



- (51) **International Patent Classification:**
G06F 17/30 (2006.01) *G06F 17/00* (2006.01)
- (21) **International Application Number:**
PCT/US2015/013554
- (22) **International Filing Date:**
29 January 2015 (29.01.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** NOR, Igor; Kiryat Technion, 32000 Haifa (IL). BARKOL, Omer; Kiryat Technion, 32000 Haifa (IL). MAURER, Ron; Kiryat Technion, 32000 Haifa (IL).
- (74) **Agents:** KIRCHEV, Ivan, T et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).
- (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

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

- (54) **Title:** PROCESSING AN ELECTRONIC DATA STREAM USING A GRAPH DATA STRUCTURE

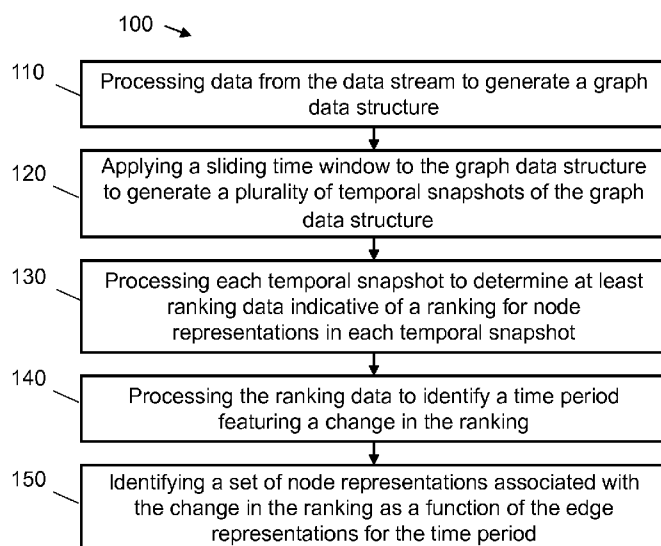


Fig. 1

(57) **Abstract:** Certain examples described herein enable the processing of an electronic data stream using a graph data structure. In these examples a sliding window is applied to the graph data structure to generate at least one temporal snapshot. Each of these temporal snapshots is processed to determine ranking data indicative of a ranking for node representations within a given snapshot. Ranking data for a set of temporal snapshots is then processed to identify a time period featuring a change in the ranking data. This time period is then used to identify node representations within the graph data structure that are associated with the change in the ranking data.

PROCESSING AN ELECTRONIC DATA STREAM USING A GRAPH DATA STRUCTURE

BACKGROUND

[0001] Modern computer systems generate large amounts of data. For example, networked computer systems implementing websites may allow for the posting of information over time. As time passes a large body of information may be collected. These websites may comprise social networks, collaborative websites such as wikis and other content management systems and content providers that allow user comments. Other computer systems that generate large amounts of data include system logs, transaction systems and electronic messaging systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Various features and advantages of the present disclosure will be apparent from the detailed description which follows, taken in conjunction with the accompanying drawings, which together illustrate, by way of example only, features of the present disclosure, and wherein:

[0003] Figure 1 is a flow diagram showing a method for processing an electronic data stream according to an example;

[0004] Figure 2 is a schematic illustration showing data representative of a data update according to an example;

[0005] Figure 3 is a schematic illustration showing a representation of a graph based on a data stream;

[0006] Figure 4 is a schematic illustration showing data representative of a graph data structure according to an example;

[0007] Figure 5 is a schematic illustration of an apparatus for processing an electronic data stream according to an example;

[0008] Figure 6 is a flow diagram showing instructions encapsulated within a set computer-readable instructions according to an example; and

[0009] Figure 7 is a schematic chart showing ranking data according to an example.

DETAILED DESCRIPTION

[0010] Certain examples described herein allow for the useful processing of an electronic data stream. These examples may be seen as a form of unstructured data analysis. Certain examples convert data from the electronic data stream to a graph data structure. The graph data structure is then processed to enable particular entities associated with the data stream to be identified. This processing involves considering different time intervals, e.g. through the application of a sliding time window. A sliding time window may define a plurality of time intervals based on a varying time value and a predefined window size, e.g. a time of a modification to the graph data structure plus a predefined time period.

[0011] In one described case, nodes in the graph data structure represent topics or labels that are applied to, or are present in, the electronic data stream. In this case, users that are historically associated with a large growth in the use of particular topics or labels are identified through a temporal analysis of the graph data structure. These users may be seen as “early adopters” that may be monitored and/or analyzed to predict future large growth topics or labels. In another case, the data stream may comprise log records and entities associated with the data stream may comprise particular computer devices that are identified to be associated with a change in data stream. These devices may be deemed “problematic”. Similarly, information retrieval systems wherein documents are linked may be analyzed using the methods and apparatus described herein to identify and/or predict trends and future changes. Certain examples describe herein enable large bodies of data to be processed in a useful manner. For example, they may use historical data to make intelligent predictions about future system states.

[0012] Figure 1 shows a method 100 for analyzing a dynamic electronic data stream according to an example. In this case, the data stream is dynamic as it changes over time, e.g. may be represented as a sequence of data updates. For

example, the data stream may comprise a document that is updated by different users over time and/or may comprise a database of data updates that are added over time. A data update may be associated with a particular action from a set of available actions. For example, an action may be the addition, deletion or edit of at least a portion of a markup language document or text comment. In this case, the data stream is electronic as it is composed of electronic data, e.g. data that is processed by at least one computer device. This electronic data may comprise text data, markup language data, dictionary data and/or database data. The electronic data may be stored on a storage device and processed in the memory of a computer device. An example of data representative of a data update from a data stream is shown in Figure 2 and is described in more detail later below.

[0013] At block 110, data from the data stream is processed to generate a graph data structure. Data from the data stream may be accessed from a storage device and/or received from a computer device as a message stream. In one case, data from the data stream may be obtained by performing a query on at least one database. For example, if the data stream relates to comments or posts on a website, data associated with each comment or post may be stored under fields of a database. Alternatively, data may comprise a series of documents (or document portions), such as markup language or word processing documents. The graph data structure is representative of a graph with nodes and edges. An example graph is described with reference to Figure 3 later below. In one case, the graph data structure comprises at least one edge representation, each edge representation having data identifying the nodes that are linked by the edge. An example graph data structure is described with reference to Figure 4 later below.

