



- (51) International Patent Classification:
G06F 16/215 (2019.01) *G06F 16/21* (2019.01)
- (21) International Application Number:
PCT/MY2019/050093
- (22) International Filing Date:
14 November 2019 (14.11.2019)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
PI 2018001926 14 November 2018 (14.11.2018) MY
- (71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).
- (72) Inventors: **DOMINGO, Ma. Stella Tabora**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000

(MY). **ISHAK, Ros'aleza Zarina**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **GOON, Wooi Kin**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **ABDULLAH, Muhammad Raziq Ramesh**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **MOHAMED, Sa'niah**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: **PYPRUS SDN BHD**; Unit 16-08, 16th Floor, Block A, Damansara Intan, No. 1, Jalan SS20/27, Selangor, Petaling Jaya, 47400 (MY).

(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, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,

(54) Title: SYSTEM AND METHOD FOR MANAGING DUPLICATE ENTITIES BASED ON A RELATIONSHIP CARDINALITY IN PRODUCTION KNOWLEDGE BASE REPOSITORY

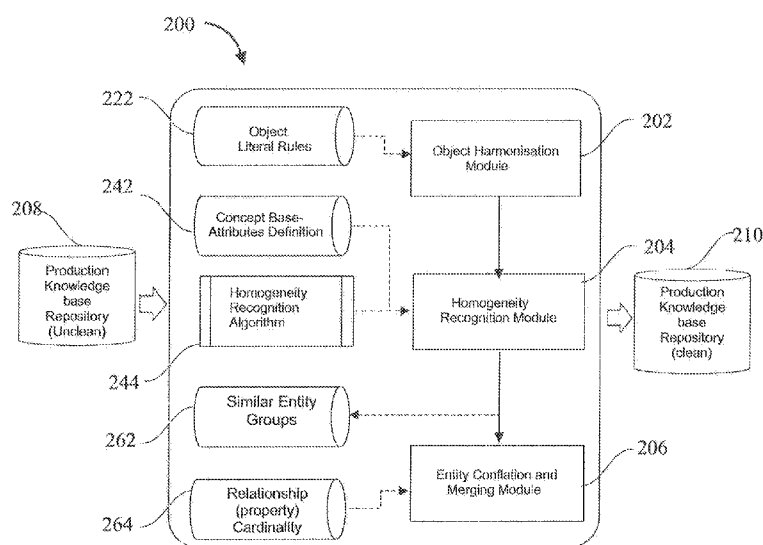


FIG. 2

(57) Abstract: Disclosed is a system and method for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository. The method comprises steps of performing a first level detection of duplicates in existing data present in the production knowledge base repository through an object harmonisation module (202). The first level detection identifies duplicates of one or more attribute objects within a specific entity. The object harmonisation module (202) implements a sanitization and standardization operation on the identified attribute objects. Then the method performs a second level detection of duplicates between entities of a specific concept through a homogeneity recognition module (204). The homogeneity recognition module (204) identifies duplicates according to base-attributes of the specific concept based on a predefined similarity threshold. The method then enables a

KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- *of inventorship (Rule 4.17(iv))*

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

SYSTEM AND METHOD FOR MANAGING DUPLICATE ENTITIES BASED ON A RELATIONSHIP CARDINALITY IN PRODUCTION KNOWLEDGE BASE REPOSITORY

5

FIELD OF INVENTION

[0001] The present invention relates to a production knowledge base repository, in particular to system and method of preserving semantic relationship and data integrity in the production knowledge base repository while merging duplicates and ensuring data
10 quality.

BACKGROUND

[0002] Typically, a database stores data which is a collection of values for multiple fields. These databases accumulate duplicate data over a period due to various reasons ranging
15 from error-prone data-entry to merge multiple databases. Further, the quality of data can have a significant impact on organizations that rely on data to function and conduct business. In today's world, many organizations utilize big data as a way to better support their companies and serve their customers.

[0003] Additionally, big data warrants innovative processing solutions to create new
20 growth opportunities, improve product quality and assist decision making. The effectiveness of big data technology relies on the accuracy of data. However, duplicate data has become one of the most common data quality errors for businesses. Further, the organizations suffer from the data loss and missing semantic relationships during data cleansing.

[0004] Hence, many initiatives have been put in place to counter the problem of duplicate
25 data. However, the existing systems and methods have many drawbacks such as poor data integrity by having similar data, utilisation of valuable disk space by duplicate data, complicated and extensive effort in the manual reconciliation of duplicate and non-useful data, time-consuming cleansing and sanitisation of data, inaccurate and inconsistent data
30 which is hard to use etc.

[0005] US patent number 8,838,549 B2 issued to Bodapati, et al. discloses a system and method for detecting and removing duplicate records. It achieves detection and removal of duplicates by finding duplicates through matching a group of fields in the data records. The method uses an application of appropriate matching functions to match information across different fields within a group selectively. However, the method for detecting and eliminating duplicates does not manage the duplicate entities based on relationship cardinality and normalizes the data to predefined fields in the knowledge base such as name, email, phone and address. Further, this reference uses the knowledge base to store predefined format, mapping, and rules. It further involves clustering of the records based on the generated keys and calculates the matching percentage of each record concerning a reference record in a cluster. Furthermore, it merges the records with non-overlapping information and purges the records having identical information in each cluster.

[0006] US patent number 8,577,899 B2 issued to Maxwell discloses a system and method to support curation of items in a searchable knowledge base. It mines one or more search queries of the searchable knowledge base. Then it uses a processor to determine one or more pairs of items, where each of the pairs of items includes a correlation value exceeding a threshold. The correlation values for the pairs of items are based upon the frequency the items of the pairs of items co-occur within the search queries. Further, a curator may manually review the pairs of items. However, the supporting curation of items in a searchable knowledge base is not able to accurately eliminate the duplication of data in a production knowledge base repository and selects the pair of items having a correlation value exceeding a threshold value. Further, the disclosed ranking is based on the frequency of pair items co-occur in the search results of search queries.

[0007] It is, therefore, a need for a reliable and efficient system and method to improve data consistency and accuracy by eliminating the duplication of data in a production knowledge base repository.

[0008] The present invention mainly deals the technical problems existing in the prior art. in response to these problems, the present invention provides a system and method of rendering data-sets for differentiating a plurality of potential threat actors.

SUMMARY

[0009] There is a need for a system and method to manage duplicate entities based on relationship cardinality in a production knowledge base repository. Furthermore, there is also a need for a system and method to preserve the semantic relationship and data integrity in the production knowledge base repository while merging duplicates and ensuring data quality.

[0010] An aspect of the present invention relates to a method for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository. The method comprises steps of performing a first level detection of duplicates in existing data present in the production knowledge base repository through an object harmonisation module. The first level detection identifies duplicates of one or more attribute objects within a specific entity. The object harmonisation module implements a sanitization and standardization operation on the identified attribute objects. Then the method performs a second level detection of duplicates between entities of a specific concept through a homogeneity recognition module. The homogeneity recognition module identifies duplicates according to base-attributes of the specific concept based on a predefined similarity threshold. The method then enables a user to determine the similarity of the entities and further enables the user to merge the similar entities through an entity conflation and merging module. The entity conflation and merging module enables the user to nominate a primary entity within a similar entity. The entity conflation and merging module implements merging of a plurality of secondary entities into the primary entity according to the relationship cardinality. The data in the secondary entities do not overwrite the data in the primary entity.

[0011] In another aspect of the present invention, the object harmonisation module further performs steps of extracting and grouping a plurality of entity triples per concept-attribute. The object harmonisation module quantifies the attribute objects for the specific entity and then identifies a predefined classification of attribute object. The classification comprises a resource description framework (RDF), a uniform resource identifier (URI), and a plurality of literals. Further, object harmonisation module includes a step of sanitizing and standardizing a format of a plurality of object values. Further, the object harmonisation module compares the object values and identifying exact duplicate of the object values. The object harmonisation module then harmonizes object data by deleting one or more

duplicate triples of duplicate object URI/literals for an entity to ensure the uniqueness of the attribute objects and updates the production knowledge base repository.

[0012] In a further aspect of the present invention, the step of sanitizing and standardizing the format of object values further comprises steps of removing of unnecessary characters from the object labels and literals, and modifying into the standard format of object labels and object literals based on a predefined object literal rules.

[0013] In yet a further aspect of the present invention, the homogeneity recognition module comprises a concept base-attribute definition for analysis of a reference scope. The homogeneity recognition module performs a step of identifying if the concept has more than one base-attribute and executing the homogeneity recognition based on a predefined similarity threshold. Then the homogeneity recognition module repeats the execution of homogeneity recognition for each base-attribute having a first base-attribute as the primary base-attribute in identifying the concepts has more than one base-attribute. The homogeneity recognition module further computes a similarity ranking as per similar entity group, and stores the identified list of similar entity group for usage of the entity conflation and merging module.

[0014] In yet another aspect, the similarity ranking is obtained by receiving a base-attribute count (X) of the specific concept, identifying the primary base-attribute for the specific concept, computing the primary base-attribute value per entity by multiplying 0.65 to X, computing the base-attribute value per entity by dividing 0.35 by the difference of X-1, and multiplying the result by X. The similarity ranking is based on the summation of primary base-attribute value and the base-attribute values per entity.

