

(51) International Patent Classification:
G06F 17/30 (2006.01)(21) International Application Number:
PCT/EP2010/051055(22) International Filing Date:
29 January 2010 (29.01.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09151971.0 3 February 2009 (03.02.2009) EP(71) Applicant (for all designated States except US): **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).(71) Applicant (for MG only): **COMPAGNIE IBM FRANCE** [FR/FR]; Tour Descartes, La Defense 5, 2 Avenue Gambetta, F-92400 Courbevoie (FR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **EL-ASMAR, Mohamed** [EG/EG]; IBM Egypt, Pyramids Heights Office Park C10, Alexandria Desert Road, PO BOX 166 El-Ahram Giza, Cairo, 9999 (EG). **RAGHEB, Ahmed** [EG/EG]; IBM Egypt, Pyramids Heights Office Park C10, Alexandria Desert Road, PO BOX 166 El-Ahram Giza, Cairo, 9999 (EG). **FATHY, Mohamed** [EG/EG]; IBMEgypt, Pyramids Heights Office Park C10, Alexandria Desert Road, PO BOX 166 El-Ahram Giza, Cairo, 9999 (EG). **EL-SHISHINY, Hisham** [EG/EG]; IBM Egypt, Pyramids Heights Office Park C10, Alexandria Desert Road, PO BOX 166 El-Ahram Giza, Cairo, 9999 (EG).(74) Agent: **BELL, Mark**; Le Plan du Bois, F-06610 La Gaude (FR).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR SEMANTIC SEARCHING

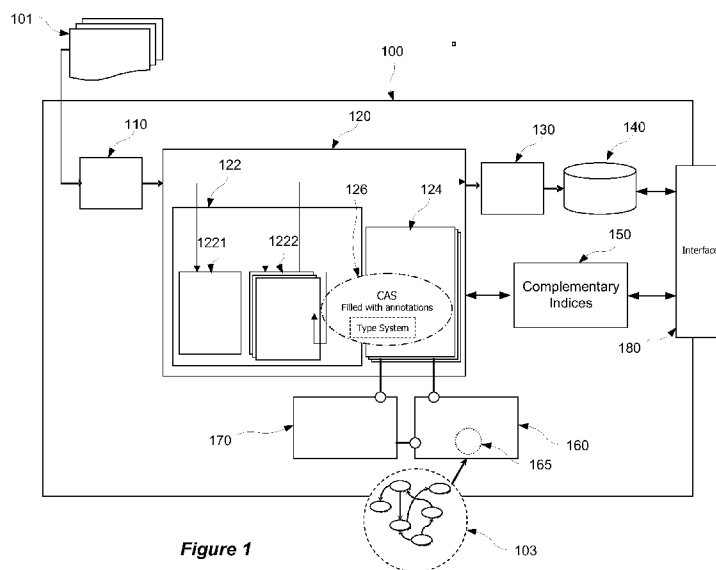


Figure 1

(57) Abstract: A method of processing source document to provide complementary semantic databases to enable an improved semantic search. Source documents are parsed and type annotations indicating adherence to one of a set of predefined types are associated with particular words in the source documents. For each annotated word semantic keys are identified that corresponding to the annotations associated therewith. These semantic keys correspond to semantic keys defined in a semantic model, which associated each semantic key with a plurality of semantic descriptors. On this basis a complementary database is compiled associating each word with the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective word. A number of different complementary databases are proposed.



Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

Method and System for Semantic Searching

Field of the invention

The present invention relates to the field of semantic searching, and more particularly to the
5 field of processing source documents for semantic search.

Background of the invention

Typical search engines like Google and Yahoo use a keyword-based approach for information search and retrieval, where queries consist of keywords, and search results are
10 represented as a ranked, flat list of documents, with textual sections containing spans of text with mentions of keywords. Incorporating semantic analysis based on different knowledge representation sources (such as ontologies, topic maps, semantic nets...) and text analytics can improve search results over simple keyword-based search. Some search engines such as "OmniFind" allow for the inclusion of semantic information in the indexing
15 phase through integrating Text Analytics Engines via UIMA (Unstructured Information Management Architecture, as described in for example:

http://domino.research.ibm.com/comm/research_projects.nsf/pages/uima.index.html). This semantic information allows the formation of semantic queries in the form of "XML Fragments", which is a good step towards semantic searching.

20 In the Proceedings of the Ninth International Conference on Information Visualization (IV'05), 1550-6037/05, 2005 - Nicolas Bonnel, Alexandre Cotarmanac'h, and Annie Morin (Meaning Metaphor for Visualizing Search Results.) an approach is described which visualizes search results using a 3D metaphor. For example, search results can be visualized as a city metaphor where related documents are clustered into blocks and each
25 document is represented as a building.

The article entitled "FacetMap: A Scalable Search and Browse Visualization." IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 12, NO. 5, TEMBER/OCTOBER 2006 - Greg Smith, Mary Czerwinski, Brian Meyers, Daniel Robbins, George Robertson, and Desney S. Tan, uses a multi-faceted representation of
30 documents. A facet is a possible categorization of documents, for example, "By date", "By type" or "By Author". Facets are visualized as squares. Each square contains possible values of such facet visualized as bubbles. For example, the "By Location" facet may contain two bubbles: "United states" and "Canada". Each bubble is a short string of the

value of the facet, and a count of how many documents this bubble contains. This visualization lacks discovery capabilities; because bubbles are visualized as a black-box showing no clues about what concepts and relations documents in this bubble contains (i.e. it doesn't show how documents in this bubble are related to other facets), so, the user must go inside the bubble and start reading its documents, to know if this information is what he needs.

The following commercial products attempt also to provide a solution to the problem under consideration:

10

- KartOO (<http://www.kartoo.com/>): a term-based visualization of search results that allows the user to visualize web pages-terms relationships. It depicts a map of links and high ranked terms, and when the user hovers over a web page, it highlights terms that occurred in this page; and when the user hovers over a term, it highlights web pages where this term occurred. When the user clicks on a term, this term is added to the query to refine it, thus narrowing down the search. This visualization helps a user understand term co-occurrences, and to get a grasp of what a web page is all about without the need to read it. KartOO doesn't use any semantic information, and relies just on terms mentioned in web pages. Also, it does not show relations between terms; except the "co-occurrence" relation.

20

Grokker (<http://www.grokker.com/>): views search results categorized based on their Semantic Web metadata. It visualizes categorization in two forms:

- a) "Outline view" which shows categories in a hierarchy tree.
- b) "Map view" which shows categories as a cluster map. When user clicks on a category, search results belonging to this category are shown on the right side as a classical flat ranked list of hits, with textual summarization. The user can also filter content. Organizing search results into a tree structure is helpful to the user, allowing him/her to navigate through the tree hierarchy till reaching relevant information, but, semantic relations are richer than just a tree.

30

- Aduna AutoFocus (<http://www.aduna-software.com/technologies/autofocus/overview.view>): creates an index for a source (a folder for example) to enable multi-faceted searching and discovery. A user starts the query with a specific value of a facet (for example, documents containing a specific keyword);

documents matching this facet value will be shown as a bubble; when this bubble is selected, the “Navigation” view on the left gets updated to show how documents in this bubble can be categorized using other facets (suggested keywords found in documents belonging to this bubble, or type of documents found in this bubble). This allows the user
5 to refine the query by adding other bubbles. This visualization allows both narrowing down of retrieved documents and expansion to include other documents using static metadata associated with documents but no focus on semantic concepts.

In US 2007/0038608 an Internet search system is described, mainly for product search, that
10 also searches additional data sources to get information directly related to the contents of the retrieved web pages, but does not appear on them, in order to improve the ranking of the found web pages.

WO 2006/116273 A2 describes a system for categorizing terms, phrases, documents and/or
15 clustering term co-occurrence with respect to a taxonomy. It provides an automated means for assigning objects, such as websites, to appropriate categories of a taxonomy.

WO 2006/128123 A2 describes a search engine that uses Natural Language Processing (NLP) techniques and ontological semantics in analyzing the meaning of queries and the
20 searched text, thereby producing equivalent meanings to a sequence of user initiated words.

WO 2004/075466 A2 describes an integrated implementation framework and resulting medium for knowledge retrieval, management, delivery and presentation. The framework is based on two servers that work together to provide context and time-sensitive semantic
25 information retrieval services.

WO 2007/059287 A1 describes a system for searching for information in a data set by syntactically indexing and performing syntactic searching of data sets using relationship queries in order to improve the search result accuracy.

30

Given the above, there is a need for systems that allow users to do advanced semantic search in a collection of documents that leads, through innovative visualization of search results, to further knowledge exploration and content discovery.

Summary of the invention

According to the present invention there is provided a method of processing source
5 artefacts according to the appended independent claim 1, and a computer program, a
computer readable medium and a system according to appended claims 12 to 14
respectively. Preferred embodiments are defined in the appended dependent claims 2 to 11
and 15.

10 Further advantages of the present invention will become clear to the skilled person upon
examination of the drawings and detailed description. It is intended that any additional
advantages be incorporated herein.