[0014] At block 120, a sliding time window is applied to the graph data structure to generate a plurality of temporal snapshots of the graph data structure. In this case, each temporal snapshot represents a respective time interval set by a defined window size. For example, a start time may be defined for the system generating the data stream. This may be the upload and/or construction of a particular comment stream and/or website. A number of time intervals may then be defined based on changes to the data stream. For example, the addition of new

edge representations may trigger the application of the sliding window. In one case, a temporal snapshot may only be generated if a significant change is detected in a particular window. For example, a sliding action may be applied when any new edge representation is detected; this may result in a graph data structure being generated for the time period equal to the timestamp of the new edge representation plus a defined window size.

[0015] In one case, at least one statistical (“change”) metric may be computed that is compared to a respective at least one predefined threshold to determine if a “significant” change has occurred with the new time period. In one case the change metric may comprise a determination of a number of edge representations that are added in the particular window. If the comparison indicates that a “significant” change has not occurred, e.g. a statistical metric is below a defined threshold, then the graph data structure for the time period may not be further processed and a temporal snapshot may not be generated. If the comparison indicates that a “significant” change has occurred, e.g. a statistical metric is above a defined threshold, then the graph data structure for the time period may be further processed and a temporal snapshot may be generated and stored, e.g. using a “store” operation as defined in more detail below.

[0016] At block 130, each temporal snapshot is processed to determine at least ranking data indicative of a ranking for node representations in each temporal snapshot. This may comprise ranking nodes in the graph data structure. A ranking function may be applied to node representations in each temporal snapshot. The ranking function may be based on, amongst others, the PageRankTM function, the Hyperlink-Induced Topic Search (HITS) function and the Stochastic Approach for Link-Structure Analysis (SALSA) function. These ranking functions are configured to be applied to a static graph data structure. As such, each temporal snapshot may be considered a static graph data structure for this analysis. In a case where the nodes of the graph data structure represent users and topics, the ranking function may rank topic node representations within each temporal snapshot. In one case, a clustering function may be applied at this block to group nodes by different factors. This block may also comprise storing the result of the processing

and/or retrieving previously stored results. In one case, the processing at block 130 may be applied sequentially when generating and/or storing a temporal snapshot. For example, if a “significant” change is detected, part of generation of a temporal snapshot may comprise determining ranking data for the snapshot. The ranking data may thus form part of any stored temporal snapshot. This may then be followed at a later point in time by the further application of the sliding window and the conditional generation of a temporal snapshot for a new time interval. The processing at block 130 may comprise computing ranking data for each node representation in each temporal snapshot, i.e. on a static basis. The output of any ranking function may comprise values that are assigned to each vertex in the graph data structure, e.g. one ranking value per node per snapshot.

[0017] At block 140, ranking data that is computed and/or retrieved at block 130 is processed to identify a time period featuring a change in the ranking. For example, an ordered ranking of a set of topics, output in relation to each temporal snapshot after the processing at block 130, may be analyzed to determine changes in the ranking. If each node representation in each temporal snapshot has an associated ranking value, e.g. as computed in block 130, then a signal of values over the time period of the set of temporal snapshots may be obtained for each node representation. Figure 7 shows a schematic example of such a signal for a given node representation. In Figure 7 a ranking value (“*R*”) is represented by the y-axis. This ranking value changes with time, as represented by the x-axis (i.e. Figure 7 is time-series representation of changes in a ranking value of a node representation). In one case, preprocessing operations may be applied to the ranking value, e.g. smoothing, resampling and/or the determination of a moving average.

[0018] At least one criterion may be set to detect a change, e.g. a change in the signal of values as shown in Figure 7. For example, a criterion may comprise a continued upward movement in the ranking results (e.g. by *X* places per time interval over at least *Y* time intervals). In Figure 7, a peak detection function is applied to the signal of values to detect at least one peak. Such a peak detection function may process the signal of values to determine a time where a first

derivative of the signal is zero. Figure 7 shows a maximum peak detected at point P, which has a ranking value of F and occurs at time $t=T_1$. This identifies an end period of significant growth. The time period featuring a change in the ranking may be computed from such a peak value. For example, a start of the growth may be computed as the “base” of a fitted peak, e.g. a time following a previous trough and/or a time when the first derivative passes a given threshold. In Figure 7 a start of the time period is identified as T_0 . Hence, the time period featuring a change in the ranking is the “growth period” shown in Figure 7, which extends from T_0 to T_1 . In one case, to reduce the effects of short term trends, ranking data may be divided into a plurality of equal time intervals and a highest ranking growth determined for each equal time interval.

[0019] At block 150, a set of node representations associated with the change in the ranking are identified based on a function of the edge representations for the time period. With reference to Figure 7, the time period in that example is the “growth period” which extends from T_0 to T_1 . With reference to Figure 7, one example function is shown below:

$$M = -\log(F) * \sqrt[p]{\sum_i \left(-\log \left(\frac{t_i - T_0 + \varepsilon}{T_1 - T_0 + \varepsilon} \right) \right)^p}$$

where M is a metric value for a participant (e.g. a user); F is the peak value of the ranking signal; ε is a small configurable factor used for the edge case of $t_i=T_0$; t_i represents times that the participant occurs during the time period (e.g. for $i=0,1,\dots,n-1$ – these may be times that a user posted or otherwise interacted within the time period); T_0 and T_1 are the start and end times of the time period; and p represents a configurable scaling factor. The factor “ p ” may be used to scale the influence of being early in the growth period. For example, if $p=1$, there is linear scaling with a time distance from T_0 (e.g. node representations added at time further away from T_0 have a lower weight). If p is greater than 1 then there is a non-linear weighting for participant involvement closer to T_0 . Using this equation node representations may be identified that have a value of M above a predefined threshold.