[0015] In a further aspect, the entity conflation and merging module comprises a user interface for verification. The entity conflation and merging module presents a list of similar entity groups in a concept sorted by the similarity ranking. Then the entity conflation and merging module displays relationships with the corresponding objects for each entity, enables the user to verify data, and nominates the primary entity and the secondary entity. The entity conflation and merging module further validates property cardinality as defined in the relationship cardinality reference. The entity conflation and merging module then merges the duplicate entities to ensure the preservation of the

semantic relationship. The entity conflation and merging module further updates the production knowledge base repository with cleansed data.

[0016] The merging of duplicate entities comprise steps of replacing URI of secondary entities triple with the primary entity URI based in the property cardinality, appending the data in the primary entity not contained in the secondary entities to ensure the preservation of the semantic relationship of the primary entity, and removing the redundant data triples.

[0017] A further aspect of the present invention relates to a device in a network. The device includes a non-transitory storage device having embodied therein one or more routines operable to manage one or more duplicate entities based on a relationship cardinality in a production knowledge base repository. The one or more routines include an object harmonisation module, a homogeneity recognition module, and an entity conflation and merging module.

[0018] The object harmonisation module is configured to perform a first level detection of one or more duplicates in existing data present in the production knowledge base repository. The first level detection identifies duplicates of one or more attribute objects within a specific entity. The object harmonisation module implements a sanitization operation and a standardization operation on the identified attribute objects. The homogeneity recognition module performs a second level detection of one or more duplicates between entities of a specific concept. The homogeneity recognition module identifies duplicates according to one or more base-attributes of the specific concept based on a predefined similarity threshold. The entity conflation and merging module enables a user to determine the similarity of the entities and further enables the user to merge the similar entities. The entity conflation and merging module enables the user to nominate a primary entity within a similar entity. The entity conflation and merging module implements merging of a plurality of secondary entities into the primary entity according to the relationship cardinality. The data in the secondary entities do not overwrite the data in the primary entity.

[0019] Yet, an aspect of the present invention relates to a system to manage one or more duplicate entities based on a relationship cardinality in a production knowledge base repository. The system includes a processor and a memory. The memory stores machine-readable instructions that when executed by the processor cause the processor to perform a

first level detection of one or more duplicates in existing data present in the production knowledge base repository through an object harmonisation module. The first level detection identifies duplicates of one or more attribute objects within a specific entity. The object harmonisation module implements a sanitization operation and a standardization operation on the identified attribute objects.

[0020] Further, the processor is configured to perform a second level detection of one or more duplicates between entities of a specific concept through a homogeneity recognition module. The homogeneity recognition module identifies duplicates according to one or more base-attributes of the specific concept based on a predefined similarity threshold. The processor is further configured to enable a user to determine the similarity of the entities and further enables the user to merge the similar entities through an entity conflation and merging module. The entity conflation and merging module enables the user to nominate a primary entity within a similar entity. The entity conflation and merging module implements merging of a plurality of secondary entities into the primary entity according to the relationship cardinality. The data in the secondary entities do not overwrite the data in the primary entity.

[0021] Accordingly, one advantage of the present invention is that it automatically identifies the similar data in the production knowledge base repository.

[0022] Accordingly, one advantage of the present invention is that it merges duplicate data while preserving the semantic relationship, data integrity, and data quality.

[0023] Accordingly, one advantage of the present invention is that it speeds up the cleansing of the data in the production knowledge base repository to ensure that it is relevant, up to date, qualified and useful to the end-users.

[0024] Accordingly, one advantage of the present invention is that it speeds up the process of eliminating similar data in the production knowledge base repository while a system is running.

[0025] Accordingly, one advantage of the present invention is that it utilizes a user-friendly interface to implement the present system and method.

[0026] Other features of embodiments of the present invention will be apparent from accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

5 [0027] In the figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label with a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description applies to any one of the similar components having the same first reference label irrespective of the second
10 reference label.

[0028] FIG. 1 illustrates a flowchart of a method for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository, in accordance with an embodiment of the present invention.

15 [0029] FIG. 2 illustrates a block diagram of a duplicate entity cleaning system, in accordance with an embodiment of the present invention.

[0030] FIG. 3 illustrates an operational flowchart of the object harmonisation module, in accordance with an embodiment of the present invention.

[0031] FIG. 4 illustrates an operational flowchart of the homogeneity recognition module, in accordance with an embodiment of the present invention.

20 [0032] FIG. 5 illustrates an operational flowchart of the entity conflation and merging module, in accordance with an embodiment of the present invention.

[0033] FIGs. 6A-6I illustrate examples of processing duplicates through the embodiments of the present invention.

DETAILED DESCRIPTION

[0034] Systems and methods are disclosed for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository. Embodiments of the present invention include various steps, which will be described below. The steps may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor programmed with the instructions to perform the steps. Alternatively, steps may be performed by a combination of hardware, software, firmware, and/or by human operators.

[0035] Embodiments of the present invention may be provided as a computer program product, which may include a machine-readable storage medium tangibly embodying thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process. The machine-readable medium may include, but is not limited to, fixed (hard) drives, magnetic tape, floppy diskettes, optical disks, compact disc read-only memories (CD-ROMs), and magneto-optical disks, semiconductor memories, such as ROMs, PROMs, random access memories (RAMs), programmable read-only memories (PROMs), erasable PROMs (EPROMs), electrically erasable PROMs (EEPROMs), flash memory, magnetic or optical cards, or other type of media/machine-readable medium suitable for storing electronic instructions (e.g., computer programming code, such as software or firmware).

[0036] Various methods described herein may be practised by combining one or more machine-readable storage media containing the code according to the present invention with appropriate standard computer hardware to execute the code contained therein. An apparatus for practicing various embodiments of the present invention may involve one or more computers (or one or more processors within a single computer) and storage systems containing or having network access to computer program(s) coded in accordance with various methods described herein, and the method steps of the invention could be accomplished by modules, routines, subroutines, or subparts of a computer program product.

[0037] The present invention discloses a system and method whereby the similar data in a production knowledge base repository is automatically identified. The system and method

include an object harmonisation module, a homogeneity recognition module, and an entity conflation and merging module. The function of the object harmonisation module is to perform the first level of identification of duplicates involves data sanitization, standardization, and identification of duplicate objects per concept-attribute of a specific entity. It harmonizes data to ensure the uniqueness of objects in every entity property. Further, the function of the homogeneity recognition module is to perform a second level of duplicate identification. It analyses and groups homogenous entity based on attributes and its objects according to the defined similarity threshold and ranking the similarity of entities. The function of the entity conflation and merging module is to enable the data verification and merging process on similar entity groups by subject matter experts to ensure data correctness and the preservation of data semantic relationship.

[0038] Although the present invention has been described with the purpose of managing duplicate entities based on relationship cardinality in the production knowledge base repository, it should be appreciated that the same has been done merely to illustrate the invention in an exemplary manner and any other purpose or function for which explained structures or configurations could be used, is covered within the scope of the present invention.

[0039] Exemplary embodiments will now be described more fully hereinafter with reference to the accompanying drawings, in which exemplary embodiments are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. These embodiments are provided so that this invention will be thorough and complete and will fully convey the scope of the invention to those of ordinary skill in the art. Moreover, all statements herein reciting embodiments of the invention, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future (i.e., any elements developed that perform the same function, regardless of structure).

[0040] Thus, for example, it will be appreciated by those of ordinary skill in the art that the diagrams, schematics, illustrations, and the like represent conceptual views or processes illustrating systems and methods embodying this invention. The functions of the various elements shown in the figures may be provided through the use of dedicated hardware as

well as hardware capable of executing associated software. Similarly, any switches shown in the figures are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the entity
5 implementing this invention. Those of ordinary skill in the art further understand that the exemplary hardware, software, processes, methods, and/or operating systems described herein are for illustrative purposes and, thus, are not intended to be limited to any particular name.

[0041] Specific details are given in the following description to provide a thorough
10 understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practised without these specific details. For example, circuits, systems, networks, processes, and other components may be shown as components in block diagram form in order not to obscure the embodiments in unnecessary detail. In other instances, well-known circuits, processes, algorithms, structures, and
15 techniques may be shown without unnecessary detail to avoid obscuring the embodiments.

[0042] The term “machine-readable storage medium” or “computer-readable storage medium” includes, but is not limited to, portable or non-portable storage devices, optical storage devices, and various other mediums capable of storing, containing, or carrying instruction(s) and/or data. A machine-readable medium may include a non-transitory
20 medium in which data can be stored, and that does not include carrier waves and/or transitory electronic signals propagating wirelessly or over wired connections. Examples of a non-transitory medium may include but are not limited to, a magnetic disk or tape, optical storage media such as compact disk (CD) or versatile digital disk (DVD), flash memory, memory or memory devices.

[0043] FIG. 1 illustrates a flowchart **100** of a method for managing one or more duplicate
25 entities based on a relationship cardinality in a production knowledge base repository, in accordance with an embodiment of the present invention. The method comprises a step **102** of performing a first level detection of duplicates in existing data present in a production knowledge base repository through an object harmonisation module. The first
30 level detection identifies duplicates of one or more attribute objects within a specific entity. The object harmonisation module implements a sanitization and standardization operation on the identified attribute objects.

