such as optical, magnetic or electronic memory media. Although server **22** is shown in FIG. **1**, for the sake of simplicity, as a single unit, in practice the functions of the server may be carried out by a number of different processors, such as a separate processor (or even a separate computer) for each of the functional blocks shown in the figure. Alternatively, the functional blocks may be implemented simply as different processes running on the same computer. All such alternative configurations are considered to be within the scope of the present invention.

[0023] Server **22** comprises a classifier **34**, which automatically assembles a listing of business objects based on information in repositories **28** (and possibly other sources, as well), and then tags the documents in system **20** according to instances of the business objects that occur in the documents. The classifier recognizes and resolves variant forms of the business object names, such as shortened names and abbreviations, using techniques of natural language processing, and may assign confidence scores to instances of the business objects depending on the level of certainty that a given variant actually refers to the business object in question. A crawler **38** collects documents from system **20**, and classifier **34** builds an index of the documents, for use in subsequent search and update operations, according to occurrences of the business objects in the document text. Classifier **34** stores the business object listing and index in an internal repository **36**, which typically comprises a suitable storage device or group of such devices. Details of the processes of identifying data objects and tagging documents are described further hereinbelow with reference to FIG. **2**.

[0024] In addition, classifier **34** may create a general index of strings appearing in the documents, for purposes of subsequent keyword-based searching, as is known in the art.

[0025] A searcher **40** receives requests, typically from client computers **30**, to search the documents in system **20** for a certain business object or combination or type of business objects. The search queries may also specify keywords, in addition to the business objects, as well as logical operators connecting the business objects and (optional) keywords in the queries. The searcher extracts documents from system **20** that contain instances of the business objects specified by the query and scores each document according to factors such as the number of occurrences of the business objects and the confidence level. The score may also reflect occurrences of specified keywords in the documents, as well as factors such as document type and metadata. Searcher **40** ranks the documents according to their scores and returns the result to the requesting client. Details of the search process are described hereinbelow with reference to FIG. **3**.

Business Object Tagging and Search

[0026] FIG. **2** is a flow chart that schematically illustrates a method for classifying and tagging a set of documents according to a business object set, in accordance with an embodiment of the present invention. The method is described, for the sake of clarity, with reference to the system architecture shown in FIG. **1**, but the principles of this method may similarly be applied in tagging and indexing of data objects in other applications. Examples of some of the functions shown in FIG. **2** are described below in the Appendices.

[0027] To begin the process, crawler **38** loads business objects to classifier **34** from repositories **28** of system **20**. These repositories may include, for example, records maintained by applications such as a CRM system and a HR system, as well as computer system management applications, such as Microsoft Active Directory. Such applications typically have an application program interface (API), which the crawler can use to access the tables of business objects and their properties. Classifier **34** uses the information provided by the crawler to build a table of each type of business objects, such as customers, employees, products, etc. The crawler continually samples repositories **28** in order to report changes in the business object listings.

[0028] In some embodiments, crawler **38** also loads and analyzes, for each business object, permission and control access details, specifying which users are allowed to view the business object and its details and which users are allowed to change these points. The crawler converts the access list into a standard Access Control List (ACL) form and saves the ACL in the business object repository.

[0029] Classifier **34** may also identify new business objects in unstructured documents, as described below in reference to step **58**. This identification is typically based on morphological, syntactic and semantic analysis of the document using appropriate rules.

[0030] When crawler **38** delivers a new business object from one of repositories **28**, classifier **34** activates a business object (BO) comparer function **50** to compare the new business object to the business objects that are already listed in repository **36**. The comparer function calculates a similarity factor between the new business object and each of the existing business objects. If the factor is above a high threshold, the classifier will treat the two business objects as identical, i.e., as alternative names of the same object. The classifier will then merge the record of the new business object into the record of the existing business object that it matched. Some examples of rules that may be used in calculating and applying similarity factors are presented in Appendix A below.

[0031] On the other hand, even when the similarity factor is not high enough to support a conclusion that the new object is identical to an existing object, there may still be a similarity relation between the objects if the factor is above a certain lower threshold. In this case, the classifier adds the new business object to the list in repository **36** and records a similarity relation between the new and existing business objects. This similarity relation is used subsequently in tagging and scoring occurrences of the business objects in documents, as described hereinbelow.

[0032] The comparer function may also discover and record other relations between business objects. For example, it may find that two employees share a telephone number, or that two organizations share a domain name. These relations may also be used in tagging and scoring, and may in addition be queried directly by clients.

[0033] Various methods and considerations may be used in computing similarity factors, and the scoring formula may vary depending on the type of business object involved. For example, if two business objects of type "person" have the same social security number, they can be assumed to be one and the same. If two customers have identical postal addresses, they may be considered to be the same business objects, although if they share only the same city, street and building number, they may receive a lower similarity factor.

[0034] A business object analyzer function **52** of classifier **34** uses the information provided by crawler **38** and comparer function **50** in building, for each business object, a set of identifiers, including variants, that will serve as the basis for tagging instances of the business object in the documents in system **20**. Each business object is typically identified by a name and appropriate additional properties, such as ID number, telephone number, e-mail address, etc. A listing of representative properties for different business object types is presented below in Appendix B. To generate the possible variants, the analyzer parses the name and other properties in order to create the set of partial names, synonyms and acronyms that may refer to instances of the business object in the documents in system **20**. The name and properties may be specified separately in different languages if necessary, and the analyzer may automatically identify the language as part of the parsing process.

[0035] Further aspects of the operation of the business object analyzer function are described below in the section headed "Business Object Analysis and Validation."

[0036] Classifier **34** stores the listings of business objects and their properties in internal repository **36**, as noted above. These listings are typically not static, but rather are updated continually in response to changes occurring in the records and other documents in repositories **28**. A method for updating business objects is described hereinbelow with reference to FIG. **4**.