[0020] In one case the set of node representations comprises user node representations; as such the set of node representations may be seen to be representative of “early adopters”, i.e. users that historically have been statistically correlated with topics that rise in ranking. In one case, the set of user node representations associated with the change in the ranking may be further processed in relation to a set of topic node representations to determine a metric based on a proportion of the set of user node representations that are coupled to each of the set of topic node representations. This metric may be seen as indicative of a predicted future growth of a topic associated with a given topic node representation. For example, the metric may be used for trend detection, based on an assumption that a node’s importance is expected to grow proportionally to the measure of “early adopters” that are linked to the node. In one case, the metric may mathematically or statistically summarize the number of node representations from the set of node representations that have recently been linked to a given node.

[0021] For example, the ranking signal shown in Figure 7 may represent an output of a ranking function for a node representation associated with the topic “Scala” (“Scala” being a particular programming language). In this case, “early adopters” are users that discussed this topic in the “growth period”; e.g., in technical terms, users identified in post data labelled with the topic “scala” that posted between time periods T_0 and T_1 . The metric function shown above has a higher value for node representations that post closer to time T_0 , e.g. that are associated with the growth at an early stage. Node representations defined as “early adopters” are thus those that are repeatedly associated with early periods of growth across at least one topic, e.g. as weighted by the above function.

[0022] The example method 100 of Figure 1 provides logic and efficiency advantages when compared to comparative graph topology or community discovery functions. On application to a number of collaborative websites the method has demonstrated detection of trends that are subsequently confirmed by recorded data. By using a sliding time window, e.g. analyzing different time intervals as if they are complete graph data structures, the method may be applied

to highly dynamic data streams. These data streams are typically difficult to analyze using comparative ranking functions as these functions are configured for application to a single static graph data structure. In these cases, the data stream may be large, meaning that due to processing and memory limitations it is not possible to process the data stream as a single graph data structure. As such, the described example method overcomes these processing and memory limitations by breaking the large single graph data structure into a number of temporal snapshots, wherein each of these temporal snapshots may be processed separately. Moreover, temporal snapshots and ranking results associated with the processing of those temporal snapshots may be stored in non-volatile storage and retrieved when necessary. This overcomes the need to load a large graph data structure into volatile memory (e.g. random access memory).

[0023] Figure 2 shows an example 200 of a data update 210 from a data stream. Figure 2 is provided as an example to help explain the method of Figure 1 and the apparatus of Figure 6 and should not be seen as limiting. For example, the format of the data update and constituent portions may vary according to the characteristics of each individual implementation. The data update 210 of Figure 2 is shown as being implemented in a pseudo-markup language. In other examples, similar information may be stored in fields of a database and/or a data format such as JavaScript Object Notation (JSON).

[0024] In the example 200 of Figure 2, the data update 210 is defined between two “<data_update></data_update>” markup tags. In the example 200 there are four sets of child tags. These include: a set of tags defining at least one topic 220 (“<topics></topics>”); a set of tags defining at least one user 230 (“<users></users>”); a set of tags defining text for the data update 240 (“<text></text>”); and a set of tags defining a time associated with the data update 250 (“<time></time>”). The topics in this example comprise three labels: “#serialport”, “#c++” and “#phone”. These indicate subject matter labels associated with the text of the data update. These labels may be selected by the posting user from a list, entered by the user or determined automatically through natural language processing of the post text data. This example shows that a single user

“user123” is associated with the data update. The text of the data update is a post by the user. The time associated with the data update may comprise a time a last action was performed in relation to the data update, e.g. the time the text was first published on the Internet and/or edited. The example of Figure 2 thus shows a data update from a sequence of data updates that forms a data stream where each data update at least comprises a timestamp, data indicative of at least one user and data indicative of at least one topic.

[0025] Figure 3 shows an example of a graph 300 that may be constructed based on data updates similar to that shown in Figure 2. Figure 4 then shows an example 400 of a graph data structure 410 that may be used to define at least a portion of the graph shown in Figure 3.

[0026] The graph 300 of Figure 3 has 5 nodes and 4 edges. Two nodes 310, 350 relate to topics, e.g. they each may relate to a different one of labels “#serialport”, “#c++” and “#phone” from Figure 2. Node 310 is shown as topic 1 (T1) and node 350 is shown as topic 2 (T2) in Figure 3. Three nodes 320, 330 and 340 then relate to users, e.g. they may each relate to a different one or users as defined by the tags 230 in Figure 2. Node 320 is shown as user 1 (U1), node 330 is shown as user 2 (U2) and node 340 is shown as user 3 (U3). The nodes of Figure 3 are linked by edges 315, 325, 335 and 345. User node U1 320 is linked to topic node T1 310 by edge 315. User node U2 330 is also linked to topic node T1 310, this time by edge 325. User node U3 340 is linked to topic node T1 310 by edge 335 and is also linked to topic node T2 350 by edge 345. In this example, the edges indicate that a user was present in user tag 230 in a data update that also had a topic present in topics tag 220. For example, U1 may represent “user123” in Figure 2 and T1 may indicate the topic “#serialport”.

[0027] Figure 4 shows the data that may result from the processing of a data update such as 210 to define a graph data structure that implements a graph such as that shown in Figure 3. For example, the example 400 of Figure 4 may result from block 110 of Figure 1. Figure 4 again uses a pseudo-markup language to better explain the present example methods and apparatus; however, this format is not limiting and may vary between implementations.

[0028] In Figure 4 a graph data structure 410 is defined between two “<graph></graph>” tags. The graph data structure 410 has a plurality of child tags 415 that represent edges of a graph, e.g. that comprise edge representations. Two edge representations are shown in Figure 4: a first edge representation 415-A and a second edge representation 415-B. Each edge representation has a set of two or more node representations, in this example a tuple representing a source node and a destination node. These node representations may be defined in a separate list, array or dictionary data structure. Identifiers for the node representations are shown between “<node></node>” tags 420. The upper edge representation 415-A has a tuple “(U1, T1)” indicating that source node U1 is connected to destination node T1, e.g. as shown by edge 315 in graph 300. The edge representation 415-A in Figure 4 also has an edge weight, as represented by value 0.3 between value tags 435 (“<value></value>”). The edge weight may be provided to indicate a strength of a connection between a topic and a user. In one case this value may be 1 indicating a binary link. The edge weight may be used when a topic is determined automatically from a data update, e.g. it may be proportional to a frequency of terms defined as being associated with a particular topic. Each edge representation 415 in Figure 4 also has a timestamp 440 indicated between a set of timestamp tags (“<timestamp></timestamp>”). This may be generated based on the time 250 associated with the data update. It is representative of a date and/or time when an association between the nodes, in this example between the topic and the user, occurred. The second edge representation 415-B is shown to relate to edge 345 as shown in Figure 3, said edge associating user U3 and topic T2.