Brief description of the drawings

15

Embodiments of the present invention will now be described by way of example with
reference to the accompanying drawings in which like references denote similar elements,
and in which:

- 20 Figure 1 is a block diagram showing components of a system common to certain
embodiments described hereafter;
- Figure 2 shows details of the complementary indices 150 in accordance with a first
embodiment;
- Figure 3 shows a flowchart describing certain steps implementing the first embodiment;
- 25 Figure 4 shows details of the complementary indices 150 in accordance with a second
embodiment;
- Figure 5 shows details of the complementary indices 150 in accordance with a third
embodiment;
- Figure 6 shows a block diagram of a system in accordance with a fourth embodiment;
- 30 Figure 7 shows a flowchart for the modified parsing & annotation linking processes
according to the fourth embodiment;
- Figure 8 shows a flowchart for the behaviour of the modified Topic ID based
annotation linker according to the fourth embodiment;

- Figure 9 shows a subset of the information provided in a Topic Map representation of a semantic model 103;
- Figure 10 shows the mapped Semantic Model 165 from the previous Topic Map shown in figure 9; and
- 5 Figure 11 shows an example for the annotations created by the domain specific annotator 1222.

Detailed description

- 10 As described hereafter, there is provided a method and system for processing source artefacts, that is, electronic documents or other electronic data files. The method comprises the steps of associating type annotations with elements of one or more source artefacts, where said type annotations indicate that adherence to one of a set of predefined types, for each element thus annotated identifying semantic keys corresponding to the annotations
- 15 associated therewith in a semantic model, said semantic model comprising a plurality of said semantic keys each being associated with a plurality of semantic descriptors; and associating the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, with that element.
- 20 Figure 1 is a block diagram showing components of a system common to certain embodiments described hereafter. As shown in figure 1 there is provided a search engine 100. The search engine 100 comprises a crawler 110, which retrieves source artefacts 101 such as documents, from a specific source (e.g. Web content, Database, File system ... etc.). The search engine 100 further comprises a Parser 120, which may perform various Natural
- 25 Language Processing (NLP) tasks on the artefacts retrieved by the crawler 110. Common NLP tasks carried out by the parser may include language identification, tokenization, stemming, part-of-speech tagging, and normalization. The search engine 100 further comprises an Indexer 130, which stores the data created by the parser 120 in the main index 140 that facilitates fast and accurate information retrieval. There is also provided a Search
- 30 Runtime (not shown), which provides an interface to the Main index 140 that is used by the Search application to issue search queries and retrieve the relevant documents for that query. The Search runtime should provide façade API 180 for search applications to submit queries and retrieve the results (Search Engine APIs) from the main index 140 and the complementary indices 150.

Traditional searches are augmented and improved by leveraging semantic data to disambiguate semantic search queries and web text in order to increase relevancy of results. According to the present embodiment, this is done by enabling different semantically aware processing components to be plugged into the parser 120 to perform semantic analysis on the crawled artefacts 101. The analysis results are then written to one or more indices using a mapped semantic model 165 derived from a semantic model 103. This information enables retrieval of artefacts based on semantic queries.

10 According to the present embodiment, the parser component uses UIMA (Unstructured Information Management Architecture) to provide the infrastructure needed to manage and compose multiple analysis components.

UIMA (<http://sourceforge.net/projects/uima-framework>) refers to an analysis component as an annotator; a software component that implements the UIMA annotator interface to produce and record annotations over regions of an artefact (a text document in this case). An annotation is the association of a metadata, such as a label, with a region of text. For example, the label "City" associated with a region of text "Cairo" constitutes an annotation. In UIMA, an annotation is represented as a special type in a UIMA type system.

20 Annotations are recorded by an annotator in a data structure named the CAS. The CAS (Common Analysis Structure) is the primary data structure which UIMA analysis components use to represent and share analysis results. A UIMA Type system is a collection of "types". A type is a specification of an object in the CAS used to store the results of analysis. Types usually contain Features, which are attributes, or properties of the type. According to certain embodiments of the present invention, a UIMA type system is used to represent the semantic model, where a UIMA type represents a topic type which is the class of topics that a particular topic belongs to (e.g. Country, City, Organization ... etc), with UIMA type features as attributes of the topic type. As shown, there is provided a lexical analysis annotator 1221 and a plurality of domain specific annotators 1222.

30

When several annotators 1221, 1222 are grouped together, UIMA provides a component to wrap the aggregated annotators in a single unit. This unit is named the Aggregate analysis engine component 122. In the invented system, this component is composed from a Lexical analysis annotator 1221, which performs the Natural Language Processing (NLP) tasks for

the traditional search, and Domain Specific Annotators 1222 (either statistical, rule based or dictionary based) that analyze the documents based on semantic information for a specific domain. The resulting annotations are stored in the CAS 126.

- 5 UIMA defines another component – the CAS consumer - that receives the CAS after processing is done by an Analysis Engine. It is responsible for taking the results from the CAS and using them for some purpose. The CAS Consumer may also perform collection-level analysis, saving these results in an application-specific, aggregate data structure.
- 10 In the system of the present embodiment, there is provided a Complementary Indices Creator component 124, which is responsible for building the complementary indices. According to the present embodiment the Complementary Indices Creator component 124 is implemented as a CAS consumer. Thus it runs in the UIMA analysis pipeline. The Complementary Indices Creator 124 scans the CAS 126 for the annotations created by the
- 15 domain specific annotators 1222. It uses an Annotation Linker component 170 to link an annotation to a topic in a semantic model 103. A semantic model mapper 160 is needed to map from a semantic model representation 103 (e.g. ontology, topic maps ... etc) to the topic model used by the system 100. The semantic model mapper 160 may be seen as providing a mapped semantic model 165, which is a translation of the semantic model 103
- 20 into a format consistent with the structure expected by the complementary indices creator 124. It will be appreciated that this mapped semantic model 165 need not exist as a permanent entity, but rather represent the capacity of the semantic model mapper 160 to provide translations from the semantic model 103 in the required format on demand. The complementary indices creator 124 uses the semantic model mapper 160 to retrieve the
- 25 information needed from the semantic model 103 and add it to one or more complementary indices 150.

- In order for search applications to access and use the information stored in the complementary indices 150, there is provided an interface 180. This interface 180
- 30 complements the Search runtime providing an API to submit queries and retrieve results from the complementary indices and possibly also the main index. This API is the Complementary indices' API component. The system described herein stores semantic information in multiple indices. These indices are referred to collectively as the "complementary indices" 150.

According to certain embodiments, there may be provided a search application (not shown), running for example at a client side, to provide the end-user with a GUI that contains controls and views to support semantic search visualization.

5

The Semantic Model Sub-System:

In order to enable semantic search visualization and discovery, semantic information needs to be stored in a semantic model representation 103 that enables different types of semantic
10 information to be represented.

Exemplary semantic model elements include:

- **Topic:** anything that can be spoken about or conceived of by a human being (e.g. Egypt, Cairo, Arab League ... etc). It could have multiple names (e.g. IBM and
15 International Business Machines) and a topic name may not be unique (e.g. Alexandria of Egypt and Alexandria of Italy).
- **Topic Type:** the class of topics that a particular topic belongs to (e.g. Country, City, Organization ... etc).
- **Attribute:** any information that is specified by a topic type as being relevant to its
20 topic instances (e.g. Population, GDP ... etc).
- **Relation:** a way that topics can be related to one another (e.g. Cairo is capital of Egypt).
- **Topic identifier:** a token that provides an unambiguous indication of the identity of a
25 topic.

25

A semantic model sub-system provides the link between the annotation space, e.g. the CAS 126 and the topic space, e.g. the semantic model representation 103. It allows the annotations produced by the domain specific annotators 1222 to be linked to topics in a semantic model 103. Then the relevant information is stored in the indices 150 using a
30 uniform representation that conforms to the semantic model elements explained above. The semantic model sub-system contains two components:

Semantic Model Mapper 160- Semantic information can be represented using different model representations (e.g. topic maps, RDF, Ontolingua ... etc.). The "Semantic model

mapper" component 160 maps the schema of the underlying semantic model representation 103 into the schema understood by the complementary indices creator 150 and the annotations linker 170 (i.e. Topics, Attributes, Relations ... etc).

- 5 The implementation of a Semantic model mapper component 160 is dependant on the semantic model 103 to be mapped. There is provided an interface to allow using different implementations for different semantic model representations.

10 The idea of the semantic model mapper 160 is to enable using any semantic model representation in the system 100. According to certain embodiments the semantic data is transformed and represented using the classes: Topic, Attribute, and Relation.

Annotation Linker 170- The "Annotation linker" component 170 attempts to link an annotation stored in the CAS 126 to a topic in a semantic model 103 of a specific domain.
15 The annotation linker 170 communicates with the model through the Semantic model Mapper. The annotation linker 170 analyzes the information in the annotation (i.e. annotated text (surface form), UIMA type, and annotation features) and the semantic model 103 in order to collect sufficient evidence that this annotation represents a certain topic in the mapped semantic model 165.

20

As the annotation linker 170 analyzes the information provided in the annotation and attempts to find a matching topic, the annotation linker component 170 should understand how the information is structured in both the UIMA type system, that defines the annotation structure, and the semantic model 103, that defines the topic structure.