[0037] Classifier **34** applies the business object listings described above in tagging instances of business objects that occur in documents **56**, which are collected by crawler **38** from system **20**. A basic tagger **54** loads the list of business objects from repository **36**, including all the possible variants, and searches each document for the patterns corresponding to the business object name and variants. Tagger **54** may also use other lexicons of relevant terms, such as common first names and common organizational suffixes (such as "corp."), in addition to the business object names, as well as vocabularies

and/or regular expressions. Tagger **54** typically analyzes the tokens (such as words of natural language text) appearing in each document both typographically and morphologically for similarity to the names that are to be tagged. The list of possible variants of a particular business object that is to be used for this purpose may be adjusted according to the language of the document that is to be tagged.

[0038] The tagger also checks the context of each business object name or variant that is found in the document to make sure that the reference is valid. For example, before identifying an occurrence of the name "Pandora" in a document as referring to a customer by this name, the tagger checks to ensure that "Pandora" is not part of another name, such as of a person named "Pandora Smith."

[0039] The tagger tags each name that may be an instance of a given business object both with the business object name and with a confidence score. For this purpose, the document is converted to text and then tokenized, i.e., separated into single words. For each token, the relevant features are saved, such as typographical features (alphabetic token or numerical, capitalized or not, etc.) and part of speech (proper noun, noun, verb, etc.) Each token may also be compared to relevant lexicons, as noted above. Typically, full names receive higher scores than partial or abbreviated names, and the score may be increased or decreased based on the nature and number of variants of the business object in question that appear in the document being tagged.

[0040] As noted above, classifier **34** may encounter business objects in documents **56** that are not included in the listings in repository **36**. To deal with such objects, as well as other object-related entities, the classifier invokes an entity extraction module **58**. This module applies rule-based natural language processing to identify and extract business entities such as persons and organizations, as well as ancillary data entities, such as locations, dates, telephone numbers, etc., which may refer to business objects. The classifier may use the extracted entities to support identification of existing business objects or may add new business objects to the listings in repository **36** based on the extracted entities, either automatically or interactively with the support of a system manager, for example.

[0041] Classifier **34** also actuates a relation extraction module **60** in order to identify relations between business objects (or other entities) and other entities or properties appearing in documents **56**. This module may, for example, extract relations such as company location (or headquarters), which identifies the relation between a company business object and a place; or affiliation/employment, which identifies the organization at which a person business object is employed and his position in the organization.

[0042] After the business objects in a given document have been tagged, and entities and relations have been extracted, a resolver function **62** determines which business objects are actually referenced in the document. The resolver is typically invoked to resolve ambiguities, which may occur when a given string may refer to more than one business object (as when two persons have the same name), or when it is not certain that a name extracted from the document actually matches a business object that it resembles. The resolver computes a score for each business object to which the ambiguous entity might refer. The score may be based, for example, on how fully a partial name in the entity matches the full name of the business object or on other information appearing in the document that may be more relevant to one business object or the other. Typically, the resolver chooses to tag the ambiguous string as an instance of the business object with the higher score.

[0043] After tagging of entities in the text of document **56** is completed, classifier **34** may apply score derivations **64** in order to add relevance tags to the document for other business objects that do not occur explicitly in the document. For this purpose, the classifier typically computes relevance scores of other business objects that are related to the business objects occurring in the document. Relations that may be used in score derivation include, for instance, similarity (as explained above), container relations (one entity contains another), hierarchical relations, and affinity relations (such as the affinity between a customer and an invoice issued to the customer).

[0044] For example, if the classifier has found that a given document refers to a person who works in the finance department of the business, it may give the document a certain relevance score with respect to the finance department business object, even if the finance department is not mentioned in the document. Typically, the related business object, such as the finance department in this example, will receive a lower relevance score than the actual business object in the document. In the present example of a container relation, the derived score that is assigned to the finance department may drop in inverse proportion to the size of the department.

[0045] Classifier **34** classifies each document **56** according to the business objects that it has tagged in and with respect to the document, and stores the results in a classification repository **66** (which may be part of repository **36**). The classification results may be organized in an inverted index of business objects for use in subsequent searching. The tagged document itself may be stored in a document repository **68** (which may also be part of repository **36**). Rather than storing the entire document, however, it may be sufficient for the classifier to store document metadata, containing the tag information for the document and pointing to the location of the document in system **20**. Each instance is thus saved and later retrieved by the document ID of the document in which it was found and the character offset (i.e., the index of the character within the document at which the instance begins).

[0046] Appendix C below presents an example of tagging a sample document using the methods described above.

[0047] FIG. **3** is a flow chart that schematically illustrates a method for searching the set of documents in system **20**, in accordance with an embodiment of the present invention. The search is performed by searcher **40** after the documents have been tagged and indexed according to the method of FIG. **2**.

[0048] Searcher **40** receives a search query, typically from one of client computers **30**, at a search input stage **70**. Typically, the user of the client computer inputs the search terms and limitations via a suitable graphical user interface (GUI), and a program running on the computer converts the query to a structured form that is accepted by searcher **40**. Alternatively, the user may compose the query directly in this structured form.

[0049] As part of the query, the user specifies one or more business objects, at an object specification step **72**. These objects may be chosen by the user from a list of the objects held in repository **36**, or they may alternatively be entered manually by the user. In the latter case, the user may, for example, enter a partial name or nickname, and searcher **40** then automatically identifies the corresponding business object in repository **36** using techniques similar to those described above as part of the tagging process. Additionally

or alternatively, the user may specify that the search should be conducted over all business objects in a certain group or of a certain type, such as all customers in a given geographical area or all employees in a given department.

[0050] The user may also specify one or more keywords, in the form of a word or a phrase, at a keyword input step **74**, as in text-based search engines known in the art. The business objects that were specified at step **72** and the keywords, if any, specified at step **74** may be joined by logical operators, which are specified by the user at a logic specification step **76**. Such operators may include, for example, AND, OR, NOT, and may group the search terms into sub-queries. The user may also specify scoring refinements, indicating how much weight searcher **40** should give each part of the query in computing document scores.