[0029] The graph data structure 410 as shown in Figure 4 may be generated by the processing at block 110 of Figure 1. Hence, said processing may comprise: representing data indicative of at least one user for the given data update as a user node representation in the graph data structure; representing data indicative of at least one topic for the given data update as a topic node representation in the graph data structure; generating a value of an edge representation in the graph data structure between the user node representation and the topic node

representation; and associating the timestamp of the data update with the edge representation.

[0030] In one case, the value of an edge representation may be set to a static value. In another case, the value of an edge representation, e.g. between a user node representation and a topic node representation, may be configured to decrease as a function of time that has passed since the timestamp. For example, the value may be calculated as a function of the timestamp 440. In this case, once a value of an edge representation drops below a predefined threshold, e.g. zero, it may be removed from a graph data structure. For example, when constructing a temporal snapshot between a start time and an end of a defined window time, any edge representations that have a value below the predefined threshold may be omitted from the graph data structure forming the temporal snapshot. In one case, a time associated with the time interval, e.g. one of the start, median and end time, may be compared to the timestamp of each edge to apply the function that decreases the value of the edge representation. Depending on the implementation the function may be discrete or continuous. The function is thus representative of the decreasing relevance of a connection between nodes, e.g. a topic and a user, as time passes.

[0031] In one case the sliding window that is applied comprises two parts: a window and a sliding action. In this case, the window may define a time period and a graph data structure generated for this time period represents a set of activity within the time period. This may comprise all the edge representations that are modified in the time period, e.g. all new data updates or modifications to existing data updates that occurred during the time period. The size of the window in this case, e.g. the length of the time period, may be set as a configurable system parameter. A value of this system parameter may be selected based on a particular implementation. The sliding action in this case is representative of a movement of the window in time. For example, the sliding action may be applied based on the addition of new edge representations to the graph data structure, wherein new edge representation result from modifications to the data stream, e.g. the addition, deletion or edit of data updates. For example, the starting time for an

n^{th} time period may be calculated from the time of the n^{th} edge representation timestamp, e.g. the time interval for this case is the n^{th} edge representation timestamp plus the defined window length.

[0032] More generally, the sliding action may be applied to generate temporal snapshots based on a timestamp of a new edge addition plus the defined window length. In another example, the time interval may alternatively represent time intervals between fixed times, such as days, months or years. In any case, when the sliding action is applied this may initiate the re-calculation of values of the edge representations that have timestamps within the new time interval resulting from the sliding action. As described above, if the values of the edge representations are a function of time, this may result in certain edge representations being omitted from the temporal snapshot that represents the graph data structure of the new time interval. For example, a new time interval may add certain edge representations that did not occur in a previous time interval and omit certain edge representations that have timestamps falling outside the new time interval.

[0033] One advantage of applying a sliding window as described in the examples above is that it enables “load” and “store” functions to be defined and implemented for the graph data structure. A “store” function may be configured to store, e.g. in a computer-readable storage medium, the graph data structure associated with temporal snapshots, e.g. data similar to that shown in Figure 4. For example, every time the sliding action described above is applied to generate a graph data structure associated with a new start timestamp plus a defined window size, this graph data structure may be stored using the “store” function. In one case, the “store” function may store a graph data structure associated with temporal snapshots in non-volatile memory, e.g. magnetic or solid-state storage, during processing that is being performed using volatile memory, e.g. random access memory. In one case, a “store” function may be applied automatically based on predefined criteria. For example, these criteria may comprise determining a number of edge representations that were modified since a last “store” operation and/or may comprise calculating statistical metrics that may be compared to predefined thresholds to determine if a “significant” change in the

graph data structure has occurred. The criteria may be applied when performing processing associated with a new time interval as discussed above. The “store” function may be calibrated and configured based on the system specifications of an implementing apparatus, e.g. based on the speed of storage in non-volatile memory, the available amounts of volatile and non-volatile memory, the size of an average temporal snapshot etc. Additionally the “store” function may be calibrated and configured in association with a “load” function that retrieves previously stored data. In certain cases, the “store” function may also store the result of block 130 as applied to a temporal snapshot, e.g. a ranking data value for nodes in the graph data structure of the temporal snapshot.

[0034] A “load” function may be configured to receive a timestamp and construct a graph data structure relevant to the timestamp. In one case, the “load” function may be configured to receive a start time in the form of a timestamp, and construct an appropriate temporal snapshot. This is performed using at least one graph data structure previously stored with the “store” function. The “load” and/or “store” functions may be configured such that a “load” function takes a time below a predefined threshold and such that the total size of the stored and/or loaded data is below at least one predefined threshold. The threshold may be configured based on the system specifications of an implementing apparatus, e.g. based on similar characteristics to those described as examples above. For example, it may be configured by determining when the sliding action is applied and/or by setting the size of the window.

[0035] In one case, the “load” function may also be configured to return results associated with the calculation of ranking data for each temporal snapshot. For example, these results may be stored as part of a given “store” function. The results may be stored as data associated with a given time interval. The “load” function may then be configured to receive a timestamp and/or a time interval and return previously calculated results, e.g. results associated with at least one temporal snapshot that contains and/or overlaps with the received timestamp and/or time interval. For example, a “load” function may have an associated query

function that is configured to provide temporally relevant results as metadata in response to a query comprising time variables.

[0036] The above described adaptations may provide a benefit when managing large amounts of data (e.g. quantities of data beyond a given threshold). For example, a social network or online community may have millions, if not billions, of data updates over time. Moreover, the quantity of data increases as time passes. It then becomes difficult, if not impossible, to apply comparative ranking functions on this data, or apply known static graph analysis. The examples described herein provide technical methods to process data from a data stream that may be applied to large amounts of data by applying a sliding window that breaks a graph data structure down into smaller temporal snapshot that can be processed statically within the system constraints of example processing apparatus. The results from this processing can then be processed over time to identify salient nodes. The examples described herein are also able to scale with advances in technology; e.g. as advances are made in processing power and non-volatile memory the configuration and size of data associated with the “store” and “load” functions may be varied without modifying the underlying processing.