[0044] Then a step **104** of performing a second level detection of duplicates between entities of a specific concept through a homogeneity recognition module. The homogeneity recognition module identifies duplicates according to base-attributes of the specific concept based on a predefined similarity threshold. The predefined similarity threshold can be defined by user. In one embodiment, the predefined similarity threshold can be set as 0.8.

[0045] It follows with a step **106** of enabling a user to determine the similarity of the entities and further enables the user to merge the similar entities through an entity conflation and merging module. The entity conflation and merging module enables the user to nominate a primary entity within a similar entity. The entity conflation and merging module implements merging of a plurality of secondary entities into the primary entity according to the relationship cardinality. The data in the secondary entities do not overwrite the data in the primary entity.

[0046] FIG. 2 illustrates a block diagram of a duplicate entity cleaning system **200** in accordance with an embodiment of the present invention. The duplicate data cleaning system **200** includes routines, programs, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types to process data stored in a production knowledge base repository **208**. The duplicate data cleaning system **200** comprises an object harmonisation module **202**, a homogeneity recognition module **204** and an entity conflation and merging module **206** adapted to process data in the production knowledge base repository **208**. The object harmonisation module **202** is adapted to perform a first level detection of duplicates based on a set of object literal rules **222**. Object literal rules are a set of rules adapted for standardising the format of object of object labels and literals, which may vary from one knowledge base to another depending on its configurations. Giving a non-limiting example, the concept: Person with concept-attribute: rdfs:label, the object literals (1) first letter of a word should be capitalised (2) should not contain @!\$%&*()+=<>?/ characters . In another example, the concept: Person with concept-attribute: sample:hasEmail, the object literals (1) must contain @ character at most once (2) must not have space and (),,;,<>[\] characters. It is understood to a skilled person that entities of different concept-attributes may be formatted in a predefined format known in the art. The homogeneity recognition module **204** adapts homogeneity recognition algorithm **244** to carry out second level detection of duplicates based on concept base-attributes definition **242**. Homogeneity recognition is a process using combination of string

metrics to identify similarities. Concept Base-Attributes definition are defined by application domain. For example: Concepts: C1, C2...Cn. Base-Attributes: B1, B2...Bn. If concept C1 = Person, base-attributes can be B1 = rdfs:label, B2 = sample:hasEmail, B3 = sample:fromOrganisation .

5 **[0047]** The detected duplicates are grouped and stored in a similar entity groups **262**. The grouped entities can be used for validating the duplicate entities per concept. The entity conflation and merging module **206** retrieves a relationship (property) cardinality reference **264** to aid the duplicate entities conflation and merging according to the user preferences.

10 **[0048]** After the entities of the knowledge base repository **208** are processed by the duplicate entity cleaning system **200**, a clean production knowledge base repository (210) is output and stored with the duplicate entities cleaned up.

15 **[0049]** FIG. 3 illustrates an operational flowchart **300** of the object harmonisation module **202**, in accordance with an embodiment of the present invention. The object harmonisation module **202** retrieves a list of the concepts at step **302** and a list of the attributes of a concept at step **304**. Following that, the object harmonisation module **202** extracts all entity triples for each attribute at step **306**. Samples of the triples are provided in the FIG. 6A.

20 **[0050]** At step **318**, the object harmonisation module **202** processes an entity at a time. At step **320**, the entity is checked if it has more than one object for the attributes. If there isn't, the object harmonisation module **202** checks if there are more entity to check, if there are the step of **318** and **320** are repeated. If there is no more entity to check, the object harmonisation module **202** checks if there are more attributes and/or more concepts to process. Similarly, when there are more attributes and/or concepts to be processed, steps **302**, **304**, **306**, **318** and **320** are repeated accordingly. FIG. 6B provides an example of literal objects being processed by the object harmonisation module **202**.

25 **[0051]** The triple objects comprise "Leigh spencer", "Leigh S." and "leigh_spencer", whereby the sanitization process removes unwanted symbol mentioned above. Following that, the objects standardize the font case into title case (i.e., capitalized the first character of each word). Thereafter, the sanitized triples are deleted.

30 **[0052]** Returning to the step **320**, when the entity have found to have more than one object for the attribute, the object harmonisation module **202** identifies if the object is a resource

description framework (RDF) uniform resource identifier (URI) at step **312**. At step **315**, if the object is an RDF URI, its URI label is retrieved from the production knowledge-based respiratory **208**. The object harmonisation module **202** proceeds to sanitise labels and literals based on object literal rules **222** at step **308**. If the object is not an RDF URI, the object harmonisation module **202** proceeds immediately to the step **308** to sanitise the labels and the literals. The sanitised labels and literals will be normalised in a standardise format at step **310**. Based on the standard format entities, the object harmonisation module **202** determines, in this first level of detection, if the values of the object of triples for a specific entity and attribute are duplicates at step **316**. If there isn't any duplicate, the object harmonisation module **202** loops back to the earlier process to harmonize the other objects. If a duplicate is found, the object harmonisation module **202** retains only the triple of the object and delete the triple of the other objects at step **314**. In one embodiment, selection of triple is based on the completeness and uniqueness of information. In another embodiment, a first among the triples with duplicate object values for a specific entity and attribute will be retained while the others can be deleted.

[0053] FIG. 6C shows an example of entity harmonisation process for an RDF URI object.. As shown, the objects are identified as RDF URI. As explained above, the object harmonisation module **202** retrieves the URI label. After the process, the triples are deleted.

[0054] FIG. 4 illustrates an operational flowchart **400** of the homogeneity recognition module **204**, in accordance with an embodiment of the present invention. The homogeneity recognition module **204** carries out a second level detection of one or more duplicates between entities of a specific concept. The homogeneity recognition module **204** identifies duplicates according to one or more base-attributes of the specific concept based on a predefined similarity threshold. The second level detection starts with retrieving a list of all main concepts and its designed based-attribute from concept base attributes definition **401** at step **402**. In one embodiment, the concept based-attributes definition can be defined by an application domain specifics to its knowledge base design known in the art. At step **404**, the homogeneity recognition module **204** lists all entity having the base-attributes with its corresponding object from the production knowledge base respiratory **403**. In one embodiment, the production knowledge base respiratory **403** stores entities processed by the object harmonisation module **202**. At step **406**, the homogeneity recognition module **204** obtains a base-attribute count, X, to be processed. At step **408**, if there is only one

entity in the same group, the homogeneity recognition module **204** executes its homogeneity recognition with base-attribute object based on defined similarity threshold at step **409**, wherein the homogeneity recognition is based on a homogeneity recognition algorithm **411**. In one embodiment, the homogeneity recognition module may adapt any homogeneity recognition or detection algorithms, proprietary or opened source, that are well known in art. Similarly, the similarity threshold for purpose of homogeneity is can be set according to the adapted algorithm. Then, the homogeneity recognition module **204** checks if there are any similar entities at step **413**. If similar entities found, at step **414**, it stores as a list of entity group in a similar entity group (data) **415**. If there are no similar entities found at step **413**, the module **204** further checks if there are more concepts at step **416**. When more concept is found, the process loops to step **404**.

[0055] Referring back to the step **408**, when there is more than one entity of the same group, the homogeneity recognition module **204** execute its homogeneity recognition with base-attribute object based on a defined similarity threshold **410**. The homogeneity recognition is based on homogeneity recognition algorithm **411**. Similarly, the homogeneity recognition module **204** determines if similar entities are found at step **412**. If no similar entities are found, it stores **414** as a list of entity group in the similar entity group **415**. If similar entities found at step **412**, it computes a similarity ranking data at step **422**, then checks for more base-attribute at step **420**. If there isn't, similarity ranking is computed at step **418**, and stores a list of similar entity group at step **414**. If there are more base-attribute, it loops to the step **410** to perform homogeneity recognition.

[0056] Over the above process, the homogeneity recognition module **204** repeats the execution of homogeneity recognition for each base-attribute having a first base-attribute as the primary base-attribute in identifying the concepts that have more than one base-attribute. The homogeneity recognition module **204** further computes **422** the similarity ranking as per similar entity group **412**, and stores **414** the identified list of similar entity group for later use.

[0057] In an embodiment, the similarity ranking **418** is obtained by receiving a base-attribute count (X) of the specific concept, identifying the primary base-attribute for the specific concept, computing the primary base-attribute value per entity by multiplying 0.65 to X, computing the base-attribute value per entity by dividing 0.35 by the difference of X-1, and multiplying the result by X. The parameters are use to give more weight to the

primary base-attribute then other base-attributes. In other embodiments, other ratios or the parameters can be used. In an embodiment, the similarity ranking is based on the summation of primary base-attribute value and the base-attribute values per entity.

[0058] Returning to the step **414**, besides storing the outcome to the similar entity group
5 **415**, the process loops to the step **416** to determine if more concept to be processed.