[0051] Searcher **40** scores the documents in the repository or repositories of system **20** according to the search query, at a document scoring step **78**. Typically, this stage in the process uses the indices of business objects and, if appropriate, keywords that have been stored in repository **36**. As noted earlier, for each business object in the query, each instance occurring in a given document contributes to the score of that document, wherein the contribution depends, inter alia, on the level of confidence with which the business object was identified in the document. Furthermore, if the document contains a business object that is related to one of the business objects in the query (as identified by relation extraction module **60**), the related business object may also contribute to the document score.

[0052] The final score of each document is typically a weighted sum of the object scores, which are generated by matching the business objects in the query to the document, and of the keyword scores, due to matching of keywords in the document. In general, the object scores receive greater weight, although the weights may be adjusted based on user preference and application requirements. Searcher **40** ranks the documents according to the scores, at a ranking step **80**, and returns the ranked results to the user. Typically, the searcher returns a certain number of the documents that had the highest scores, or all documents with scores above some threshold.

[0053] The search results may be filtered by searcher **40** according to applicable permission (access control) list constraints, which are saved in repository **36** for both documents and business objects. The searcher checks and applies these constraints in a manner that is transparent to the user: If the user is not authorized to view or access a certain document, the document will not be included within the user search results. If the user is not authorized to view a certain business object, that business object will not be included within the business object tree displayed to the user (although nothing will change in the business object repository itself) and the business object tagging results referring to the business object within the searched document(s), if any, will not be displayed to the user.

Business Object Analysis and Validation

[0054] Business object analyzer function **52** (referred to hereinbelow simply as the business object analyzer) analyzes the business object name, its properties and its linked and related business objects if available in order to identify a complete list of variants (also referred to as variations). The variants include all strings that may (theoretically) be used within the text of a document as a reference to the business object. Each variant is a couple of a search string and a set of context-based, natural language constraints that must be met in order for a given instance of the variant (a business object candidate reference) to be considered a valid reference to that business object. The constraints are optional and differ from one type of business object to another. Examples of such constraints are listed below in Appendix D. The business object analyzer analyzes the business object name and its properties before any document is processed and prepares the variants to be used later, when documents are actually processed.

[0055] It is possible to store each variant explicitly, as a list of pairs, each pair consisting of the search string and the required constraints. Since there are typically many business objects in repository **36**, however, each with several variants, it is generally more efficient to store the variant pattern strings and, separately, the required constraints.

[0056] Each variant that may occur in a document is typically also assigned a score—a certainty level (expressed as a percentage, with 100% as maximum and 0% as minimum), indicating the likelihood that an instance of this variant is indeed a valid reference to the specific business object in question. Typically, the shorter the variant string, the more frequent it is within documents and texts in general, and the fewer the available contextual cues around the instance in a document, the lower will be the variant score. For example, the score of a variant of a person's full name (e.g., David Carlisle Fisher) will be higher than the score of the variant consisting of the very frequent first name David when occurring alone. Some examples of variants and their respective scores are listed below in Appendix A.

[0057] Once a business object candidate reference is found within the text of a document, it is linked to the business object if the candidate string is identical to a recognized variant string and if the specific instance within the text obeys the variant constraints.

[0058] The creation of the variants is based on analyzing the linguistic and semantic structure of the business object name and attributes. As noted earlier, in natural language texts the variant is often a shorter version of the name, instead of the full name. The ways in which such variants may be created and used by document authors are based mainly on the base type of the business object. For example, a person (employee) name is usually compounded of a first name, last name and (possibly) middle name. In most references to a person, either the first name or the last name (depending on the context) is omitted.

[0059] In organizations names, however, there is no formal division into parts. There is one name that often includes some semantically meaningful suffixes, and the business object name should be analyzed accordingly. The business object analyzer therefore uses rules to analyze each name lexically and linguistically, thus identifying which words or tokens are less important suffixes, which may be omitted, and which are "core" words that cannot be omitted. For example (as in Appendix C below), in the organization name Avnet Components Israel Ltd., all tokens except Avnet may be omitted. On the other hand, in Israel Corporation, no token may be omitted.

[0060] Further examples of types of variant strings and their respective scores in listed below in Appendix A, while typical constraints are given in Appendix D.

Analyzing Links and Relations between Business Objects

[0061] Since business objects are complex objects, each business object may contain other business objects, or be a

member in another business object, or have another relation with a second business object. In an embodiment of the present invention, these relations are also processed by the business object analyzer, for subsequent consideration in classifying and tagging documents **56**. In other words, when the business object analyzer deals with a given business object, it also identifies and marks the related business objects, as indicated by the organizational repository.

[0062] For example, if John Adams, director of Technical Services of Alcatel, is mentioned in an e-mail, then this e-mail may be tagged as having a reference to "Alcatel" (albeit with a relatively low score), even if Alcatel is not mentioned at all within the mail. Similarly, if a different e-mail was sent to the group "Alcatel Top Executives" of which "John Adams" is a member, then that mail should be tagged or classified with a reference to "John Adams".

[0063] When documents are tagged and classified, each business object is searched directly according to its direct variants. After tagging is completed, however, the tagging results may also be used to derive the relevance score of the document to other, related business objects, as noted above. For example, if a document refers to David Fisher, and David Fisher is identified as working in the Finance Department, then the document is related (with lower score) to the Finance Department business object.

[0064] As noted earlier, the relations between business objects may be similarity relations, container relations (such as distribution list membership), or other relations. The score derivation formula may be configured according to the nature of the relation and the of the business objects themselves. For example, in the case of the container relation, the derived score may be inversely proportional to the distribution list size. The score derivation algorithm is described in greater detail in Appendix F.