25

As the information structure can be different for the annotation structure (i.e. created by different annotators 1221, 1222) or the semantic model 103 (i.e. collected from different sources), the implementation of the annotation linker 170 should consider such differences.

30 There may be provided an interface to allow the plugging of annotation linker code. The annotation linker code is developed for a specific UIMA type system definition and a specific semantic model 103. The linker code analyzes the annotations created by the domain specific annotator(s) 1222 using the UIMA type system definition and attempts to match it to a topic in a semantic model 103 for the same domain. Multiple implementations

can be developed for the different domains for which annotators 1222 and semantic models 103 exist.

First embodiment- Topic and Mention indices

- 5 Figure 2 shows details of the complementary indices 150 in accordance with a first embodiment. According to this first embodiment, there are defined two complementary indices, each comprising entries defined as a key/values pair:
Specifically, there is provided a Topic index 252, whose keys are topic identifiers, each key's associated values being the type, names, attributes, and topic identifiers for related
10 topics.

There is further provided a Mention index 254, whose keys are topic names, the value of each key being the topic identifier(s) for topic(s) this name refers to.

- 15 The complementary indices 252, 254 need to be built and provided to the search applications by the search engine.

Figure 3 shows a flowchart describing certain steps implementing the first embodiment.

- 20 1. The Search Engine Crawler 110 runs periodically to retrieve new and updated artefacts 101 at step 205.
2. The Parser processes the artefacts to retrieve the data needed to build the indices 140, 150. This is performed in two stages:
- 2.1 The standard lexical analysis annotator 1221 performs the NLP tasks performed by
25 a typical search engine at step 210. The results are stored in an intermediate form (processed artefacts 240) that the search engine indexer can operate on to create the main index 140.
- 2.2 The Domain specific annotators 1222 process the documents and annotate the topic
30 mentions at step 215. The annotations contain the attribute values for the specific topics being annotated. The annotations are stored in the CAS 126. This step thus constitutes associating type annotations with elements of one or more source artefacts, where type annotations indicate that adherence to one of a set of predefined types.

3. The complementary indices creator 124 analyzes the domain specific annotations in the CAS 126 and builds the complementary indices 150. This is done in three stages:

3.1 The Annotation Linker component 170 links an annotation to a topic at step 220.

5 The Annotation Linker maps the semantic model 103 to the topics model representation used by the system. In particular, the Annotation linker serves to identify in a semantic model 103, semantic keys corresponding to the annotations associated with each element annotated at step 215, semantic model 103 comprising a plurality of said semantic keys each being associated with a plurality of semantic descriptors.

10 According to a preferred embodiment, the annotation linker 170 uses a Semantic Model Mapper 160 as described herein to map the semantic model 103 to the topics model representation used by the system 100.

3.2 The complementary indices creator retrieves the semantic information associated with a topic and maps it into semantic descriptors at step 225, thereby associating
15 the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, with that element.

According to a preferred embodiment, the complementary indices creator uses the Semantic Model Mapper component to retrieve the semantic information associated
20 with a topic and maps it into semantic descriptors. Preferably, as described above, step 225 of associating the semantic descriptors associated with each semantic key identified as corresponding to the annotations associated with a respective element, with that element, is implemented by compiling a complementary Topic index 152 belonging to the complementary indices 150, storing a reference to each said
25 element and said associated semantic descriptor.

Finally at step 230 the search engine indexer 130 creates the main index 140.

Preferably, as described above, step 215 of associating type annotations with elements of
30 one or more said source artefacts is carried out by means of an UIMA annotator or a UIMA aggregate analysis engine, where the resulting annotations are stored in a UIMA Common Analysis Structure, and where said steps 220 of identifying semantic keys and 225 of associating the semantic descriptors, are carried out in the role of CAS consumers.

Preferably, as described above, the key of step 225 comprises a topic identifier and the
5 semantic descriptors include at least one of the type, names, attributes, and topic identifiers
for related topics.

Preferably, the key associated with a semantic descriptor in said complementary index is a
topic name and the semantic descriptors include the topic identifier for topics this name
10 refers to. The complementary indices 150 thus comprises the topic index 152.

Second embodiment- Co-occurrence index

Figure 4 shows details of the complementary indices 150 in accordance with a second
embodiment. In this embodiment, the system description is the similar as that of the 1st
15 embodiment described with respect to figures 2 and 3. According to this embodiment, an
additional complementary index is added to relate topics together if they frequently co-
occurred in the collection of documents. This allows discovering the relation between the
topics even if they are not related in the semantic model. This further complementary index
is referred to as the "Co-occurrence index 254. An index entry may take the form of a
20 key/value pair, wherein:

- The key is a topic identifier.
- The associated value is a list of topic identifiers for topics that co-occurred at least once
in a document with the entry topic. The topic identifiers are ranked based on how
frequent they co-occurred in documents, given a selected criterion of how close a
25 co-occurring topic was to the entry topic (i.e. same sentence, same paragraph or
same document). Topics in this list may not have a relation with the entry topic in a
semantic model.

In order to build and use the Co-occurrence index 254, two other components may need to
30 be modified from the system in 1st Embodiment:

- Complementary indices creator 124: this component creates the Co-occurrence
index by tracking the topics that co-occurred within a specified boundary. The
boundaries are defined during lexical analysis by the lexical annotator.

- Interface APIs 180: a new API is added to the interface APIs that receives a topic identifier and a boundary (sentence or paragraph or a document) then returns topic identifiers of the co-occurred topics. It adds to the visualization functionality by enabling the search application to link a topic to its co-occurring topics, which allows a richer graph representation as in the following example:

Thus according to the present embodiment at the step 225 of associating the semantic descriptors, the key is a topic identifier and the semantic descriptors include a list of topic identifiers for topics that co-occurred at least once in an artefact.

10

By way of illustration of the benefits arising from the co-occurrence index, it may be imagined that a search for the word "Ethiopia" in the topic index alone, may produce results such as

Somalia (borders with)

15 WHO (Member of)

IAEA (Member of)

ILO (Member of)

Sudan (borders with)

IMF (Member of)

20 UN (Member of)

Africa (Contained in)

Addis Ababa (Contains)

etc.

- 25 However, but further consulting the co-occurrence index, further results that are not associated with Ethiopia in the semantic model but which frequently co-occur in the parsed artefacts may also be identified. By this means the limitations of the semantic model in terms of the level of detail it can represent, and how frequently it can be updated can be alleviated. On this basis, further hits such as:

30

horn of Africa

cradle of humanity

spicy food

may be identified.

The user can change the scope of the co-occurrences by selecting the proximity of the co-occurring word to the search term for example, same Sentence, same paragraph same document, a given number of words, characters etc. This factor may be referred to as the scope of the co-occurrence search.

Third Embodiment- attribute index

Figure 5 shows details of the complementary indices 150 in accordance with a third embodiment. In this embodiment, the system description is the similar as that of the 1st embodiment described with respect to figures 2 and 3. An additional complementary index can be added to allow the search application to populate the search fields with values for topic attributes. This helps to simplify building the search query. This further complementary index is referred to as an "Attribute index". An entry in this index may take the form of a key/value pair, wherein

- The key is an attribute name for a specific topic type.
- The associated value is a list of this attribute values in all topics of the given topic type.

In order to build & use the "attribute index", two components may need to be modified from the system of the 1st Embodiment,:

- Complementary indices creator 124: This component will create this index while the "complementary indices creator" is looping on the annotations linked to the topics.
- Façade APIs 180: an additional API is added to the "Façade APIs" that receives an attribute and returns all possible values for this attribute. This allows a new functionality which is to populate the GUI query fields with attribute values so it simplifies the search query. So a user can now search the attribute name and value.

Thus according to the present embodiment at the step 225 of associating the semantic descriptors, the key is an attribute name for a specific topic type and the semantic descriptors include a list of this attribute values in all topics of the given topic type.

Fourth Embodiment

Figure 6 shows a block diagram of a system in accordance with a fourth embodiment.

In this embodiment, the system description is the similar as that of the 1st embodiment described with respect to figures 2 and 3. According to this embodiment, certain annotators 1221, 1222 create annotations based on dictionary lookup to annotate topic mentions in text, and use dictionaries 690 built from the semantic models integrated with the system, instead
5 of a pre-built dictionary by a dictionary builder 695. An entry in a dictionary 695 may contain:

- Key: This key is a sequence of characters. It represents a topic name.
 - Value: The associated value is a list of topic ids & topic types for topics with the name in the key.
- 10 The “Keys” are gathered from the topics’ names of the semantic model 103 and the values are gathered from the topics’ identifiers and types.

A further difference is related to the Annotation Linker 170. An annotation created by the dictionary based annotator contains a feature that represents a topic identifier. The
15 “Annotation Linker” matches a “Topic ID” of a topic in the semantic representation with the annotation feature of “Topic Identifier”.