[0037] An output of the example method 100 may comprise at least one of: a list of node representations that are historically associated with a growth in edge representations over time; time-series metrics such as growth rate, volume of data updates and a ratio associated with node representations (e.g. the ratio in the inner logarithmic function in the above equation); node representations that are experiencing and/or are predicted growth; and node representations that are linked by edge representations to the node representations that are experiencing and/or are predicted growth. In the example of Figures 2, 3 and 4 these may respectively comprise: a list of users that are historically associated with a growth in edge representations over time, so called “early adopters”; time series metrics such as topic growth rate; topics that are currently and/or predicted to growth with regard to presence in data updates (e.g. as tagged topics 220), so-called “trending topics”; and topics that are related to the “trending topics”, e.g. that have a connection to the “trending topics” as represented by the graph data structure.

[0038] In one implementation, the example method 100 may be applied to an online software-engineering community. In this implementation, the data stream may comprise data retrieved using an application programming interface (API) provided by at least one computer device implementing the online software-engineering community. In this case, a data update may relate to a post on the online software-engineering community, which is retrievable, either individually or in a batch, via a request made to the API. In this case, users may comprise software engineers that are registered members of the online software-engineering community, e.g. that have a user account implemented by the computer device. Topics may comprise different tags applied to posts in the online software-engineering community. Trending topics may relate to different technologies that are associated with posts as tags. As such, the described examples, enable a prediction to be made as to which technologies may experience growth based on software engineers that are identified as being associated with historic growth in topics. In other implementations, the described examples may also be applied to scientific publication systems, information retrieval systems, social networks, and fault logging systems.

[0039] The case of users and topics as nodes in a graph data structure, wherein edges indicate that a user has posted data associated with a topic, is one example that has been provided for ease of explanation. In other examples, different entities may be associated with nodes of the graph data structure and different actions or events may be associated with edges. For example, if the data stream comprises telecommunication data, nodes in the graph data structure may represent user equipment or the users of such equipment (e.g. as identified using the international mobile subscriber identity – IMSI) and edges may represent calls that are made between said equipment or users. If the data stream results from an online social game, then nodes may represent participants in the game and edges may represent interactions in the game universe between said participants. In an e-commerce case, nodes may be representative of customer or product identifiers and edges may represent transactions.

[0040] Figure 5 shows a schematic illustration of an apparatus for processing a dynamic electronic data stream according to an example 500. The apparatus 510 may comprise at least one computing device. The apparatus of Figure 5 comprises an interface 520 configured to receive data from the data stream, a memory 530 and at least one processor 540. The interface may comprise a network interface arranged to receive data from the data stream over a network and/or may comprise a storage interface arranged to receive data from at least one storage device. The data stream may have characteristics as described above and/or may have a general programmatic form as shown in Figure 2. The processor may comprise central processing units arranged to implement a node identifier 550 stored in memory 530. The node identifier 550 may be embodied in a set of machine readable instructions.

[0041] The node identifier 550 may be configured to cause the apparatus 510 to perform a number of operations. These operations include causing the apparatus 510 to: parse the data received at the interface and generate a graph data structure, the graph data structure indicating nodes coupled by edges; determine when new edges are added to the graph data structure; process a set of time windows (e.g. apply a sliding time window) to determine if a change metric indicates a predefined change in the graph data structure, each time window being set based on a predefined window size and a start time indicative of a time when a respective new edge is added to the graph data structure; generate a plurality of temporal snapshots for time windows wherein the change metric indicates a predefined change in the graph data structure, wherein each temporal snapshot comprises at least ranking data indicative of a ranking for nodes indicated in each temporal snapshot; and process the ranking data to identify a set of nodes associated with a change in the ranking data. In this case reference to nodes and edges indicate data representing those entities, e.g. node and edge representations.

[0042] In one case, as described above with reference to Figure 4, the graph data structure comprises data indicative of at least one edge representation, each edge representation comprising: node data indicating at least two node

representations derived from the data stream, the at least two node representations being coupled by the edge representation; and time data indicating a time at which the at least two node representations are deemed to be coupled by the edge representation.

[0043] The apparatus 510 may be arranged to output a list comprising the set of node representations associated with the change in the ranking as a function of the edge representations for the time period. This may be a set of registered users, computer systems or other entities. The apparatus 510 may also be arranged to output a metric based on a proportion of the set of node representations that are coupled to a given node representation, wherein the metric is indicative of a predicted future growth of an entity or topic associated with the given node representation. As discussed above, the value of an edge representation between node representations may decrease as a function of time passed since a timestamp and wherein the edge representation is removed from the graph data structure on application of the sliding time window based on a comparison of the value of the edge representation and a pre-defined threshold.

[0044] Machine readable instructions embodying the node identifier 550 may be retrieved from a machine-readable media, e.g. any media that can contain, store, or maintain programs and data for use by or in connection with an instruction execution system. In this case, machine-readable media can comprise any one of many physical media such as, for example, electronic, magnetic, optical, electromagnetic, or semiconductor media. More specific examples of suitable machine-readable media include, but are not limited to, a hard drive, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory, or a portable disc.

[0045] Figure 6 shows an example of a method 600 for processing a data stream. This method 600 may, for example, be applied when processing a new data update from the data stream. The method 600 may be embodied on a set of computer readable instructions, e.g. may be used to implement node identifier 550. These computer readable instructions may be provided on a non-transitory

computer-readable storage medium and may be arranged to be processed by the processor 540.

[0046] At block 610, a data update is received from the data stream. As described above, the data update may comprise a timestamp, data indicative of a user, and data indicative of a topic. At block 620, the data update is processed to generate edge data in a graph data structure. The edge data may comprise data 415 as shown in Figure 4 and may represent a coupling between user nodes and topic nodes. In this case, the data update is processed and the data indicative of the user, e.g. data 230 in Figure 2, is stored as a user node representation, e.g. as referenced by the data stored within “<nodes></nodes>” tags 420 in Figure 4. In this processing, the data indicative of the topic, e.g. at least one of the labels in data 220 in Figure 2, is stored as a topic node representation, e.g. as also referenced by the data stored within “<nodes></nodes>” tags 420 in Figure 4. A timestamp is further associated with the edge data, which may comprise processing time data 250 in Figure 2 and converting it into edge timestamp data 440 in Figure 4. A value for the edge data may also be computed and/or otherwise set.