Business Object Name and Properties Validation

[0065] As noted earlier, business objects are loaded into server **22** from various organizational repositories **28**, which may include old, irrelevant or even incorrect records. For example, entries such as "Build Master" found within an organization's Active Directory cannot be a valid employee name and should be excluded at an early stage. To avoid incorrect tagging and classification due to such records, the business object analyzer typically validates the business objects and their properties.

[0066] In an embodiment of the present invention, the business object analyzer validates business object names and properties using appropriate rules, which are written separately for each business object type according to its corresponding properties and semantic characteristics. For example, names of human beings should include only letters, possibly with some punctuation or connector characters, such as a hyphen or apostrophe.

[0067] The business object analyzer may use several level of validation: The lowest level of validation is "wrong", meaning that the business object is inacceptable and will not be used (for example a numerical string as a person's name). A middle validation level or status may be "warning", indicating that an important property is missing or seems to be problematic. (For example, "Build Master" as an employee name will generate a warning, since the name consists entirely of English dictionary words without a valid lexical first name.) The highest level is "correct", for example, the employee name "David Fisher".

[0068] Additional examples of validation rules are given in Appendix E below.

Business Object Life Cycle Management

[0069] FIG. **4** is a flow chart that schematically illustrates a method for updating business objects, in accordance with an embodiment of the present invention. The method begins with initial identification of a business object, at an object identification step **90**. The business object is typically identified by analyzing a structured repository such as a database, a CRM or similar system or a spreadsheet that is maintained by the organization in question. Alternatively, the business object may be identified based on tagging unstructured documents. The processes by which such business objects are identified and recorded were described above in detail with reference to FIG. **2**.

[0070] For each new business object, classifier **34** records a link to a source of information concerning the business object, at a link recording step **92**. The link typically indicates the source record from which the business object was derived. For example, the source record may be an entry in a database or spreadsheet, or an Active Directory listing, or a page of a document on which the business object was found.

[0071] Crawler **38** periodically detects changes in the source records of business objects, at a change detection step **94**. These changes may be detected, for example, by polling the source records of the business objects that are indicated in the business object listings. Alternatively or additionally, the crawler may receive event notifications from certain data sources, such as HR and CRM databases, when changes are made. In either case, the changes in the business object records may indicate, for example, a new address or telephone number of a person or company, or a newly-discovered nickname or abbreviation. Changes may also indicate deletion of existing business objects or addition of new ones.

[0072] Upon receiving an indication that a business object has changed, classifier **34** updates the information regarding the business object in the listing of business objects in repository **36**, at a business object update step. The updated information is used in subsequent tagging and indexing of new documents, as described above with reference to FIG. **2**. Furthermore, the classifier may use the updated business object information to update the tagging and indexing of documents that have already been indexed. For example, if the update indicates a new nickname for a given person, the classifier may tag and index occurrences of the new nickname as instances of the corresponding business object, which were not previously recognized. (This tagging and indexing could be performed by checking the general keyword index for occurrences of the string corresponding to the nickname, rather than going back over all the original documents.) Documents containing the nickname may subsequently be returned in searches that include this business object among the query terms.

[0073] It will be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

APPENDIX A

RULES FOR CREATING VARIANTS
BASIC (SUB-TYPE) BUSINESS OBJECT VARIATIONS

Variation String	Instant Score	Example
Name	80%	Ben-Gurion Airport
Name without all '-' characters	70%	Ben Gurion Airport
BO Known Synonym (For each synonym, a separate variation)	70%	BGN

EMPLOYEE (SUB-TYPE) BUSINESS OBJECTS: NAME-BASED VARIATIONS

Variation String	Instant Score	Example
FirstName + MiddleName + LastName	100%	Ronald James Bleakney
FirstName + MiddleName Initial + LastName	90%	Ronald J Bleakney
FirstName + MiddleName Initial + "." + LastName	90%	Ronald J. Bleakney
LastName + FirstName + MiddleName Initial	70%	Bleakney Ronald J
FirstName + "." + MiddleName + LastName	70%	R. James Bleakney
FirstName Initial + MiddleName Initial + LastName	60%	R. J. Bleakney
FirstName + LastName	80%	Ronald Bleakney
LastName + FirstName(2)	60%	Bleakney Ronald
Honorific(1) + LastName	40%-60% (3)	Dr Bleakney
Honorific(1) + "." + LastName	40%-60% (3)	Mr. Bleakney
Honorific(1) + "." + FirstName Initial + "." + LastName	70%	Mr. R. Bleakney
Honorific(1) + "." + FirstName Initial + LastName	70%	Mr. R Bleakney
Honorific(1) + FirstName Initial + "." + LastName	70%	Mr R. Bleakney
Last Name (only)	30%-50% (4)	Bleakney
First Name Nick Name + Last Name (for each known nick name)	60%	Ron Bleakney, Ronnie Bleakney
First Name (only)	20-40%(5)	Ronald
Nick Name (only) - for each nick name	20%	Ron, Ronnie . . .

EMPLOYEE (SUB-TYPE) BUSINESS OBJECTS: NON-NAME-BASED VARIATIONS

Variation String	Instant Score	Example
Employee Mail	100%	Ron.Bleakney@radvision.com
Employee Phone Number (personal, full number)	70%	618-4534232
Employee ID Number (or Social Security number)(6)	90%	009516808

[0074] In addition, all the variations implemented for BASIC business objects are also implemented for EMPLOYEE business objects.

[0075] Notes to the above tables:

[0076] (1) Honorific: Mr, Mrs, Miss, Ms, Dr or Prof.

[0077] If Employee Gender is FEMALE, then Honorific may be only: Mrs, Miss, Ms, Dr or Prof.

[0078] If Employee Gender is MALE, then Honorific may be only Mr, Dr or Prof.

[0079] (2) This variation is not allowed if the last name is a "known" (lexicon-based) first name—in order to not confuse for example "Chaim Moshe" and "Moshe Chaim".