Figure 7 shows a flowchart for the modified parsing & annotation linking processes according to the fourth embodiment. As shown, the method starts at step 710, and proceeds
20 to step 720 at which the Lexical analysis annotator 1221 analyzes document text and produce boundary annotations (i.e. paragraph, sentence and token annotations). The method next proceeds to step 730 at which the dictionary based annotator 1222 uses the token boundaries to perform dictionary lookups in the semantic model dictionaries 690. When a match is found for a token (or multi-tokens), the dictionary based annotator 1222 creates an
25 annotation per identified topic type. The created annotations contain a feature that represents the topic ID. Finally the Topic ID annotation linker 170 links the annotations to topics in the semantic model 103 at step 740. The method then terminates at step 750

Figure 8 shows a flowchart for the behaviour of the modified Topic ID based annotation
30 linker according to the fourth embodiment. The method starts at step 810, and proceeds to step 820 at which it is determined whether there remain any annotations to be linked. In a case where no annotations remain to be linked, the method terminates at step 830. Otherwise, the method proceeds to step 840 at which the next annotation to be linked is obtained. The method proceeds to step 850 at which it is determined whether the annotation

to be linked has a "TopicID" feature, representing a particular a Topic ID. If the feature exists, the method proceeds to step 860 at which the Annotation Linker 170 directly links the annotation to a Topic in the semantic model 103 with the matching ID before returning to step 820. Otherwise the method proceeds directly to step 820. The method thus checks
5 each annotation to be linked in turn until none remain.

Example of Semantic Search Visualization and Content Discovery

In the following example, it is shown how a semantic model might be used with
10 annotations created with reference to entities in a text to build the search indices used to support semantic search visualization and content discovery, to illustrate certain aspects of the foregoing embodiments.

Figure 9 shows a subset of the information provided in a Topic Map representation of a
15 semantic model 103. According to this example, the semantic model 103 contains information about countries – cities – political, financial and social organizations – religions – geographical information (i.e. rivers, lakes, seas, oceans, islands, continents ... etc). The Semantic Model Mapper 160 is used to convert this Topic Map to a semantic model representation 165 compatible with the Complementary Indices Creator 124.

20 According to the representation format used in figure 9, there are provided a number of cartouches 911, 912, 913, 914, 915, 916, 917, 918, 919 etc., each representing a Topic, such as "India", "Contained in", "Country", "Egypt", "is capital of", "Capital", "Cairo", "Is seat of", "Population" respectively. There are further provided a number of circles 921, 922, 25 923, etc., each indicating an association role. Boxes 931, 932, 933 etc. indicate occurrences as described hereafter. Diamonds such as 941, 942 etc define associations, by connecting different topics with reference to an association role. Finally, broken arrows indicate that a topic, an association, an association role or an occurrence is an instance of the topic to which the arrow points. Thus by reading a particular group of symbols connected to an
30 object information can be derived concerning that object. For example, the group of symbols 917, 923, 916, 923, 941, 915, 922, 914, 913 tells us that Egypt is a country, and Cairo is its capital.

Figure 10 shows the mapped Semantic Model 165 from the previous Topic Map 103 shown in figure 9. As shown there are defined a plurality of Topics 1010, 1020, 1030, 1240, 1050, 1060. Each topic in this representation contains respectively a unique identifier 1011, 1021, 1031, 1041, 1051, 1061 such as Top1 for 1011, Top2 for 1021, ...etc, which is stored in the
5 Topic Map XML file, a list of names 1012, 1022, 1032, 1242, 1052, 1062, its topic type 1013, 1023, 1033, 1243, 1053, 1063, list of attributes 1014, 1024, 1034, 1244, 1054, 1064, and list of relations with the other topics 1015, 1025, 1035, 1245, 1055, 1065.

As described above, domain specific annotators 1222 are used during the parsing phase to
10 process the artefacts 101 and create annotations over semantic entities in the text in the artefact. In this example we use an annotator 1222 that creates annotations that identifies "Countries" and "Cities" in the text.

The present example will proceed on the basis of the following text:
15

Sample text 1.

"IBM inaugurated a Global Delivery Center (GDC) in **Pune** to provide clients worldwide with business consulting and application services. Located in Hinjewadi Information Technology Park, the center is spread over 180,000 square feet and will house close to
20 2,000 employees once fully staffed.

In addition to the **Pune** GDC, IBM also announced a new global delivery center in **Cairo, Egypt**. The **Pune** and **Cairo** sites will be an integrated part of IBM's vast network of business consulting and application services centers, located in eight countries around the world."
25

A domain specific 1222 annotator creates "City" annotations with respect to mentions of "Pune" and "Cairo". It also creates "Country" annotation with respect to mention of "Egypt".

30 Figure 11 shows an example for the annotations created by the domain specific annotator 1222. The Complementary Indices Creator 124 uses the Annotation Linker 170 to link annotations to matching topics from the mapped semantic model 165. Annotation Linker processing may be provided according to different approaches. Type systems acts as the contract between the Annotation Linker 170 and the domain specific annotator 1222. Also

the Annotation Linker has access to the Mapped Semantic Model 165 through the Semantic Model Mapper 160.

As an example, a rule-based implementation of Annotation Linker 170 may use some rules
5 to make the link. Rules can be fired based on logical expression which combines covered text, annotation types, features and their values, along with topic identifiers, relations, names and attributes.

A simplified example set of rules may contain - among other rules - the following:
10

...

IF Annotation.type EQUALS uima.tt.City AND Topic.Type EQUALS Country AND
Topic.Nmes EQUALS Annotation.coveredText

15
IF Annotation.type EQUALS uima.tt.Country AND Topic.Type EQUALS City AND
Topic.Nmes EQUALS Annotation.coveredText

IF Annotation.type EQUALS uima.tt.Province AND Topic.Type EQUALS province AND
20 Topic.Nmes EQUALS Annotation.coveredText AND Annotation.Country EQUALS
Topic.ContainedIn.

...

25 The grammar behind these rules:

- 1- IF: is conditional branching, where IF checks some conditions and if all these conditions passed then it do the statements in
- 2- Annotation and Topic means an instance of the Topics and Annotation, so you can access its attributes using the dot operator.
- 30 3- EQUALS checks if the right hand side's value equals the left hand side's value.
- 4- AND/OR logical operators

Given the above mentioned rules, the rule-based annotation linker can link annotations (figure 11), and topics (figure 10) in the following manner:

- "Annotation 1" to "Top1" representing city of "Cairo" (1110 in fig. 11)
- "Annotation 2" to "Top3" representing city of "Pune" (1120 in fig. 11)
- "Annotation 3" to "Top2" representing country of "Egypt" (1130 in fig. 11)

5 After annotations are linked to matching topics, the Complementary Indices Creator 124 fills the four indices Co-occurrence Index, Topic Index, Mention Index, and Attribute Index for each annotation 1110, 1120, 1130 as follows:

- 1- **Topic Index:** The key is set to matching topic identifiers and value is filled with matching topic names, type, attributes and relations from the Mapped Semantic Model 165. Table 1 below shows how the information that would be added to the topic index 252 on the basis of the present example.

Table 1.

Key: Topic identifier	Values: Topic names, type, attributes and relations	
Top3	Names	Pune
	Type	City
	Attributes	Population: 1566651
	Relations	Contained In: Top5
Top1	Names	Cairo
	Type	City
	Attributes	Population: 6053000 Latitude: 29.8667 Longitude: 31.3333
	Relations	Is Seat Of: Top4 Is Capital Of: Top2
Top2	Names	Egypt
	Type	Country
	Attributes	Population: 63575107
	Relations	Capital: Top1
... etc	... etc	

- 2- **Mention Index:** Key is set to annotated text, and the value is filled with identifiers for matching topics Table 2 below shows how the information that would be added to the Mention index 254 on the basis of the present example.

5 **Table 2**

Key: Annotated text	Values: Matching topics identifiers
Pune	Top3
Cairo	Top1
Egypt	Top2
... etc	... etc

- 3- **Attribute Index:** Key is set to attributes names contained in mentioned topics, and value is filled with all possible attribute values. Table 3 below shows how the information that would be added to the Attribute index 558 on the basis of the present example.

10 **Table 3.**

Key: Attribute name	Values: Possible attribute values
Population	1566651, 6053000, 63575107
Latitude	29.8667
Longitude	31.3333
... etc	... etc

- 4- **Co-occurrence Index:** Key is set to mentioned topics identifier, and value is filled with topic identifiers for topic which co-occurred with key's mentioned topic. Table 4 below shows how the information that would be added to the co-occurrence index 456 on the basis of the present example.

20 **Table 4.**

Key: Topic identifier	Values: Co-occurring topics identifiers
Top1	Top2, Top3

Top2	Top1, Top3
Top3	Top1, Top2
... etc	... etc

Once the above complementary indices 252, 254, 456, 558 are created, a semantic search application can make queries to these indices via the interface 180 at runtime to help the
5 end-user in visualizing the relations among topics and further knowledge discovery as explained in the disclosure.

On the basis of the described complementary indices, a wide range of search functions may be supported, e.g. via the interface 180. For example,

10 **Search:** Keyword search: a list of documents matching the search query is returned from the main index.

Semantic search: topic ids are retrieved for topics matching the search query.