[0047] At block 630, a time interval associated with the timestamp is determined. For example, this may comprise determining a time interval since a last edge was added to the graph data structure, e.g. the timestamp associated with the edge data plus a defined time interval window. In one case, for any new edge data in a graph data structure representing the data stream, a sliding time window may be shifted and a test may be applied to determine if a change in the graph data structure has occurred. For example, this may comprise determining a change metric, such as determining if a predetermined number of edge representations have been modified (e.g. added, deleted or edited) and/or calculating statistical metrics that may be compared to predefined thresholds to determine if a “significant” change in the graph data structure has occurred. This determination is shown at block 635. In this case, the following blocks may be performed if a change in the graph data structure has occurred.

[0048] At block 640, ranking data for nodes in the graph data structure is determined. This may comprise determining ranking data for nodes in the time interval determined in block 630. This block may further comprise a “store” function that stores a temporal snapshot based on the determined time interval. Block 640 may comprise the calculation of the previously described ranking functions for the time interval, e.g. static processing on the temporal snapshot.

[0049] At block 650, a time period is identified featuring a change in the ranking data over a plurality of time intervals that includes the time interval associated with the timestamp. In other words, dynamic values, e.g. values that change with time, of the results from a static analysis of each time interval are compared and processed to determine changes in the ranking data. At block 660, user nodes associated with the change in the ranking are identified as a function of the edge data for the time period. This may be performed using the metric equation set out above. This may comprise updating an existing list of users based on the processing initiated by the newly added edge data. This block may also comprise updating a list of topics that are associated with these users, e.g. as described above.

[0050] The preceding description has been presented to illustrate and describe examples of the principles described. This description is not intended to be exhaustive or to limit these principles to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

CLAIMS

What is claimed is:

1. A method for analyzing a dynamic electronic data stream comprising:
 - processing data from the data stream to generate a graph data structure, the graph data structure indicating node representations coupled by edge representations;
 - applying a sliding time window to the graph data structure to generate a plurality of temporal snapshots of the graph data structure, a sliding time window defining a plurality of time intervals based on a varying time value and a predefined window size, each temporal snapshot representing a respective time interval;
 - processing each temporal snapshot to determine at least ranking data indicative of a ranking for node representations in each temporal snapshot;
 - processing the ranking data to identify a time period featuring a change in the ranking; and
 - identifying a set of node representations associated with the change in the ranking as a function of the edge representations for the time period.
2. The method of claim 1, wherein:
 - the data from the data stream comprises a sequence of data updates, each data update comprising:
 - a timestamp,
 - data indicative of at least one user, and
 - data indicative of at least one topic; and
 - processing the data from the data stream comprises, for a given data update:
 - representing data indicative of at least one user for the given data update as a user node representation in the graph data structure;

representing data indicative of at least one topic for the given data update as a topic node representation in the graph data structure;

generating a value of an edge representation in the graph data structure between the user node representation and the topic node representation; and

associating the timestamp of the data update with the edge representation.

3. The method of claim 2, wherein:

processing each temporal snapshot comprises determining at least ranking data indicative of a ranking for topic node representations in each temporal snapshot; and

identifying a set of node representations associated with the change in the ranking comprises identifying a set of user node representations,

wherein the set of user node representations are indicative of a set of early adopters for at least one topic represented by the topic node representations.

4. The method of claim 3, comprising, for a given topic node representation:

processing the set of user node representations associated with the change in the ranking in relation to the given topic node representation to determine a metric based on a proportion of the set of user node representations that are coupled to the given topic node representation,

wherein the metric is indicative of a predicted future growth of a topic associated with the given topic node representation.

5. The method of claim 2, the value of the edge representation between the user node representation and the topic node representation decreases as a function of time passed since the timestamp.

6. The method of claim 5, wherein the edge representation is removed from the graph data structure when applying the sliding time window based on a comparison of the value of the edge representation and a pre-defined threshold.
7. The method of claim 1, wherein applying a sliding time window comprises:
 - detecting a new edge representation in the graph data structure;
 - determining a new time interval based on a time the new edge representation occurred and a predetermined window size;
 - determining a change metric for the graph data structure based on a comparison of the graph data structure for the new time interval and the graph data structure for at least one previous time interval;
 - responsive to the change metric indicating a predefined change in the graph data structure, generating a temporal snapshot and storing the temporal snapshot in a data storage device.
8. The method of claim 1, wherein processing each temporal snapshot comprises applying clustering.
9. An apparatus arranged to process a dynamic electronic data stream comprising:
 - an interface configured to receive data from the data stream;
 - a memory;
 - at least one processor, the processor being arranged to process machine readable instructions stored in memory to cause the apparatus to:
 - parse the data received at the interface and generate a graph data structure, the graph data structure indicating nodes coupled by edges;
 - determine when new edges are added to the graph data structure;
 - process a set of time windows to determine if a change metric indicates a predefined change in the graph data structure, each time window being set based on a predefined window size and a time value

indicative of a time when a respective new edge is added to the graph data structure;

generate a plurality of temporal snapshots for time windows where the change metric indicates a predefined change in the graph data structure, wherein each temporal snapshot comprises at least ranking data indicative of a ranking for nodes indicated in the temporal snapshot; and

process the ranking data to identify a set of nodes associated with a change in the ranking data.

10. The apparatus of claim 9, wherein the graph data structure comprises data indicative of at least one edge, said data comprising:

node data indicating at least two nodes derived from the data stream, the at least two nodes being coupled by the edge; and

time data indicating a time at which the at least two nodes are deemed to be coupled by the edge.

11. The apparatus of claim 9, wherein the data from the data stream comprises a sequence of data updates, each data update comprising:

a timestamp,

data indicative of at least one user, and

data indicative of at least one topic; and

the instruction instructing the processor to process the data from the data stream comprises instructions to, for a given data update:

represent data indicative of at least one user for the given data update as a user node in the graph data structure;

represent data indicative of at least one topic for the given data update as a topic node in the graph data structure;

generate a value of an edge in the graph data structure between the user node and the topic node; and

associate the timestamp of the data update with the edge.