[0080] (3) Last names are divided into 3 groups: Very common last names (about 1% of the general population or more—"Smith", "Cohen" in the Israeli population), common last names (about 0.2% of the general population ("Biton" in the Israeli population), and other last names.

[0081] Score for the variation when the last name is a "very common last name": 40% ("Mr Levi").

[0082] Score for the variation when the last name is a "common last name": 50% ("Dr Anwar").

[0083] Score for the variation when the last name is other: 60%.

[0084] (4) Score is the respective score calculated in (3) less 10%.

[0085] (5) Similar division (to that described in (3)) exists for first names:

[0086] Score for the variation when the first name is a "very common first name": 20% ("David").

[0087] Score for the variation when the first name is a "common first name": 30% ("Lucia").

[0088] Score for the variation when the first name is other: 40%.

[0089] (6) The Employee ID must be at least six characters long or at least three characters long with the first character a letter.

ORGANIZATION (SUB-TYPE) BUSINESS OBJECTS: NAME-BASED VARIATIONS

Variation String	Instant Score	Example
Organization (full) name	80%	Inxight Software, Inc.
Name with sure company suffix removed(1)	70%	Inxight Software
Name with unsafe company suffix removed (2)	60%	Inxight
Name after removing a suffix within parenthesis(3)	60%	Nestle (Canday Division) => Nestle
Name after removing country as suffix (Possibly with "de") (4)	60%	Atcolx de Mexico => Atcolx
First token of the name (5)	50%	Agilis Computersysteme => Agilis

ORGANIZATION (SUB-TYPE) BUSINESS OBJECTS: NON-NAME-BASED VARIATIONS

Variation String	Instant Score	Example
Organization mail domain	90%	@basistech.com
Organization main phone number (full)	70%	617-386-2000
Organization main fax number (full)	70%	617-386-2020

[0090] In addition, all the variations implemented for BASIC business objects are also implemented for ORGANIZATION business objects.

[0091] Notes to the above tables:

[0092] (1) This variation is not allowed if the resulting (remaining)string is a name of a country (example: "Israel Corporation"=>"Israel") or another very common term ("grupo", "asia").

[0093] (2) This variation is not allowed if the resulting (remaining) string is a name of a country, a big city, state or nationality or a dictionary English word or other common term as in (1) (example: "British Telecom"=>"British" is not allowed).

[0094] (3) This variation is not allowed if the resulting (remaining) string is a name of a country or an English dictionary word.

[0095] (4) This variation is not allowed if the resulting (remaining) string is an English dictionary word.

[0096] (5) This variation is allowed only if the first token is not one of the following: a number, a known (lexicon-based) first name, a country, state or a big city name, a nationality or another very common (junk) term.

PRODUCT (SUB-TYPE) BUSINESS OBJECTS: NON-NAME-BASED VARIATIONS

Variation String	Instant Score	Example
Product ID Number	90%	003456543-9

[0097] In addition, all the variations implemented for BASIC business objects are also implemented for PRODUCT business objects.

Location (Sub-Type) Business Objects: Name-Based Variations

[0098] The variations implemented for LOCATION business objects are solely the variations implemented for BASIC business objects. (LOCATION business objects, however, require context-natural language constraints not required for BASIC business objects, as listed below in Appendix D below. All LOCATION variations are considered name-based variations.)

Appendix B—Properties of Business Object Types

[0099] Business objects are dealt with and analyzed by type. Each type typically has its own properties and rules for analyzing the properties. Typical business object types include EMPLOYEE, ORGANIZATION, PRODUCT, LOCATION or BASIC (unknown/other base type).

[0100] The table below lists some of the properties of these business object types by way of example:

BO Attribute Name	Relevant BO Base Types	Description	Example
Bo Known Synonym	All	Known synonym of the BO	CTM 3.3 (for Click to Meet 3.3)
Employee First Name	Employees	Employee first name	Michael
Employee Middle Name	Employees	Employee middle name	Benjamin
Employee Last Name	Employees	Employee last name	Jones
Employee Mail	Employees	Employee mail	Michael.jones@radvision.com
Employee Phone	Employees	Employee phone	712-4534902
Employee ID	Employees	Employee national id number of social security number	009516888
Employee Gender	Employees	MALE or FEMALE	MALE
BO Domain Name	Organizations	Organization web or email domain	videocentric.co.uk
BO Phone Number	Organizations	Organization's main phone number	44(0)118 9740125
BO Fax Number	Organizations	Organization's main fax number	44(0)118 9740126
Product ID	Products	Product unique ID number	003456543-9

Appendix C—Example of Business Object Tagging

[0101] In this example, system **20** is used by a hypothetical company ("First Sample Corporation") selling to customers in various countries. The company documents may be in different languages accordingly (English, Italian, Dutch, etc.) In addition, even within English documents, the customers' names may be in different languages, as listed below in Table I, which is an extract from the "Customers" table in the company's CRM system. The customers may be referred to in company documents by their full names, partial names or synonyms. Other properties may be used, as well, as customer references, such as an e-mail address, Web domain, customer ID number, etc.

TABLE I

CUSTOMER TABLE

Customer ID	Customer Main Name	Customer Additional Name	Customer Internet Domain
1173	Stichting Pandora		stichtingpandora.nl
1174	Istituto Nazionale per la Fisica della Materia	INFM	infm.it
1175	Avnet Components Israel Ltd.		avnet-israel.com
1176	Israel Corporation		israelcorp.com
1177	Cruz Roja Chilena		cruzroja.cl
...			

[0102] Similarly, First Sample Corporation may have hundreds of employees from various cultural backgrounds. Each employee may have a full name (consisting of a first name, a middle name and a last name) and possibly also a nickname. Reference to an employee may be by his or her full name, partial name (first name only, last name only) or his/her nickname. Common first names such as David are probably shared by several employees. Table II is an extract from the company's HR module, listing the company's employees:

TABLE II

EMPLOYEE TABLE

Employee ID	First Name	Middle Name	Last Name	E-mail	Direct Phone
17	David	Carlisle	Fisher	david.fisher@samplecorp.com	6394444
18	Robert		Jones	bob2.jones@samplecorp.com	6394445
19	David	Jefferson	Jones	dave.jones@samplecorp.com	6394443
20	Francisco		Gomez	Paco@samplecorp.com	00-1-212-6543222
...					