Also, a list of documents that contains mentions for these topics are returned. Moreover, upon request, topics information is retrieved based on
15 topic ids. This functionality combines information from the Main index (matched documents), Mention index (matched topic ids), and the Topic index (topics information).

Topic detection: in keyword search, there is provided a method to determine if the
20 keyword matches a topic. A search is performed in the Mention index and a list of topic ids is returned for matching topics.

Visualization: given topic id for the topic to be visualized, a graph connecting this topic to all topics that is directly related to this topic in the semantic model.
25 Information in this graph enables search application to provide search refinement and discovery capabilities through the semantic visualization. Information is retrieved from the Topic index.

Clustering: this functionality allows classification of topics into groups based on a
30 specific attribute value. Given the topics to be classified and the attribute to be by which they are to be classified, the value of this attribute will be retrieved from the

Topic index for each topic, and the topics having the same value for the given attribute will be grouped.

- Topic information retrieval:** this functionality allows for retrieval of names, attributes and relations for a specific topic. This helps when visualizing a topic.

Based on the above exemplary functions, a search sequence may be envisaged along the following lines:

A user submits a search query "countries contained in Africa".

- The search application builds a query string (that follows the search engine query syntax) and submits it to the search engine using the interface 180 (*Topics retrieval API and Document retrieval*).
- The search results are then returned to the search Application. As multiple topics match the search query, a list of topic names is displayed to the user. Also documents matching the search query are displayed to the user.
- The user can see a graph of the matched topic relative to a grouping attribute. The user selects the grouping attribute. The search application uses the "Topics clustering" function and presents a graph to the user with the matched topics in relation to the grouping attribute.
- The user can select one of the topics in the matched topics graph or list. The search application uses the "Topic relations" function from the Façade APIs to retrieve a data structure with the selected topic and its related topics. The search application presents a graph with the topic relations. The search application also uses the "Document retrieval" function to return the documents matching the selected topic.
- The user can hover on the selected topic. The search application uses the "Topic information retrieval" function and presents a tooltip with the topic information (Topic names, type and attributes).
- The user can select any topic on the graph and then its information is presented to the user using the same methodology.

This example highlights the visualization functionality enabled by the invented system which allows a richer graphical representation and a user-friendly experience for search and discovery.

According to certain embodiments, there is provided a method of processing source document to provide complementary semantic databases to enable an improved semantic search. Source documents are parsed and type annotations indicating adherence to one of a set of predefined types are associated with particular words in the source documents. For
5 each annotated word semantic keys are identified that corresponding to the annotations associated therewith. These semantic keys correspond to semantic keys defined in a semantic model, which associated each semantic key with a plurality of semantic descriptors. On this basis a complementary database is compiled associating each word with the semantic descriptors associated with each semantic key thus identified as
10 corresponding to the annotations associated with a respective word. A number of different complementary databases are proposed.

A number of ways of associating the semantic descriptors associated with each semantic key identified as corresponding to the annotations associated with a respective element,
15 have been described, in the form of different complementary indices. It will be appreciated that other such associations will readily occur to the skilled person, and that different combinations of such associations comprising some or all of the associations described in detail above, other such associations are possible.

20 The detailed embodiments described above have been described with reference to the detailed system structure of figure 1. It will be appreciated that this detailed system structure can be changed in many ways- for example, certain modules may be merged or split, different functions may be carried out at different times, and certain modules may be omitted or replaced.

25 The invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

30 Furthermore, the invention can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer readable medium can be any apparatus that

can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk – read only memory (CD-ROM), compact disk – read/write (CD-R/W) and DVD.

A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

Claims

1. A method of processing source artefacts, said method comprising the steps of:
- 5 associating at least one type annotation with an element of one or more said source artefacts, where each said type annotation indicates adherence to one of a set of predefined types;
- for each element thus annotated identifying semantic keys corresponding to the annotations
- 10 associated therewith in a semantic model, said semantic model comprising a plurality of said semantic keys each being associated with a plurality of semantic descriptors;
- associating the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, with that element.
- 15
2. The method of claim 1 wherein said step of associating type annotations with elements of one or more said source artefacts, where said type annotations indicate that adherence to one of a set of predefined types, is carried out by an UIMA annotator or a UIMA Aggregate analysis engine, where the resulting annotations are stored in a UIMA
- 20 Common Analysis Structure, and where said steps of identifying semantic keys and associating the semantic descriptors are carried out in the role of CAS consumers
3. The method of claim 2 said step of associating type annotations with elements of one or more said source artefacts is carried out by means of a UIMA Aggregate analysis engine
- 25 comprising a Lexical analysis annotator, which performs the Natural Language Processing (NLP) tasks for the traditional search, and Domain Specific Annotators that analyze the documents based on semantic information for a specific domain.
4. The method of any preceding claim wherein said step of associating the semantic
- 30 descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, with that element, is implemented by compiling a complementary index storing a reference to each said element and said associated semantic descriptor.

5. The method of any preceding claim comprising the further steps of retrieving artefacts from a specified source and parsing said artefacts.
6. The method of claim 5 where said step of parsing includes at least one of language
5 identification, tokenization, stemming, part-of-speech tagging, and normalization.
7. The method of any preceding claim wherein at said step of associating the semantic descriptors said key is a topic identifier and said semantic descriptors include the type, names, attributes, and topic identifiers for related topics.
- 10 8. The method of any preceding claim wherein at said step of associating the semantic descriptors said key is a topic name and said semantic descriptors include the topic identifier for topics this name refers to.
- 15 9. The method of any preceding claim wherein at said step of associating the semantic descriptors said key is a topic identifier and said semantic descriptors include a list of topic identifiers for topics that co-occurred at least once in an artefact.
- 20 10. The method of any preceding claim wherein at said step of associating the semantic descriptors said key is an attribute name for a specific topic type and said semantic descriptors include a list of this attribute values in all topics of the given topic type.
11. The method of any preceding comprising the further steps of
receiving a search term;
25 with reference said association of the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, identifying all elements related to said search term; and
generating a listing of the elements thus identified.
- 30 12. A computer program comprising instructions for carrying out the steps of the method according to any one of claims 1 to 11 when said computer program is executed on a suitable computer device.

13. A computer readable medium having encoded thereon a set of computer programs according to claim 12.

14. A system for processing source artefacts, said system comprising:

5

an annotator adapted to associate type annotations with elements of one or more said source artefacts, where said type annotations indicate that adherence to one of a set of predefined types,

10 a linker adapted to identify in a semantic model, semantic keys corresponding to the annotations associated with each element thus annotated, said semantic model comprising a plurality of said semantic keys each being associated with a plurality of semantic descriptors,

15 a complementary indices creator adapted to associate the semantic descriptors associated with each semantic key thus identified as corresponding to the annotations associated with a respective element, with that element.

15. A system according to claim 14 and further comprising means adapted to
20 implement the steps of any of claims 2 to 12.