[0059] FIG. 6D and FIG. 6E exemplifies a homogeneity recognition of the entities. In the first round of process, entities of the primary base-attribute $B1 = \text{rdfs:label}$ are being processed. Among the entities, similar entities of the primary base-attribute are selected for the next succeeding process. In the illustrated example, PersonEntity2, PersonEntity3
10 and PersonEntity5 are selected. In the second round of process, the base-attribute $B2 = \text{sample:hasEmail}$ of each of the selected (similar) entities are process to determine for similarities. In the second round, PersonEntity2 and PersonEntity3 have an identical email address. In its third round, the base-attribute $B3 = \text{sample:fromOrganisation}$, the same same set of selected (similar) entities are processed. PersonEntity2 doesn't contain any
15 entry for B3, whereby the PersonEntity3 and PersonEntity5 contain similar entry.

[0060] Subsequently, a similarity ranking will be computed as shown in FIG. 6E. Foremost, since three entities are selected, $X=3$. Accordingly, a primary base-attribute value is $X*0.65 = 1.95$. For B2, the base-attribute value $= X[0.35/(X-1)]$ and a value 0.525 is obtained. The same goes to B3. But B2 is absent in PersonEntity5 and B3 is absent in
20 PersonEntity2, therefore, no value (or zero) will be assigned. Following that, the base-attribute value for each entity will be summed up to determine the ranking for each entity, and the selected entities will be sorted according to similarity ranking forming a similar entity group.

[0061] FIG. 5 illustrates an operational flow **500** carried out by the entity conflation and
25 merging module **206**, in accordance with an embodiment of the present invention. The entity conflation and merging module **206** allows the user to perform the final duplicates cleaning according to the user preferences. At step **502**, the entity conflation and merging module **206** prompts the user to select a concept, and based on the input, the entity conflation and merging module **206** displays a list of similar entity group under the selected
30 concept at step **504**. Following that, at step **506**, all the value of object of an entity belonging to a group is shown. At step **508**, the user is prompt to select the primary entity

in the group and subsequently, at step **510**, to select secondary entities in the group. With the selections, the entity conflation and merging module **206** implements the merging of a plurality of secondary entities into the primary entity according to the relationship cardinality, if it is required at step **512**. If the entities are to be merged, the entity conflation and merging module **206** keep properties of at least the (selected) primary entity in the group and store it in a temporary properties list **515** of the primary entity at step **514**. At step **516**, it processes properties of the secondary entities. At step **518**, the entity conflation and merging module refers to a relationship (property) cardinality reference **520** to check property cardinality of the second entities' properties. If the property cardinality $1=1$, it further checks if the property is existing in the primary entity at step **522**. If so, the secondary property is removed at step **523**. At step **524**, the entity conflation and merging module **206** determines if more secondary properties to be processed. If so, the steps **516**, **518**, **522**, **523** to **524** are repeated until no other secondary property is available.

[0062] In step **512**, the selections is performed by user to decide whether a pair of records shall be consider a duplicate, and is required to be merged. Records that contain criteria such as maximum or most information available will be considered as a primary record in the group. One or more secondary records can be merged with the primary record according to the relationship's cardinality. In one embodiment, secondary records will not be overriding the primary record, rather, any additional data in the secondary record that is not present in the primary record will be appended to the primary record for ensuring preservation of its semantic relationship.

[0063] Returning to the steps **521** and **522**, when either the property existing in the primary cardinality $1=1$ is not true or no property existing in the primary entity, the entity conflation and merging module **206** replaces secondary entity URI of the corresponding triple with the primary entity URI at step **525**. At step **526**, it adds the property to the temporary properties list of the primary entity **515**. Thereafter, the process loops back to the step **524** to check if more secondary properties.

[0064] Returning to the steps **512**, and the step **524**, when the entities are not to be merged (or no entities to be merged) or no other second properties to be processed, the entity conflation and merging module **206** check if there are more similar entity group. In one embodiment, user manually checks if there are more similar entity group through the conflation and merging module **206**. If there is, the process loops back to process **506** to

carry out the aforesaid process. When there is no other similar entity group found, the selections are updated to the production knowledge base **210** at step **528**. Subsequently, at step **529**, if there are more similar entity group, the process **500** is repeated. If no more similar entity group, the process is concluded.

5 **[0065]** FIG. 6F provides an example of processing the selected primary entity and the selected secondary entity through the entity conflation and merging module. In the illustrated example, the PersonEntity3 is assigned as the primary record, whilst the PersonEntity2 and PersonEntity5 are assigned as the secondary records.

10 **[0066]** FIG. 6G illustrates an example of merging secondary entity <PersonEntity2-URI> to the primary entity through the entity conflation and merging module. Based on the relationship cardinality of each record of PersonEntity2, relationship cardinality that are one-to-one and properties exist in the primary entity are removed, whilst the relationship cardinality is one-to-many, it replaces secondary entity URI with the primary entity in the triple. Thereafter, the knowledge base will be updated with the triple.

15 **[0067]** FIG. 6H illustrates an example of merging secondary entity <PersonEntity5-URI> to the primary entity through the entity conflation and merging module. Similarly, relationship cardinality that is one-to-many in the PersonEntity5, the secondary entity URI will be replaced with the primary entity URI in the triple and the knowledge base will be updated with the triple.

20 **[0068]** FIG. 6I illustrates an example of merged triples for the primary entity in the production knowledge base.

25 **[0069]** Thus the present system and method automatically identifies the similar data in the production knowledge base repository and merges the duplicate data while preserving the semantic relationship, data integrity, and data quality. Further, the present invention speeds up the cleansing of the data in the production knowledge base repository to ensure that it is relevant, up to date, qualified and useful to the end-users. Furthermore, the present invention speeds up the process of eliminating similar data in the production knowledge base repository while a system is running and utilizes a user-friendly interface to implement the present system and method.

[0070] While embodiments of the present invention have been illustrated and described, it will be clear that the invention is not limited to these embodiments only. Numerous modifications, changes, variations, substitutions, and equivalents will be apparent to those skilled in the art, without departing from the scope of the invention, as described in the claims.

CLAIMS:

1. A method for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository, the method comprises steps of:

5 performing, by one or more processors, a first level detection of one or more duplicates in existing data present in the production knowledge base repository through an object harmonisation module (202), wherein the first level detection identifies duplicates of one or more attribute objects within a specific entity, wherein the object harmonisation module (202) implements a sanitization operation and a standardization operation on the identified
10 attribute objects;

performing, by one or more processors, a second level detection of one or more duplicates between a plurality of entities of a specific concept through a homogeneity recognition module (204), wherein the homogeneity recognition module (204) identifies duplicates according to one or more base-attributes of the specific concept based on a predefined
15 similarity threshold; and

enabling, by one or more processors, a user to determine the similarity of the entities and further enables the user to merge the similar entities through an entity conflation and merging module (206), wherein the entity conflation and merging module (206) enables the user to nominate a primary entity within a similar entity, wherein the entity conflation
20 and merging module (206) implements the merging of a plurality of secondary entities into the primary entity according to the relationship cardinality, wherein the data in the secondary entities do not overwrite the data in the primary entity.

2. The method according to claim 1, wherein the object harmonisation module (202) further
25 performs steps of:

extracting and grouping a plurality of entity triples per concept-attribute;
quantifying the attribute objects for the specific entity;
identifying a predefined classification of attribute object, wherein the classification comprises a resource description framework, RDF, a uniform resource identifier, URI, and
30 a plurality of literals;
sanitizing and standardizing a format of a plurality of object values;
comparing the object values and identifying exact duplicate of the object values;

harmonizing object data by deleting one or more duplicate triples of duplicate object URI/literals for an entity to ensure the uniqueness of the attribute objects; and updating the production knowledge base repository.

- 5 3. The method according to claim 2, wherein the sanitizing and standardizing format of object values comprises steps of:
removing of unnecessary characters from the object labels and literals; and
modifying into the standard format of object labels and object literals based on a predefined object literal rules.
- 10
4. The method according to claim 1, wherein the homogeneity recognition module (204) comprises a concept base-attribute definition for analysis of a reference scope, wherein the homogeneity recognition module operably comprises the steps of:
identifying if the concept has more than one base-attribute;
15 executing homogeneity recognition based on a predefined similarity threshold;
repeating the execution of homogeneity recognition for each base-attribute having a first base-attribute as the primary base-attribute on identifying the concepts have more than one base-attribute;
computing a similarity ranking as per similar entity group; and
20 storing the identified list of similar entity group for usage of the entity conflation and merging module (206).
5. The method according to claim 4, wherein the similarity ranking is obtained by performing a plurality of steps comprises:
25 receiving a base-attribute count, X of the specific concept;
identifying the primary base-attribute for the specific concept;
computing the primary base-attribute value per entity by multiplying 0.65 to X;
computing the base-attribute value per entity by dividing 0.35 by the difference of X-1; and
multiplying the result by X, wherein the similarity ranking is based on the summation of
30 primary base-attribute value and the base-attribute values per entity.
6. The method according to claim 1, wherein the entity conflation and merging module (206) comprises a user interface for verification, wherein the entity conflation and merging module presents a list of similar entity groups in a concept sorted by the similarity ranking;

displays a plurality of relationships with the corresponding objects for each entity; enables the user to verify data; nominates the primary entity and the secondary entity; validates property cardinality as defined in the relationship cardinality reference; merges duplicate entities to ensure the preservation of the semantic relationship, wherein the merging of duplicate entities comprise steps of replacing URI of secondary entities triple with the primary entity URI based in the property cardinality; appending the data in the primary entity not contained in the secondary entities to ensure the preservation of the semantic relationship of the primary entity; removing the redundant data triples; and updates the production knowledge base repository with cleansed data.