[0103] Classifier **34** tags the company's documents, including the following e-mail sent by David Carlisle Fisher, the company's CFO, to his assistant, Robert Jones:

[0104] Hi Bob,

[0105] Please check what's going on with the Pandora and INFM orders. Please also write Jose Rodriguez (his mail is finanzas@cruzroja.cl) and remind him we're still awaiting payment. If this does not help, I'll ask Paco to talk with him when he gets back from Israel—you know that his English is not the best. We'll talk tomorrow about Avnet and about this new customer (Grupo Anaya S.A.).

[0106] Thanks,

[0107] Dave

[0108] David C. Fisher

[0109] Chief Financial Officer

[0110] First Sample Corporation

[0111] david.fisher@samplecorp.com

[0112] The boldface terms in the letter above represent entities (customers and employees), which are tagged using the business object information in repository **36**.

[0113] Customer Names:

[0114] The e-mail includes references to five customers, four of which are listed in the above customer table: Stichting Pandora, Istituto Nazionale per la Fisica della Materia, Cruz Roja Chilena and Avnet Components Israel Ltd. One customer is not listed in the table: Grupo Anaya S.A.

[0115] For the four customers that are listed, the references used in the e-mail are different from the full customer names in the CRM table. Classifier **34** nevertheless is able to identify all four objects, since it has automatically created the valid possible variants based on the stored properties of the objects. The classifier parses each name using linguistic and semantic heuristics in order to find the valid variants, and then searches the document for valid matches of these variants. In the present example, the variants are identified as follows:

[0116] Stichting Pandora—"stichting" means "foundation" in Dutch. The classifier thus identifies this name as a Dutch name. In Dutch, German and other languages, such a head word is usually the first word in the name, and the classifier therefore identifies "Pandora" automatically as a valid variant or synonym. (Such a rule would not generally be correct in English). In addition, the classifier distinguishes between valid contextual instances of this variant ("Pandora" as above) and invalid instances ("It is a Pandora's box" or "Pandora Smith").

[0117] Istituto Nazionale per la Fisica della Materia—The classifier identifies the reference "INFM" although it is not the full name, but rather an acronym. In this case the acronym is available within the customer listings. Even had it not been listed, however, the classifier would have created it automatically by eliminating stop-words (prepositions and articles) within the Italian name ("per", "la", "delta"), and taking the first letters of the rest of the words. The fact that the acronym is listed in Table I, however, will raise the score (confidence level) that the classifier attaches to this reference.

[0118] Cruz Roja Chilena ("Chilean Red Cross" in Spanish)—This reference is identified based not on the name, but on another property: the related e-mail or internet domain, cruzroja.cl.

[0119] Avnet Components Israel Ltd.—The classifier identifies the first token of the name, "Avnet", as a valid reference, since the name is identified as a standard English name, and the token is not identified as ambiguous (i.e., a word ordinarily used in another context). By contrast, in names such as Stichting Pandora or Istituto Nazionale per la Fisica della Materia, the classifier identifies the first word (stichting or istituto, meaning "institute" in Italian) as a meaningful keyword, which is less likely to be used alone as a reference to the business object. Similarly, for the name Israel Corporation, the token Israel alone is most likely to refer to the country and not to the company known as Israel Corporation.

[0120] Grupo Anaya S.A.—This customer is not (yet) listed but its name is extracted by entity extraction module **58**.

[0121] Employee Names and Other Person Names:

[0122] The referenced employees in the letter above are David Carlisle Fisher, Robert Jones and Francisco Gomez. In addition the letter contains a reference to another person, Jose Rodriguez, who is an employee of a customer (Cruz Roja Chilena).

[0123] All employees are referred to in the letter by their nicknames (Dave=>David, Bob=>Robert, Paco=>Francisco). The classifier automatically assigns for each person's name the appropriate list of possible nicknames, based on that person's full name and the corresponding nicknames that are common in various cultures.

[0124] The name "Dave" may theoretically refer to 2 employees: David Carlisle Fisher and David Jefferson Smith. Since the email includes a safer (more complete) version of David Fisher's name (David C. Fisher), however, and his e-mail (david.fisher@samplecorp.com), the classifier concludes that the nickname Dave should be resolved solely as an instance of David Carlisle Fisher.

[0125] The name Jose Rodriguez is extracted by entity extraction module **58**. Based on the context within the mail: "(his mail is finanzas@cruzroja.cl)," the classifier identifies him as a contact person or employee of the customer Cruz Roja Chilena. Once this link is extracted from the mail, it can be stored within the CRM database for future use.

Appendix D—Context-Based Natural Language Constraints

[0126] In addition to matching a valid business object variant string, a business object reference may also be required to obey certain constraints regarding the context within the document in which the string occur. Examples of such context-based constraints include:

For Employee, Organization and Location (Sub-Types) Business Objects:

[0127] For Named-based variation:

[0128] 1. The morphological (and hence, syntactic and semantic) role of the string within the document must be an Entity/Object name (proper noun in linguistic terms). In English such strings are usually, but not always, noted by the use of capitalization. Other languages may have different rules.

[0129] Examples:

[0130] (i) Business Object: Analog Devices, Inc.

[0131] Valid Reference: "Other critical components offered by Analog Devices are . . . "

[0132] Invalid reference: "Analog devices are used to measure electrical quantities."

[0133] (ii) Business Object: William Jackson Smith

[0134] Valid Reference: "Please call Bill."

[0135] Invalid Reference: "This bill is very problematic."

[0136] 2. The variant string must not be a part of another name or entity.