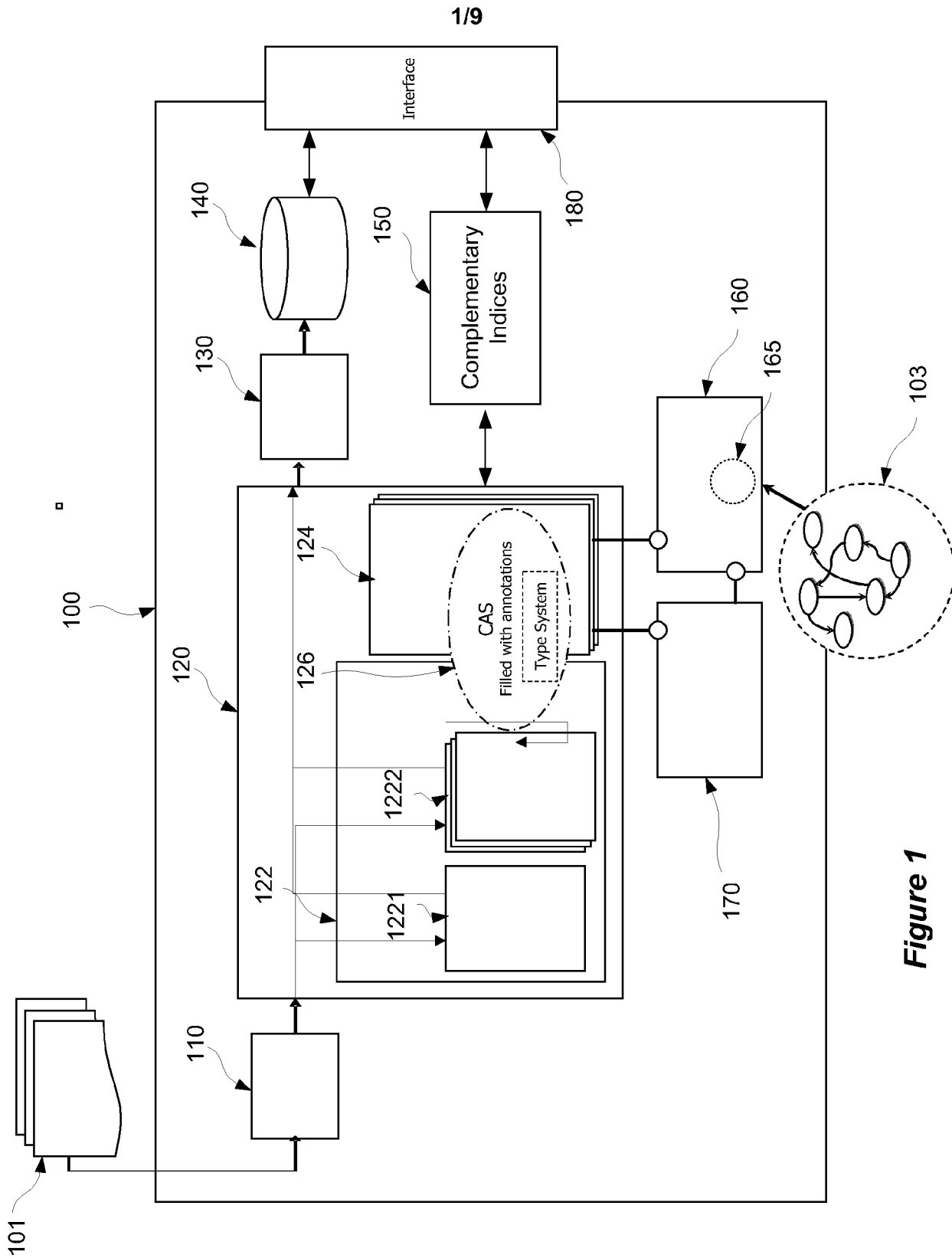
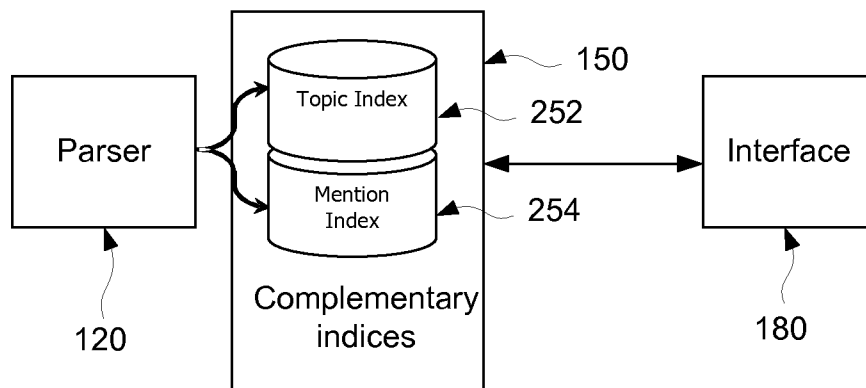
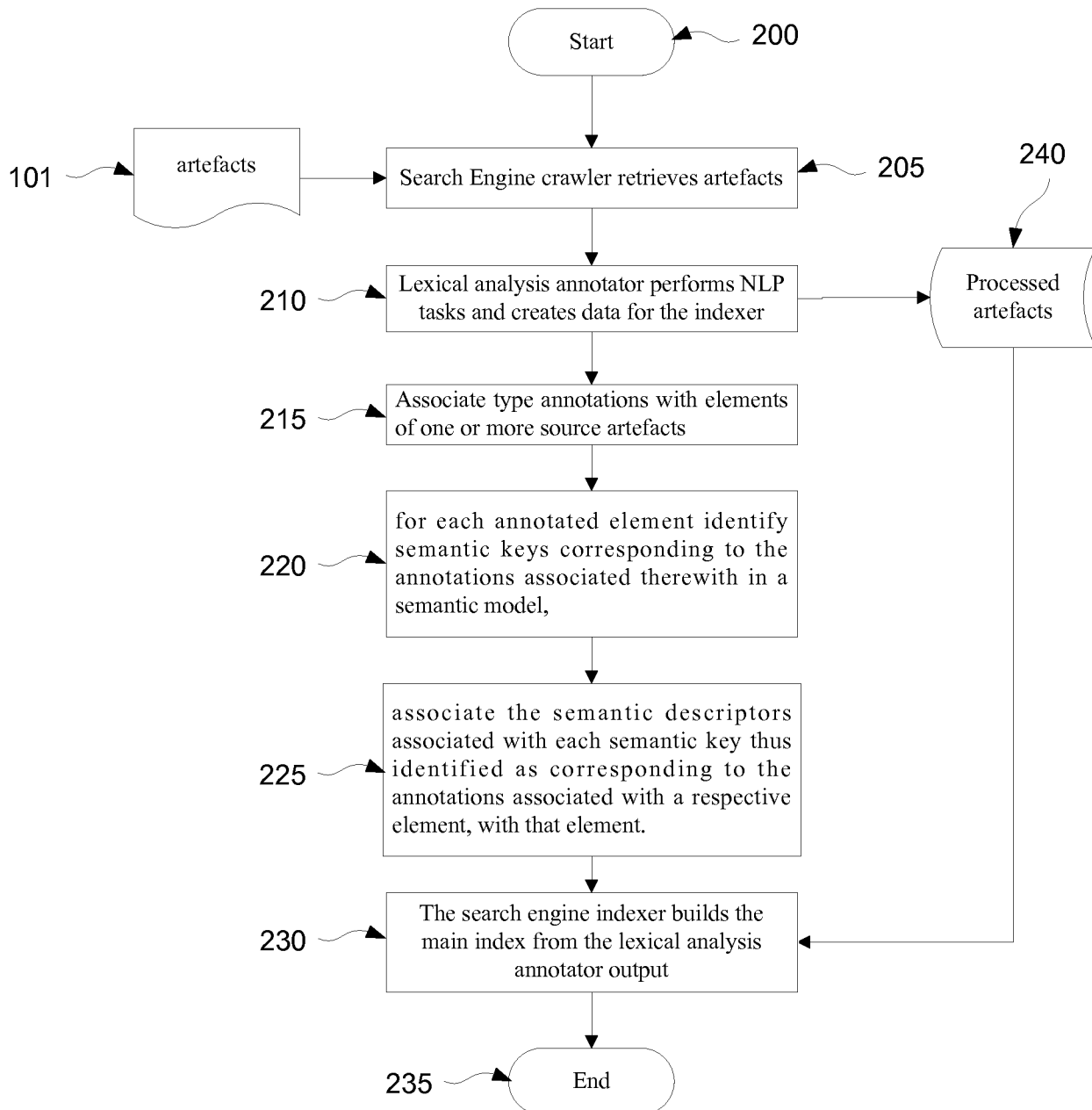
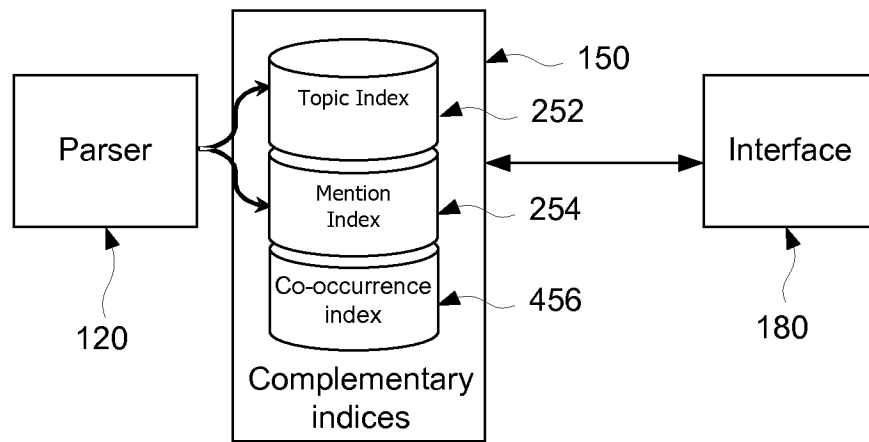
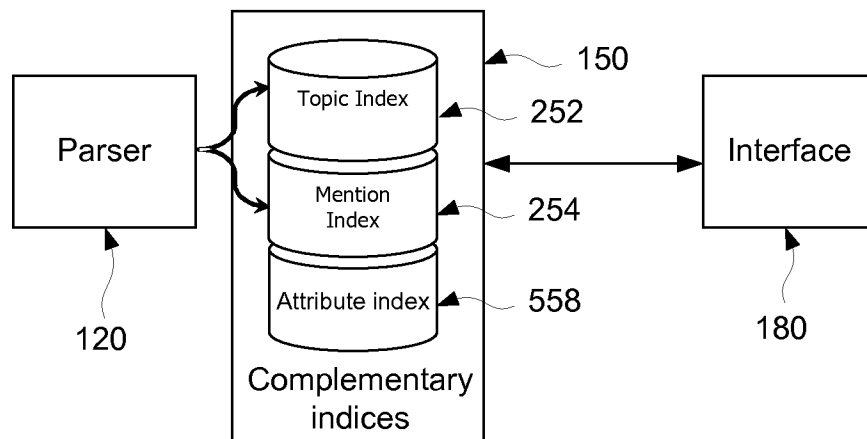