7. A system for managing one or more duplicate entities based on a relationship cardinality in a production knowledge base repository, the system comprises:
- a memory (206) to store machine-readable instructions pertaining to manage the duplicate entities; and
 - a processor (202) coupled to the memory and operable to execute the machine-readable instructions stored in the memory, wherein the processor (202) comprises:
 - an object harmonisation module (202) to perform a first level detection of one or more duplicates in existing data present in the production knowledge base repository, wherein the first level detection identifies duplicates of one or more attribute objects within a specific entity, and wherein the object harmonisation module (202) implements a sanitization operation and a standardization operation on the identified attribute objects;
 - a homogeneity recognition module (204) to perform a second level detection of one or more duplicates between a plurality of entities of a specific concept, wherein the homogeneity recognition module (204) identifies duplicates according to one or more base-attributes of the specific concept based on a predefined similarity threshold; and
 - an entity conflation and merging module (206) enables a user to determine the similarity of the entities and further enables the user to merge the similar entities, wherein the entity conflation and merging module (206) enables the user to nominate a primary entity within a similar entity, wherein the entity conflation and merging module (206) implements merging of a plurality of secondary entities into the primary entity according to the relationship cardinality, and wherein the data in the secondary entities do not overwrite the data in the primary entity.

8. The system according to claim 1, wherein the object harmonisation module (202) is configured for extracting and grouping a plurality of entity triples per concept-attribute; quantifying the attribute objects for the specific entity; identifying a predefined classification of attribute object, wherein the classification comprises a resource description framework, RDF, a uniform resource identifier, URI, and a plurality of literals; sanitising and standardising format of a plurality of object values; comparing the object values and identifying exact duplicate of the object values; harmonizing object data by deleting one or more duplicate triples of duplicate object URI/literals for an entity to ensure the uniqueness of the attribute objects; and updating the production knowledge base repository.
9. The system according to claim 1, wherein the homogeneity recognition module (204) comprises a concept base-attribute definition for analysis of a reference scope, wherein the homogeneity recognition module (204) is configured for identifying if concept has more than one base-attribute; executing homogeneity recognition based on a predefined similarity threshold; repeating the execution of homogeneity recognition for each base-attribute having a first base-attribute as the primary base-attribute on identifying the concepts have more than one base-attribute; computing similarity ranking as per similar entity group; and storing the identified list of similar entity group for usage of the entity conflation and merging module (206).
10. The system according to claim 1, wherein the entity conflation and merging module (206) comprises a user interface for verification, wherein the entity conflation and merging module (206) presents a list of similar entity groups in a concept sorted by the similarity ranking; displays a plurality of relationships with the corresponding objects for each entity; enables the user to verify data; nominates the primary entity and the secondary entity; validates property cardinality as defined in the relationship cardinality reference; merges duplicate entities to ensure the preservation of the semantic relationship; and updates the production knowledge base repository with cleansed data.