[0137] Examples:

[0138] (i) Business Object: London (capital of U.K.)

[0139] Valid Reference: "I'm flying to London next week."

[0140] Invalid reference: "Yaron London will participate in this show."

[0141] (ii) Business Object: Williams (a major oil company)

[0142] Valid Reference: "In 1966, Williams paid $287 million for the country's largest petroleum products pipeline."

[0143] Invalid reference: "He has a B.A. degree from Williams College."

For Other (Not Name) Properties (ID Number, Phone Number Etc.)

[0144] Constraint 1 above is not required, because the morphological role of such a property is different (only numerical). Other constraints, however, are usually required for numerical (or ID) properties, such as contextual cue-based constraints: Typically, the reference business object candidate must be prefixed by a (lexicon-based) term indicating that this string indeed refers to the business object property.

[0145] Example:

[0146] For the property phone number, the string (reference candidate) must be prefixed by one of the strings: "phone:", "tel.", "call us at", "1-800-", "1-808-", etc.

[0147] The goal of this constraint is to distinguish between:

[0148] "Just call us at 1-808-624-8222 and mention that you want our internet airport special of $30,"

[0149] and

[0150] "U.S. Pat. No. 6,248,222."

Appendix E—Business Object Validation Rules

[0151] Business object analyzer function **52** validates each new business object before inserting it into repository **36** in order to avoid incorrect entries that might harm the classification and tagging process (including both naive errors/junk and malicious content). For example, an entry such as "Build Master" found within the organization Active Directory cannot be a valid employee name. If the language of the business object name is specified or can be identified automatically, the analyzer uses this language in validating the business objects. Otherwise, English is used as the default language.

[0152] Based on analyzing its properties, a new business object may be classified as either:

[0153] Valid—May be used for tagging and classification without any further required action.

[0154] Invalid—Cannot be inserted or used with its current properties. At least one property must be changed or added in order to make it valid.

[0155] Warning—A possible problem with a property was found, or an important property is missing. If the user

wishes, however, the business object may be still inserted. There are two warning types:

[0156] Major Warning—The business object will not be inserted (as with invalid business objects), and a suitable message will be displayed. If authorized by the user, the business object will be inserted.

[0157] Minor Warning—The business object will be inserted (despite the warning), and a warning message will be displayed.

Validation Constraints—All Business Object Types:

[0158] A business object that does not meet the following constraints is considered invalid:

[0159] The business object name must be not empty. (If the business object Type is Employee, the Last Name must not be empty.

[0160] The business object name must be longer than one character. (For Employee business objects: the concatenation of all three name parts: first name, middle name and last name, must be longer than one character, and the last name must be longer than one character).

[0161] The business object name must be no longer than **120** characters.

Validation Constraints—Employee Business Object Type:

[0162] Required constraints (a business object not meeting these constraints is considered invalid):

[0163] The Last Name property must not be empty.

[0164] The First Name, Middle Name and Last Name properties may include only letters or the following characters: -, ',: ‘, ’, ’or ‘.

[0165] All of the above properties may be no longer than 30 characters.

[0166] Optional constraints (a business object not meeting these constraints will trigger a warning):

[0167] Employee First Name is missing—minor warning.

[0168] Employee Mail(Email) is missing—minor warning.

[0169] Employee name seems to be a general term and not a person name. (All the components of the name, including first, middle and last names, are dictionary words or organization suffixes or indicators, and the first name is not a lexicon-based first name—major warning.

Validation Constraints—Organization Business Object Type:

[0170] Required constraints (a business object not meeting these constraints is considered invalid):

[0171] The organization name cannot be a general/frequent term alone: “Systems”, “Suppliers”, etc.

[0172] Optional constraints (a business object not meeting these constraints will trigger a warning):

[0173] Organization mail/internet domain is missing—minor warning.

[0174] Words in the organization name indicate that it has probably been marked as inactive/irrelevant by the system user: “cancel”, “inactive”, etc.—minor warning.

[0175] The organization name is identical to a country name (“Israel”, “France”)—major warning.

[0176] The organization name appears to be a person’s name (starting with a lexicon-based known first name)—major warning.

Validations Constraints—Location Business Object Type:

[0177] The Location Name (in any language) may include only letters or the following characters: -, ., (,).',: ‘, ’, ’or ‘. A business object whose name (in any language) includes such characters will be considered invalid.

Appendix F—Score Derivation

[0178] FIG. **5** is a flow chart that schematically illustrates a method for deriving scores of business objects in a document, in accordance with an embodiment of the present invention. The method uses tagging results **100** that were previously assembled by tagging business object names (including variants) in the document. Server **22** gets all tagged business objects from the document, at an object fetching step **102**, and adds each business object to the appropriate listing in repository **36**, at an object recording step **104**. The server checks whether there are any more objects to process, at an object checking step **106**. When all the objects have been completed, the classification results are saved, at a result storage step **108**, and the method terminates.

[0179] As long as there are still objects to process at step **106**, the server looks up all relations for the current business object, at a relation fetching step **110**. The server checks whether the current business object has further relations to handle, at a relation checking step **112**. If there are no further relations, the process returns to step **106** to take the next business object or terminate.

[0180] As long as there are relations to process at step **112**, the server checks whether the current relation is of a type that can the document to be assigned a score with respect to the related business object, at a relation checking step **114**. If not, the method returns to step **112**. If there is a score to be assigned, the server loads the relation scoring routine, at a scorer loading step **116**. This routine assigns a score for the related business object, at a score assignment step **118**. This score is typically the maximum between any existing score that has already been assigned to the document for the related business object and a new score that may be derived from the current business object. The related business object is inserted or updated in repository **36**, at an object updating step **120**, and the new score for this business objected is added to the repository, as well, at a score recording step **122**.

[0181] This process continues until all of the relevant business objects and their relations have been handled.