Figure 1

**Figure 2**

**Figure 3**

**Figure 4****Figure 5**

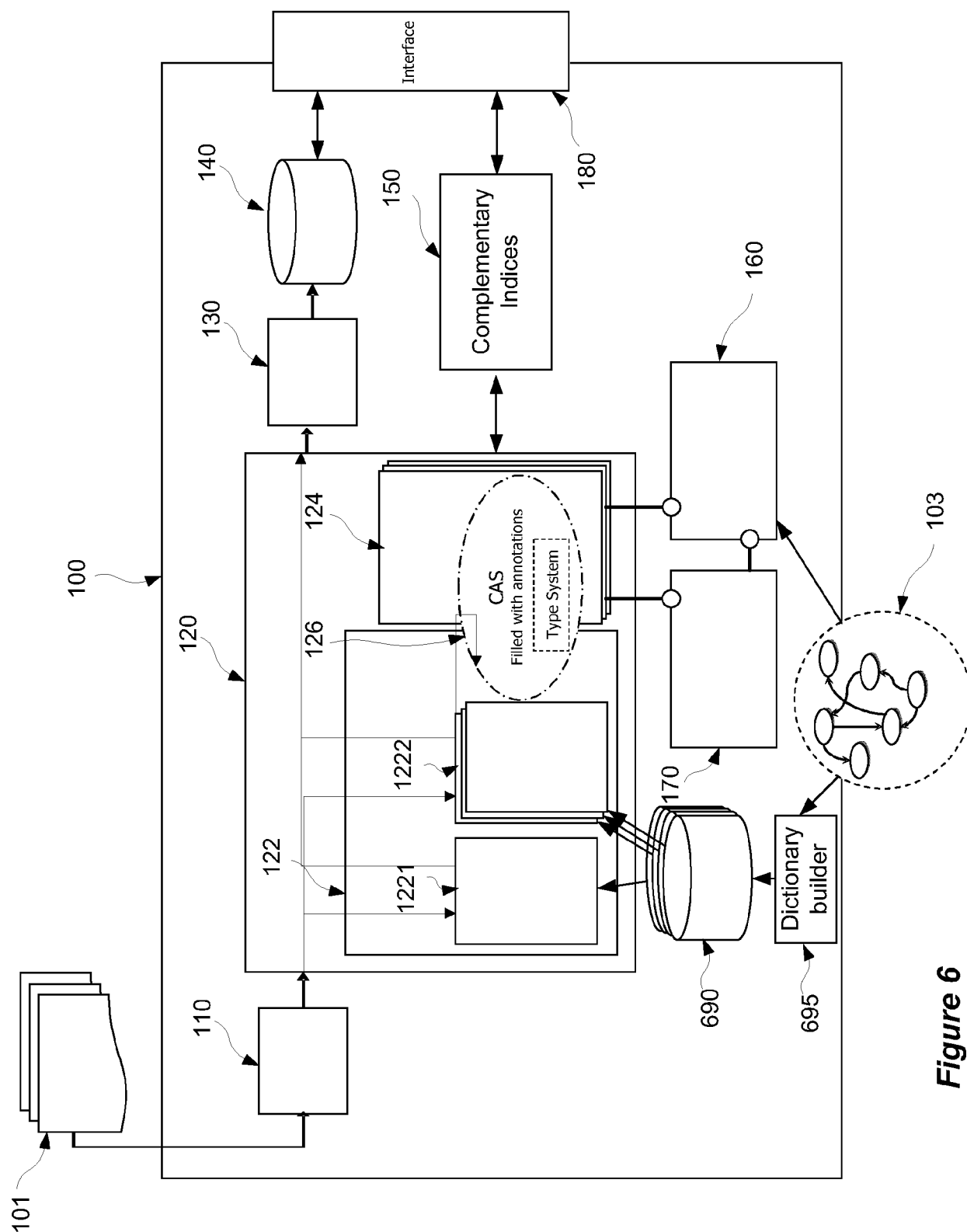
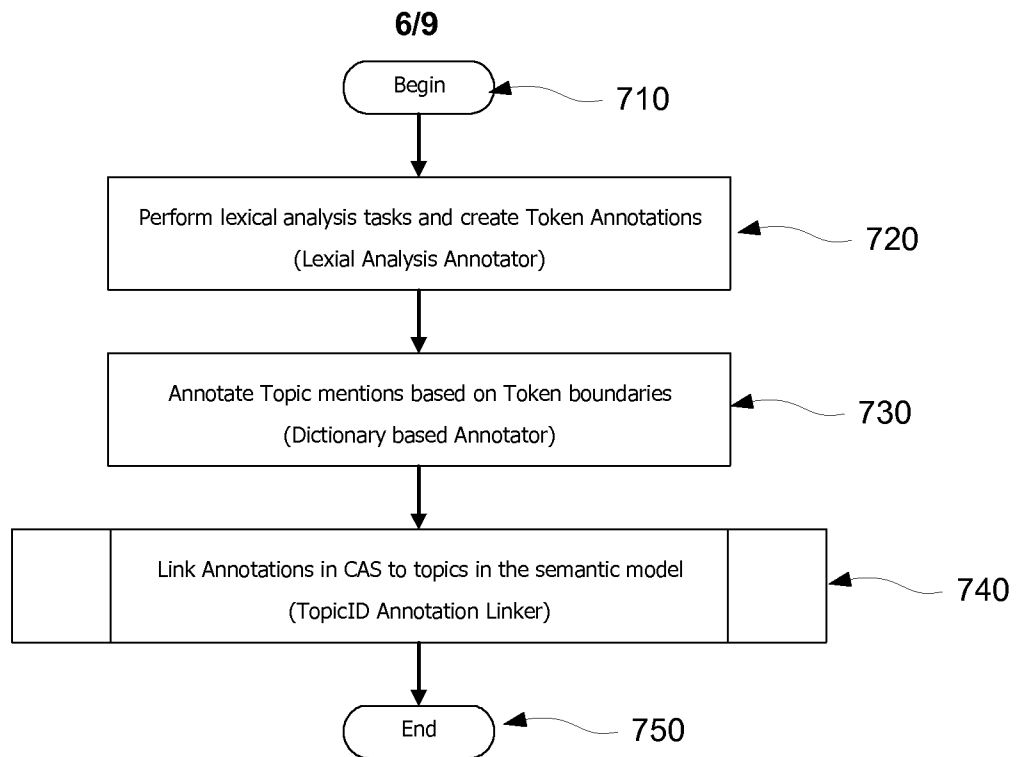
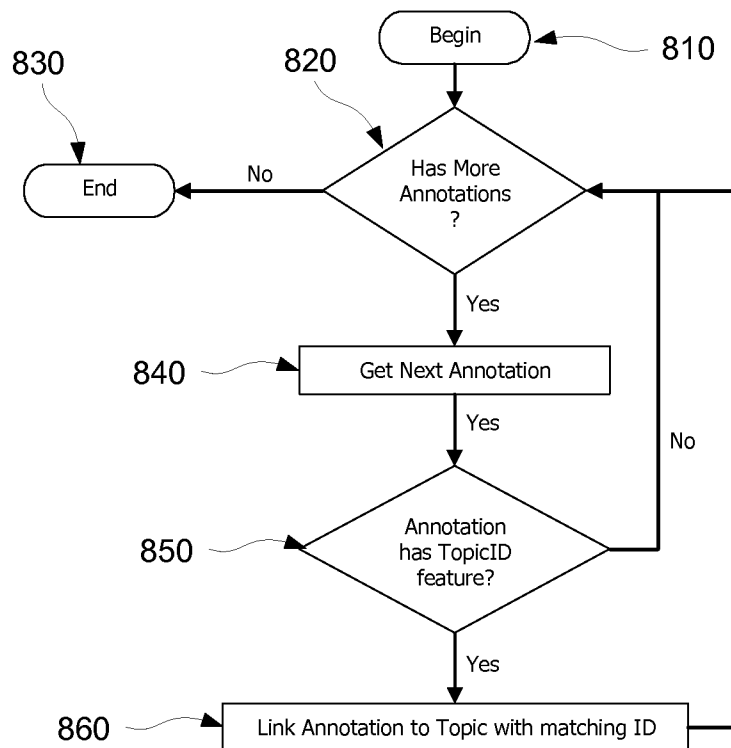