1/10

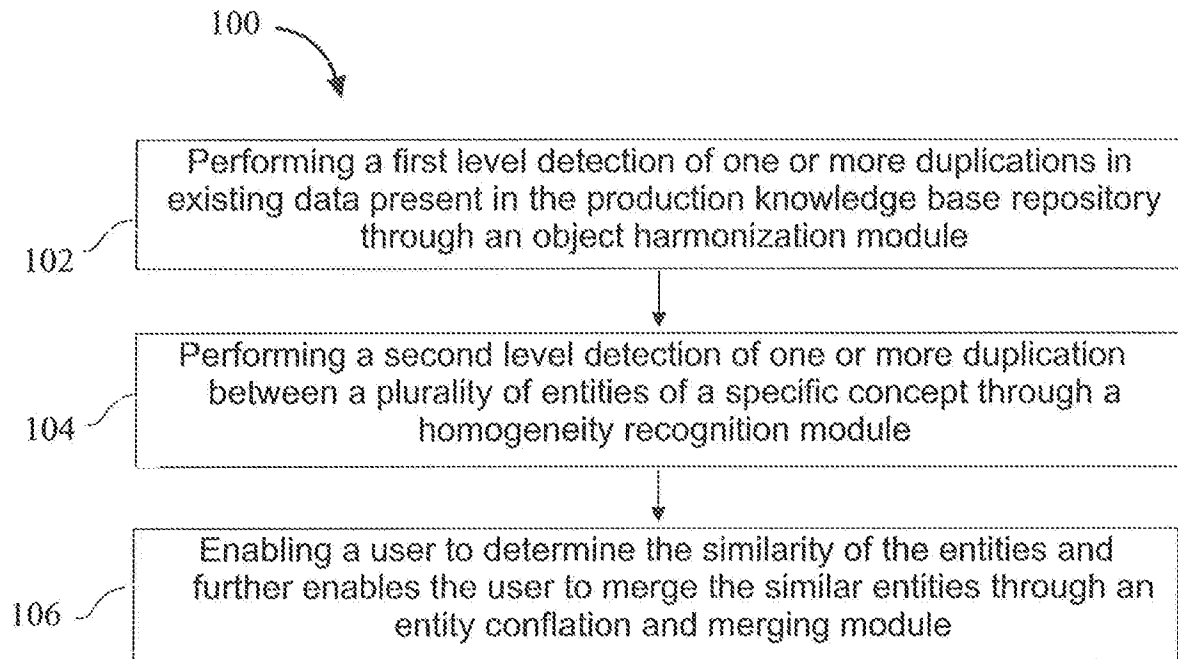


FIG. 1

2/10

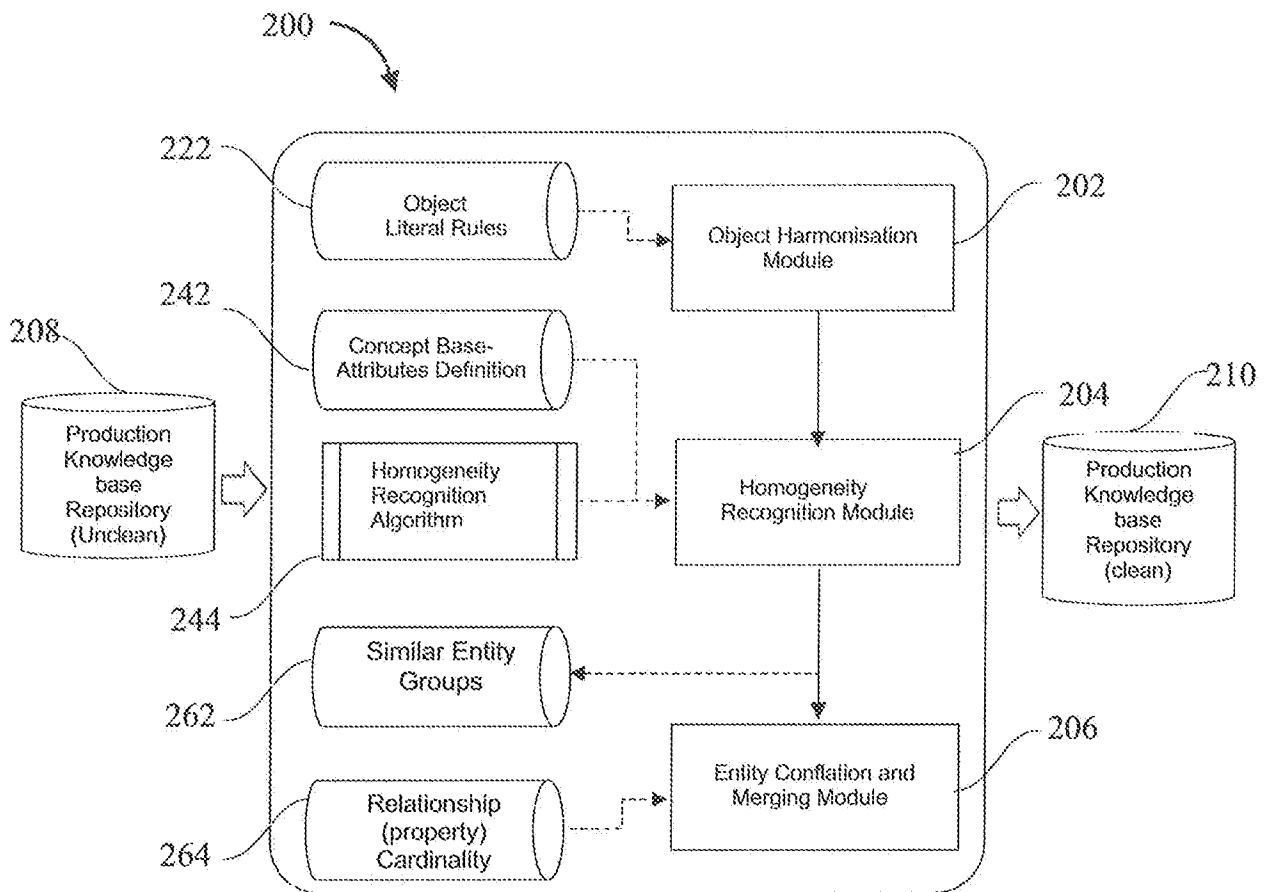


FIG. 2

3/10

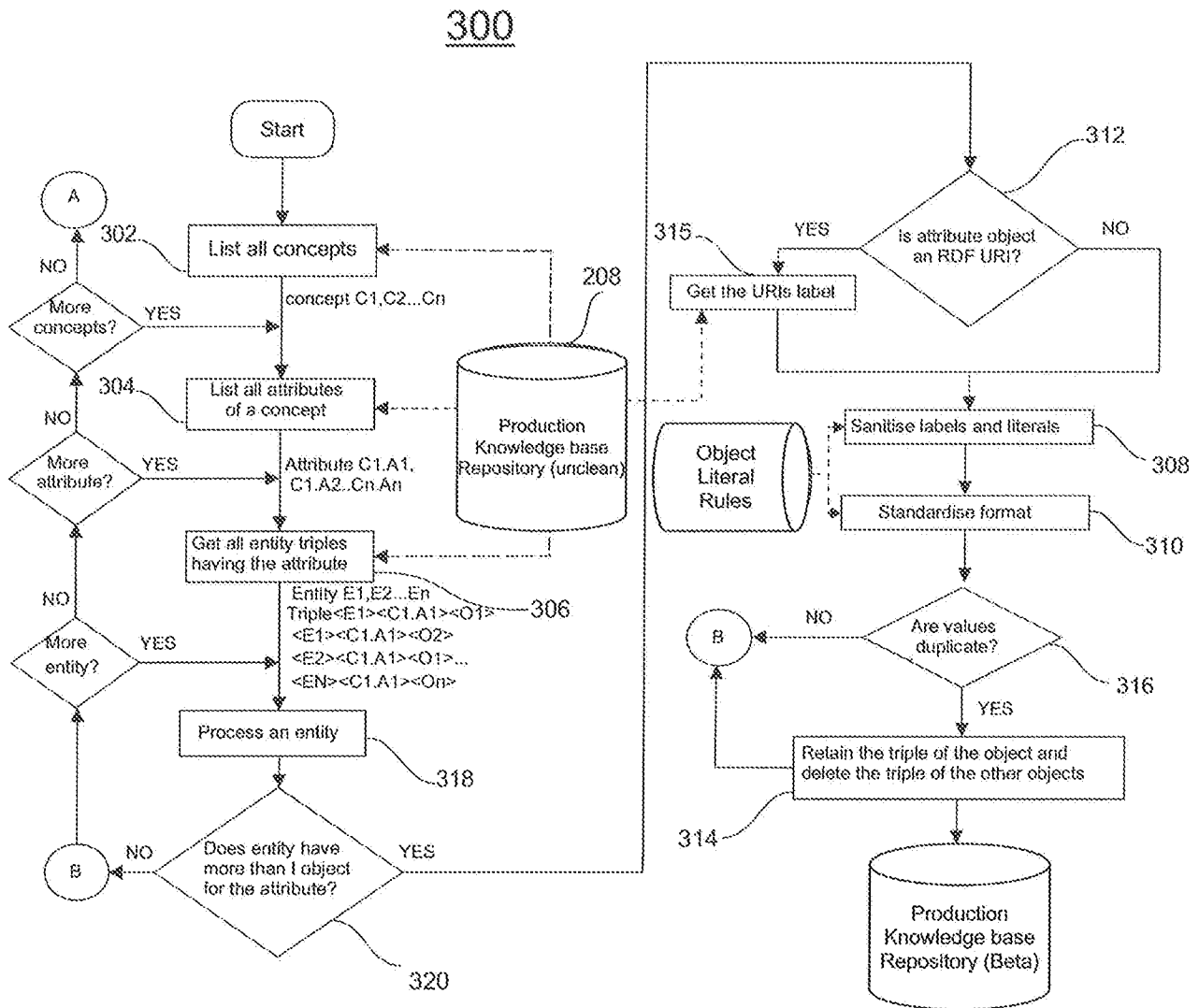
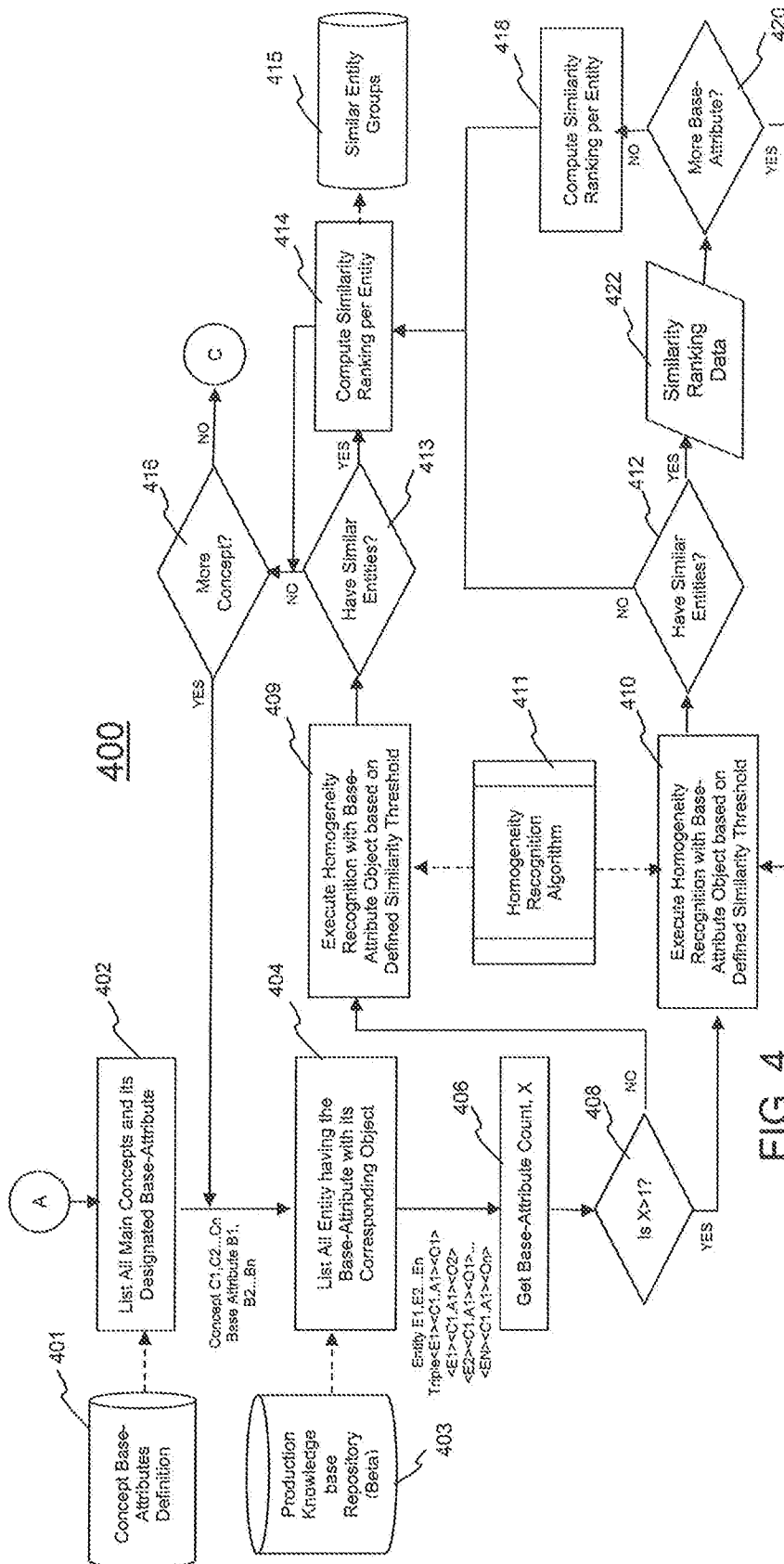
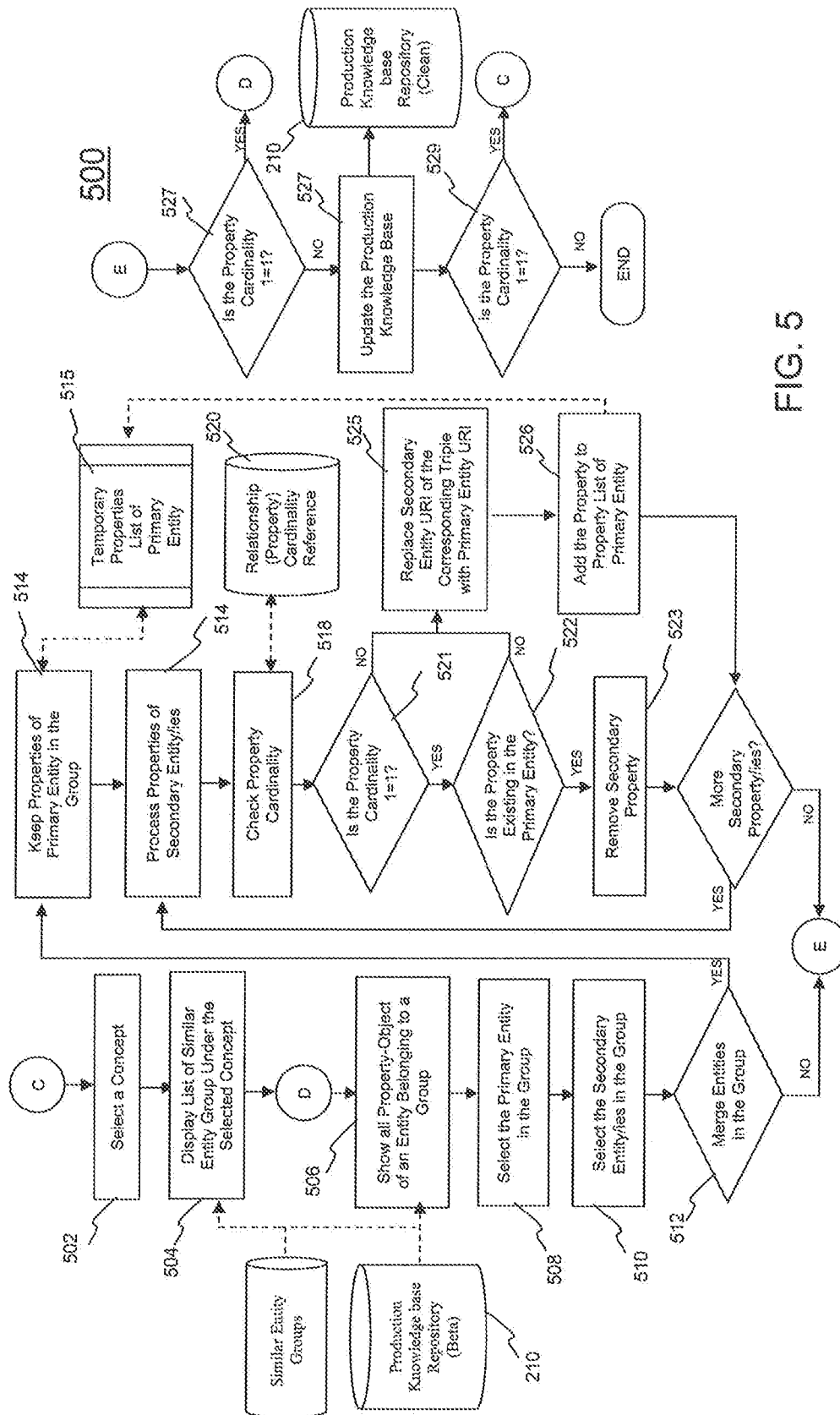