1. A computer-implemented method for processing information, the method comprising:

collecting data objects from one or more data repositories, the data objects having respective properties, which identify the data objects;

analyzing the properties of the collected data objects in order to derive respective identifiers corresponding to the data objects;

identifying a text string that matches one of the identifiers of a data object within a context in a document;

generating, responsively to the context, an indication that the identified text string is a valid instance of the data object; and

processing the document responsively to the indication.

2. The method according to claim 1, wherein the properties comprise names of the data objects, and wherein analyzing the properties comprises identifying variants that are different from the names.

3. The method according to claim 2, wherein the variants are selected from a group of variant types consisting of a part of a name, an abbreviation of the name, and a nickname.

4. The method according to claim 1, wherein analyzing the properties comprises applying natural language analysis to the properties in order to find the identifiers, and wherein generating the indication comprises recognizing the valid instance within the context in the document by applying natural language analysis to the document.

5. The method according to claim 1, wherein the data repository contains information in multiple different languages, and wherein analyzing the properties comprises deriving the identifiers that are respectively applicable in each of two or more of the languages, and wherein identifying the text string choosing the identifiers responsively to a language of the document.

6. The method according to claim 1, wherein generating the indication comprises computing a score indicative of a level of confidence that the identified text string validly represents the data object.

7. The method according to claim 6, wherein each of the identifiers is derived from the properties of the data object using a respective rule, and wherein computing the score comprises assigning the level of confidence to each of the identifiers responsively to the respective rule.

8. The method according to claim 6, wherein processing the document comprises assigning a respective document score to the document, indicative of a relevance of the document to the data object, responsively to the score.

9. The method according to claim 1, wherein analyzing the properties comprises identifying a relation between at least first and second data objects, and wherein processing the document comprises assigning a document score to the document indicating that the document is relevant to the first data object responsively to a match between the text string and one of the identifiers of the second data object and to the relation.

10. The method according to claim 9, wherein the relation is selected from a group of relations consisting of container relations, similarity relations, hierarchical relations and affinity relations.

11. The method according to claim 1, wherein analyzing the set of data objects comprises identifying respective records in the data repository corresponding to the data objects, and wherein processing the document comprises generating a listing of occurrences of the data objects in a corpus of documents, detecting a change in one of the respective records corresponding to one of the data objects, and responsively to analyzing the change, automatically updating the listing with respect to the one of the data objects.

12. The method according to claim 1, wherein processing the document comprises generating a response to a search query based on valid instances of the data objects that occur in the document.

13. The method according to claim 1, wherein collecting the set of the data objects comprises extracting a first set of the data objects from a repository of structured data, and extracting one or more second data objects, not in the initial set, from the document, and adding the second data objects to the first set.

14. The method according to claim 13, wherein adding the second data objects comprises comparing the second data objects to the data objects in the first set, and adding the second data objects upon determining that the second data objects do not match any of the data objects in the first set.

15. The method according to claim 1, wherein analyzing the properties comprises applying predetermined rules in order to validate the collected data objects before using the data objects in processing the document.

16. The method according to claim 15, wherein applying the predetermined rules comprises making a determination selected from a group of determinations consisting of determining whether the data object should be used in processing the document, whether a property of the data object should be used in processing the document, and whether a property of the data object is missing, and wherein the determination is based on at least one of comparing the property with a lexicon, comparing the property with a vocabulary, and matching the property with one or more regular expressions.

17. The method according to claim 1, wherein collecting the data objects comprises retrieving and analyzing access control information with respect to each of at least some of the data objects and the properties of the data objects, and wherein processing the documents comprises providing an output to a user while filtering at least a portion of the output using the access control information.

18. A computer-implemented method for processing information, comprising:

collecting data objects from one or more data repositories and identifying a respective record in the repositories corresponding to each of the data objects;

processing one or more documents so as to generate a listing of occurrences of the data objects in the documents;

detecting a change in the respective record corresponding to one of the data objects;

responsively to the change in the respective record, automatically updating the listing with respect to the one of the data objects; and

processing the documents responsively to the listing.

19. The method according to claim 18, wherein detecting the change comprises polling records in at least one of the data repositories that correspond to the set of data objects.

20. The method according to claim 18, wherein detecting the change comprises receiving an event message that is indicative of the change from at least one of the data repositories.

21. Apparatus for processing information, comprising:

an interface, which is coupled to communicate with one or more data repositories; and

a processor, which is configured to collect data objects from the one or more data repositories, the data objects having respective properties, which identify the data objects, to analyze the properties of the collected data objects in order to derive respective identifiers corresponding to the data objects, to identify a text string that matches one of the identifiers of a data object within a context in a document, to generate, responsively to the context, an indication that the identified text string is a valid instance of the data object, and to process the document responsively to the indication.

22. Apparatus for processing information, comprising:

an interface, which is coupled to communicate with one or more data repositories; and

a processor, which is configured to collect data objects from the one or more data repositories while identifying a respective record in the repositories corresponding to each of the data objects, to process one or more documents so as to generate a listing of occurrences of the data objects in the documents, to detect a change in the respective record corresponding to one of the data objects, to automatically update the listing with respect to the one of the data objects responsively to the change in the respective record, and to process the documents responsively to the listing.

23. A computer software product, comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the computer to collect data objects from one or more data repositories, the data objects having respective properties, which identify the data objects, to analyze the properties of the collected data objects in order to derive respective identifiers corresponding to the data objects, to identify a text string that matches one of the identifiers of a data object within a context in a document, to generate, responsively to the context, an indication that the identified text string is a valid instance of the data object, and to process the document responsively to the indication.

24. A computer software product, comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the computer to collect data objects from one or more data repositories while identifying a respective record in the repositories corresponding to each of the data objects, to process one or more documents so as to generate a listing of occurrences of the data objects in the documents, to detect a change in the respective record corresponding to one of the data objects, to automatically update the listing with respect to the one of the data objects responsively to the change in the respective record, and to process the documents responsively to the listing.

* * * * *