Figure 6

**Figure 7****Figure 8**

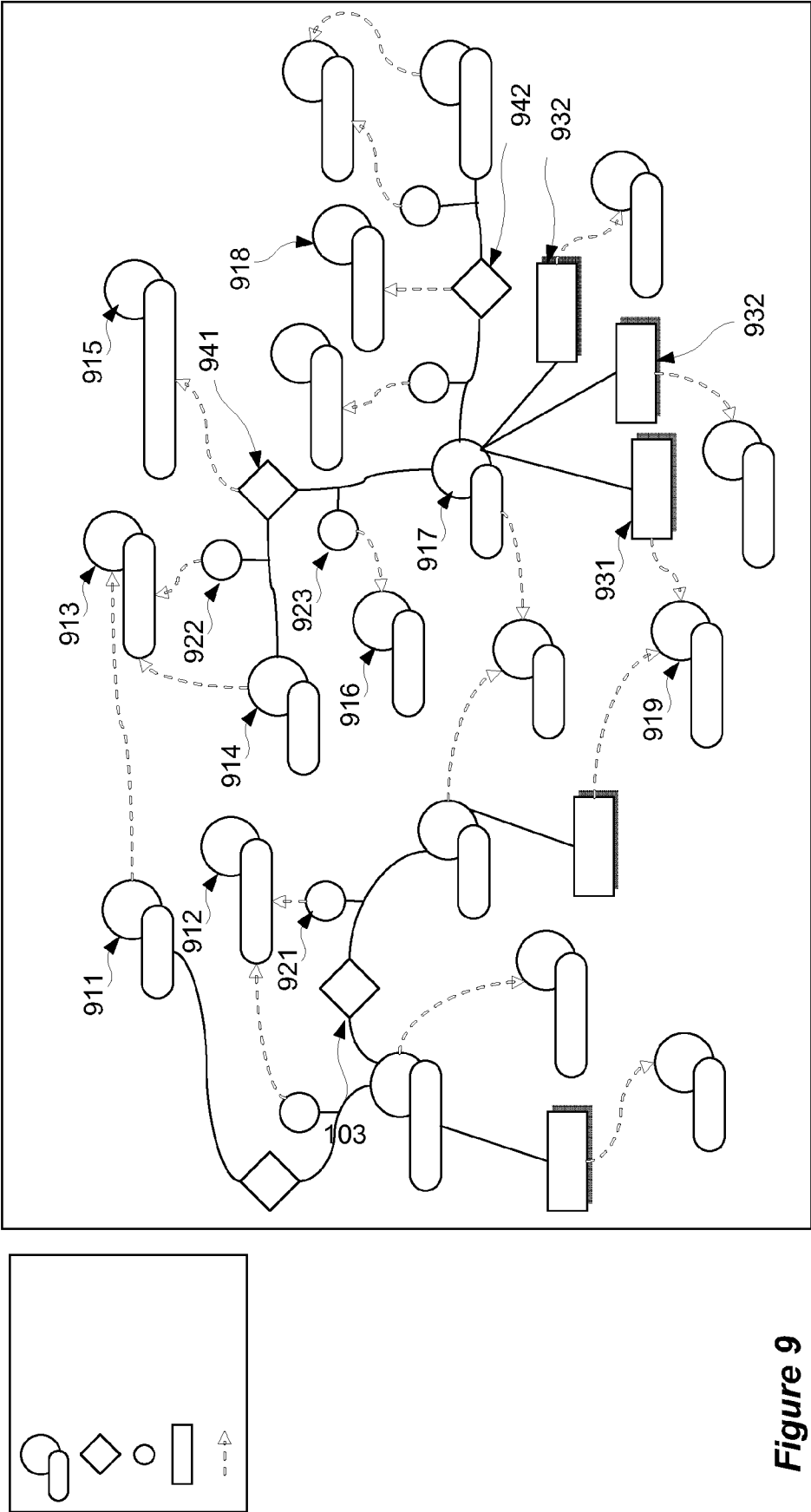


Figure 9

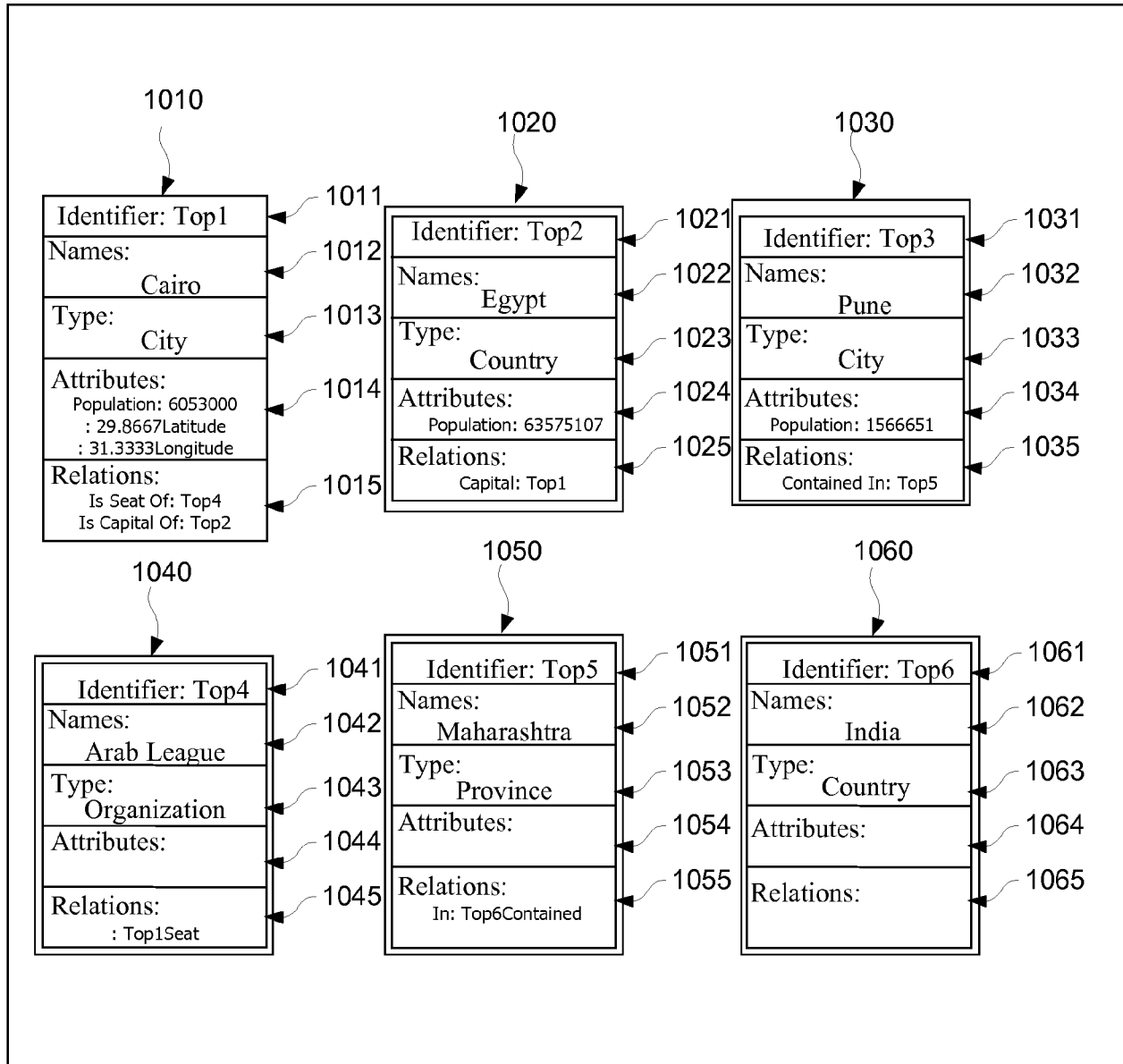
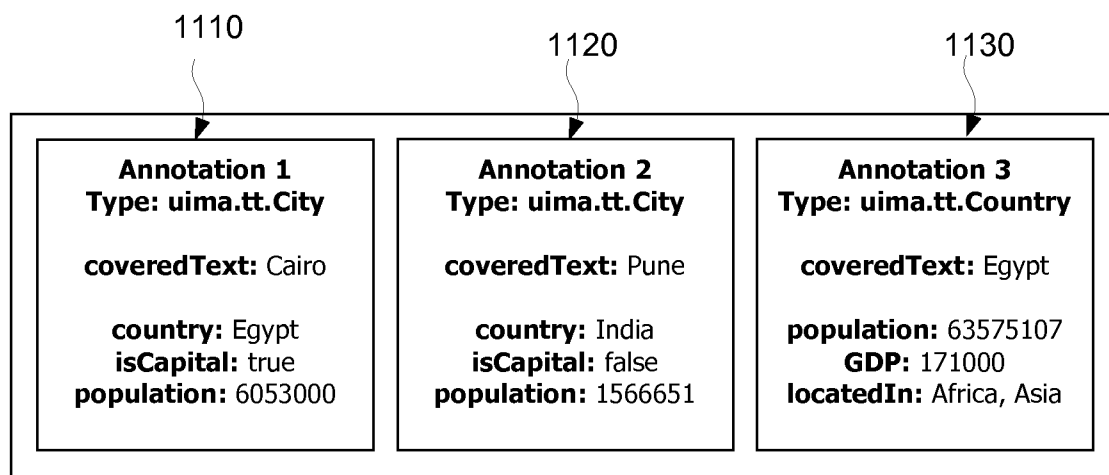


Figure 10

*Figure 11*

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/051055

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	FERRUCCI D ET AL: "UIMA: an architectural approach to unstructured information processing in the corporate research environment" NATURAL LANGUAGE ENGINEERING, CAMBRIDGE UNIVERSITY PRESS, CAMBRIDGE, GB, vol. 10, no. 3-4, 1 September 2004 (2004-09-01), pages 327-348, XP009133296 ISSN: 1351-3249 the whole document -----	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

7 May 2010

Date of mailing of the international search report

19/05/2010

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Hauck, Rainer