FIG. 3

4/10



5/10



6/10

Entity	Concept-Attribute	Object	
<PersonEntity1-URI>	<rdfs:label>	"Leslie Spencer"	Entity 1
<PersonEntity1-URI>	<sample:hasEmail>	"les_spencer@gmail.com"	
<PersonEntity2-URI>	<rdfs:label>	"Sophie Parker"	Entity 2
<PersonEntity2-URI>	<sample:hasEmail>	"sophie@email.com"	
<PersonEntity3-URI>	<rdfs:label>	"Sophie L Parker"	Entity 3
<PersonEntity3-URI>	<sample:hasEmail>	"sophie@email.com"	
<PersonEntity3-URI>	<sample:fromOrganisation>	<OrgEntity1-URI>	
<PersonEntity4-URI>	<rdfs:label>	"Aubrey Collins"	Entity 4
<PersonEntity4-URI>	<sample:fromOrganisation>	<OrgEntity2-URI>	
<PersonEntity5-URI>	<rdfs:label>	"Sophie P."	Entity 5
<PersonEntity5-URI>	<sample:hasEmail>	"sp@gmail.com"	
<PersonEntity5-URI>	<sample:fromOrganisation>	<OrgEntity1-URI>	

Base-Attributes for Concept Person:

B1 = rdfs:label -> Primary Base-Attribute

B2 = sample:hasEmail

B3 = sample:fromOrganisation

FIG. 6A

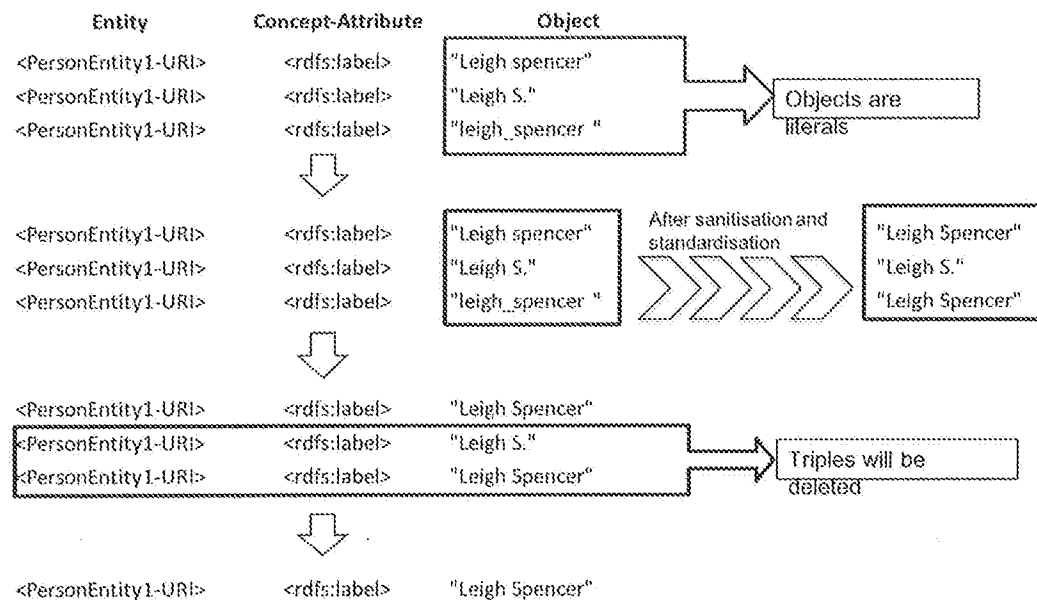


FIG. 6B

7/10

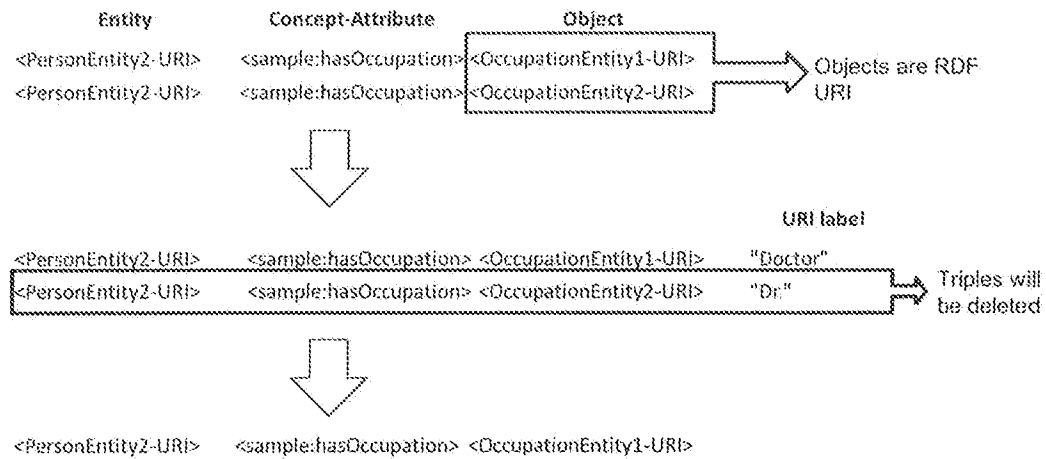


FIG. 6C

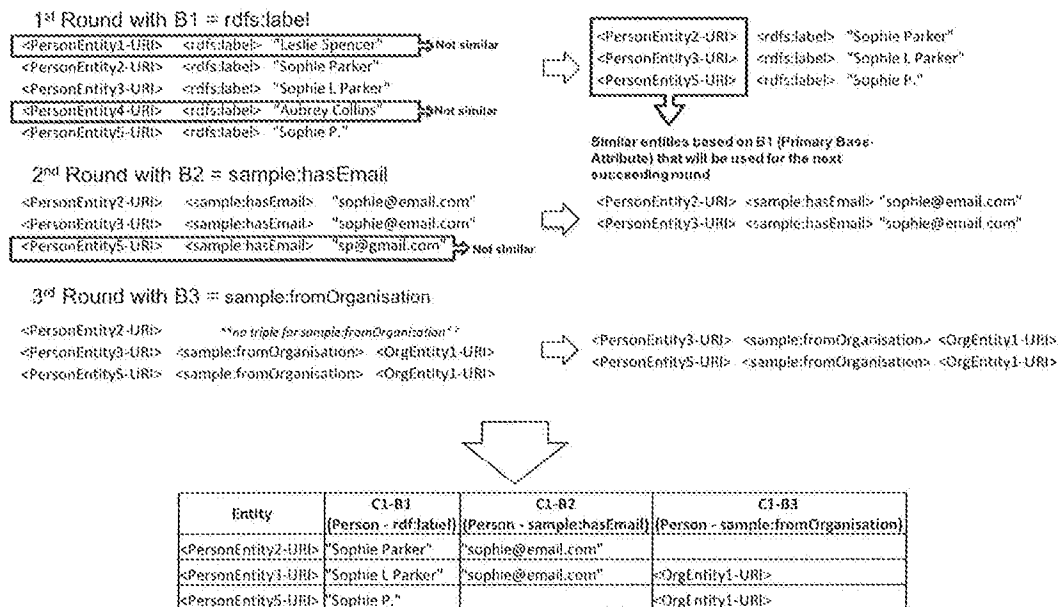
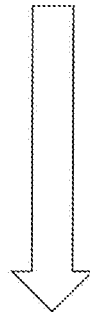


FIG. 6D

8/10

Entity	C1-B1 (Person - rdf:label)	C1-B2 (Person - sample:hasEmail)	C1-B3 (Person - sample:fromOrganisation)	Similarity Ranking
<PersonEntity2-URI>	"Sophie Parker"	"sophie@email.com"		2 nd
<PersonEntity3-URI>	"Sophie L Parker"	"sophie@email.com"	<OrgEntity1-URI>	1 st
<PersonEntity5-URI>	"Sophie P."		<OrgEntity1-URI>	3 rd



Computation of Similarity Ranking

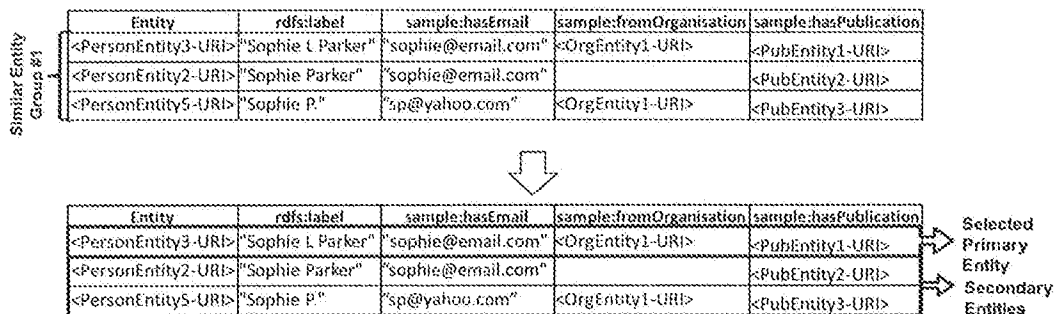
X = base-attribute count = 3
 $C1-B1, C2-B1, \dots, Cn-B1$ = primary base-attribute value = $X * 0.65$ = 1.95
 $C1-B2, \dots, C1-Bn, C2-B2, \dots, C2-Bn, \dots, Cn-B2, \dots, Cn-Bn$ = base-attribute value = $X [(0.35)/(X-1)]$ = 0.525

Entity	C1-B1 (primary base-attribute value)	C1-B2	C1-B3	Similarity Value	Similarity Ranking
<PersonEntity2-URI>	1.95	0.525		2.475	2 nd
<PersonEntity3-URI>	1.95	0.525	0.525	3	1 st
<PersonEntity5-URI>	1.95		0.525	2.475	3 rd

Similar Entity Group

Entity	C1-B1 (Person - rdf:label)	C1-B2 (Person - sample:hasEmail)	C1-B3 (Person - sample:fromOrganisation)	Similarity Ranking
<PersonEntity3-URI>	"Sophie L Parker"	"sophie@email.com"	<OrgEntity1-URI>	1 st
<PersonEntity2-URI>	"Sophie Parker"	"sophie@email.com"		2 nd
<PersonEntity5-URI>	"Sophie P."		<OrgEntity1-URI>	3 rd

FIG. 6E



Triple for Primary entity will be retain in Production Knowledge Base

<PersonEntity3-URI> <rdfs:label> "Sophie L Parker"
 <PersonEntity3-URI> <sample:hasEmail> "sophie@email.com"
 <PersonEntity3-URI> <sample:fromOrganisation> <OrgEntity1-URI>
 <PersonEntity3-URI> <sample:hasPublication> <PubEntity1-URI>

FIG. 6F

9/10

Merge Secondary Entity <PersonEntity2-URI> to Primary Entity

<PersonEntity2-URI> <rdfs:label> "Sophie Parker"
 <PersonEntity2-URI> <sample:hasEmail> "sophie@email.com"
 <PersonEntity2-URI> <sample:hasPublication> <PubEntity2-URI>

Relationship Cardinality	Property Exist in Primary Entity
one-to-one	YES
one-to-one	YES
one-to-many	YES

<PersonEntity2-URI> <rdfs:label> "Sophie Parker"
 <PersonEntity2-URI> <sample:hasEmail> "sophie@email.com"
 <PersonEntity2-URI> <sample:hasPublication> <PubEntity2-URI>

To be removed. Relationship Cardinality are one-to-one and properties exist in Primary Entity

<PersonEntity3-URI> <sample:hasPublication> <PubEntity2-URI>

Relationship Cardinality is one-to-many, Replace secondary entity URI with Primary Entity URI in the triple

<PersonEntity3-URI> <sample:hasPublication> <PubEntity2-URI>

Knowledge Base will be updated with this triple

FIG. 6G

Merge Secondary Entity <PersonEntity5-URI> to Primary Entity

<PersonEntity5-URI> <rdfs:label> "Sophie Parker"
 <PersonEntity5-URI> <sample:hasEmail> "sophie@email.com"
 <PersonEntity5-URI> <sample:fromOrganisation> <OrgEntity1-URI>
 <PersonEntity5-URI> <sample:hasPublication> <PubEntity3-URI>

Relationship Cardinality	Property Exist in Primary Entity
one-to-one	YES
one-to-one	YES
one-to-one	YES
one-to-many	YES

<PersonEntity5-URI> <rdfs:label> "Sophie Parker"
 <PersonEntity5-URI> <sample:hasEmail> "sophie@email.com"
 <PersonEntity5-URI> <sample:fromOrganisation> <OrgEntity1-URI>
 <PersonEntity5-URI> <sample:hasPublication> <PubEntity3-URI>

To be removed. Relationship Cardinality are one-to-one and properties exist in Primary Entity

<PersonEntity5-URI> <sample:hasPublication> <PubEntity3-URI>

Relationship Cardinality is one-to-many, Replace secondary entity URI with Primary Entity URI in the triple

Replace URI with Primary Entity URI

<PersonEntity3-URI> <sample:hasPublication> <PubEntity3-URI>

Knowledge Base will be updated with this triple

FIG. 6H

10/10

**Merged Triples for Primary entity in Production
Knowledge Base**

<PersonEntity3-URI>	<rdfs:label>	"Sophie L Parker"
<PersonEntity3-URI>	<sample:hasEmail>	"sophie@email.com"
<PersonEntity3-URI>	<sample:fromOrganisation>	<OrgEntity1-URI>
<PersonEntity3-URI>	<sample:hasPublication>	<PubEntity3-URI>
<PersonEntity3-URI>	<sample:hasPublication>	<PubEntity2-URI>
<PersonEntity3-URI>	<sample:hasPublication>	<PubEntity3-URI>

FIG. 6I

INTERNATIONAL SEARCH REPORT

International application No.
PCT/MY2019/050093**A. CLASSIFICATION OF SUBJECT MATTER****G06F 16/215(2019.01)i, G06F 16/21(2019.01)i**

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 16/215; G06F 12/00; G06F 17/00; G06F 17/30

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: detecting duplicates, similarity, threshold, merging entity, cardinality, not overwriting

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 9720986 B2 (QATAR FOUNDATION) 01 August 2017 columns 5, 9; and claim 1	1-10
A	US 2015-0081668 A1 (NEC LABORATORIES AMERICA, INC.) 19 March 2015 paragraphs [0051]-[0053]	1-10
A	KR 10-2014-0122400 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 20 October 2014 paragraph [0027]	1-10
A	KR 10-2015-0121505 A (SAMSUNG ELECTRONIC CO., LTD.) 29 October 2015 claims 1-5	1-10
A	AKSHATA DAGADE et al., 'De-duplication Framework to Reduce the Record Linkage Problem', In: 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, IEEE, 23 November 2017 section IV; and figure 1	1-10



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

"D" document cited by the applicant in the international application

"E" earlier application or patent 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

26 March 2020 (26.03.2020)

Date of mailing of the international search report

26 March 2020 (26.03.2020)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050093

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 9720986 B2	01/08/2017	EP 2701081 A1 US 2014-0156606 A1 WO 2014-012576 A1	26/02/2014 05/06/2014 23/01/2014
US 2015-0081668 A1	19/03/2015	EP 3044704 A1 JP 2016-529586 A JP 6123028 B2 US 2016-0147832 A1 US 9569491 B2 WO 2015-038224 A1	20/07/2016 23/09/2016 26/04/2017 26/05/2016 14/02/2017 19/03/2015
KR 10-2014-0122400 A	20/10/2014	US 2014-0310497 A1 US 9189165 B2	16/10/2014 17/11/2015
KR 10-2015-0121505 A	29/10/2015	US 2015-0302022 A1	22/10/2015