12. The apparatus of claim 11, wherein the processor is arranged to process machine readable instructions stored in memory to cause the apparatus to:

determine at least ranking data indicative of a ranking for topic nodes in each temporal snapshot; and

identify a set of user nodes associated with the change in the ranking,

wherein the set of user nodes are indicative of a set of early adopters for at least one topic represented by the topic nodes.

13. The apparatus of claim 12, wherein the processor is arranged to process machine readable instructions stored in memory to cause the apparatus to:

process the set of user nodes associated with the change in the ranking in relation to the given topic node to determine a metric based on a proportion of the set of user nodes that are coupled to the given topic node representation,

wherein the metric is indicative of a predicted future growth of a topic associated with the given topic node.

14. The apparatus of claim 9, wherein the value of the edge between the user node and the topic node decreases as a function of time passed since the timestamp and wherein the edge is removed from the graph data structure when processing the set of time windows based on a comparison of the value of the edge and a pre-defined threshold.

15. A non-transitory computer-readable storage medium comprising a set of computer-readable instructions stored thereon, which, when executed by a processing system, cause the processing system to:

receive a data update from a data stream, the data update comprising:

a timestamp,

data indicative of a user, and

data indicative of a topic;

process the data update to generate edge data in a graph data structure, the edge data representing a coupling between user nodes and topic nodes wherein:

- the user node represents the data indicative of the user,
- the topic node represents the data indicative of the topic, and
- the timestamp is associated with the edge data;

- determine a time interval associated with the timestamp;

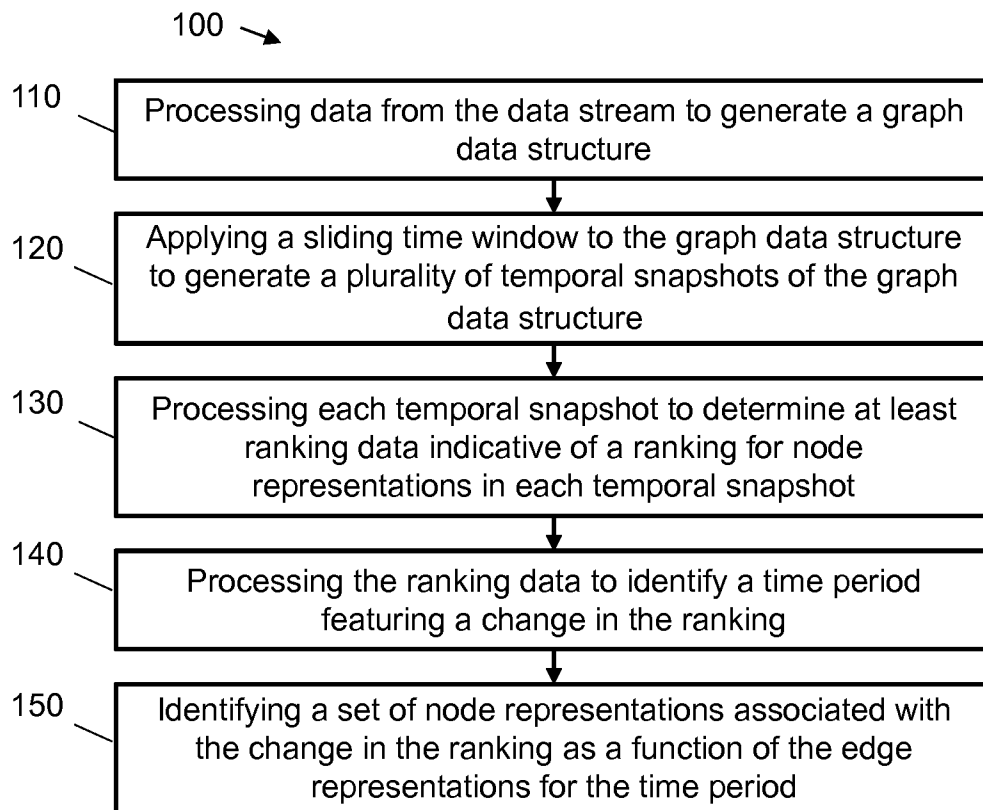
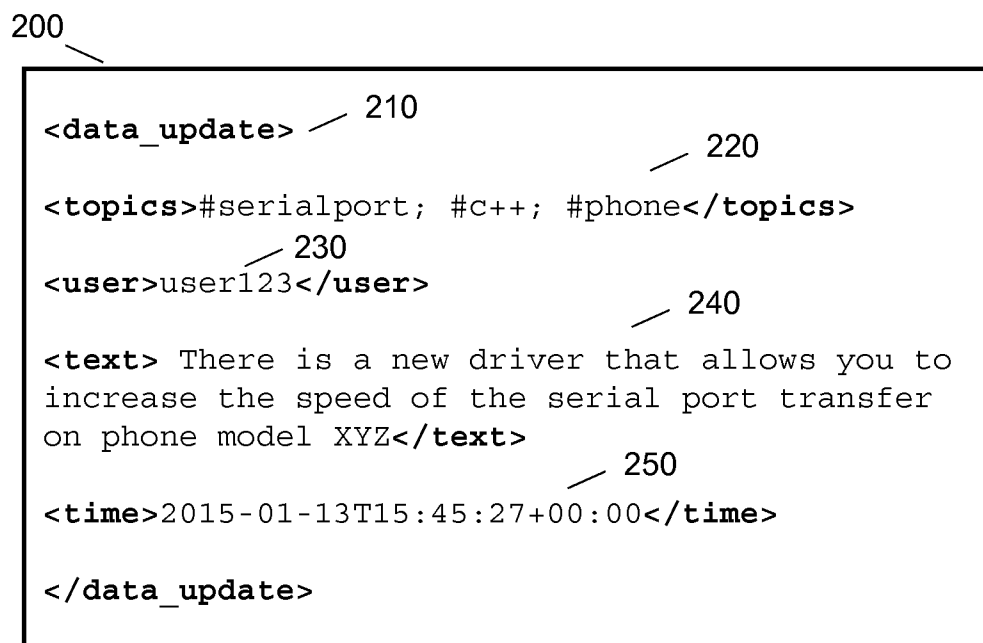
- determine if a change metric indicates a predefined change in the graph data structure and if so:

- determine, for the time interval, ranking data for nodes in the graph data structure;

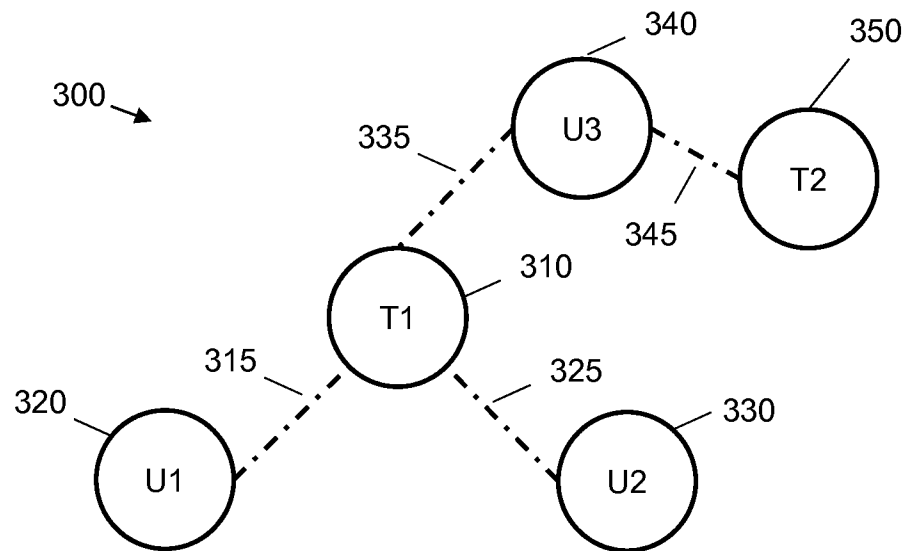
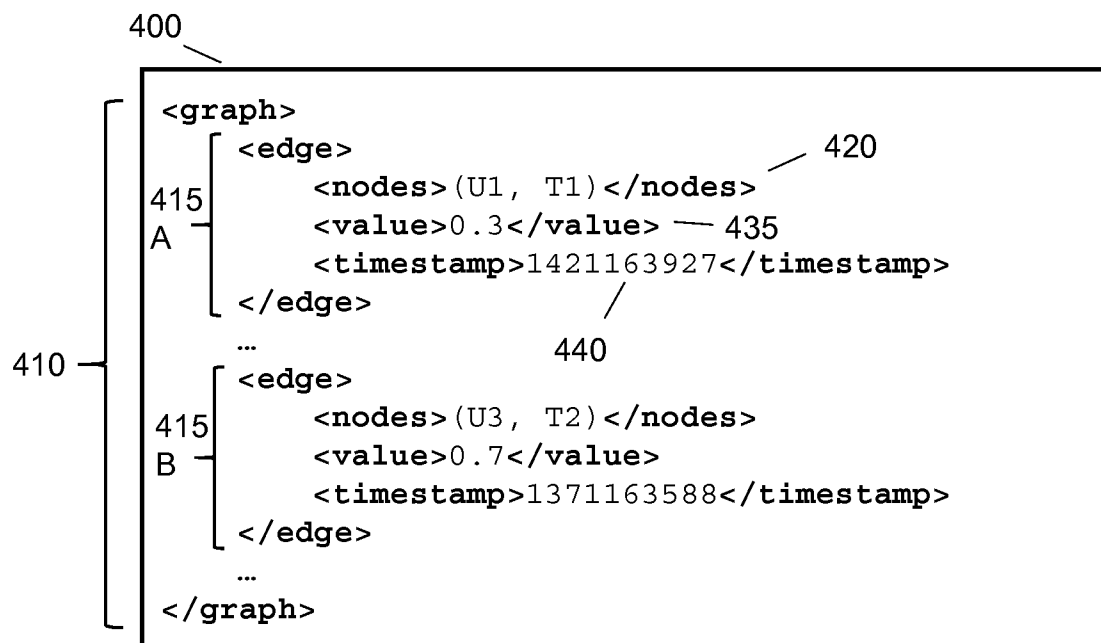
- process ranking data for a plurality of time intervals that includes the time interval associated with the timestamp to identify a time period featuring a change in the ranking data; and

- identify user nodes associated with the change in the ranking as a function of the edge data for the time period.

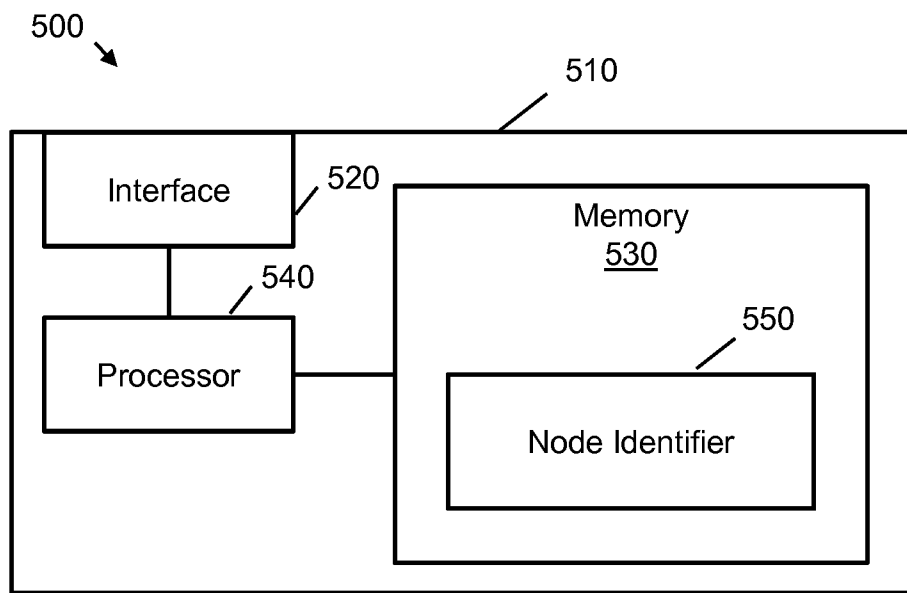
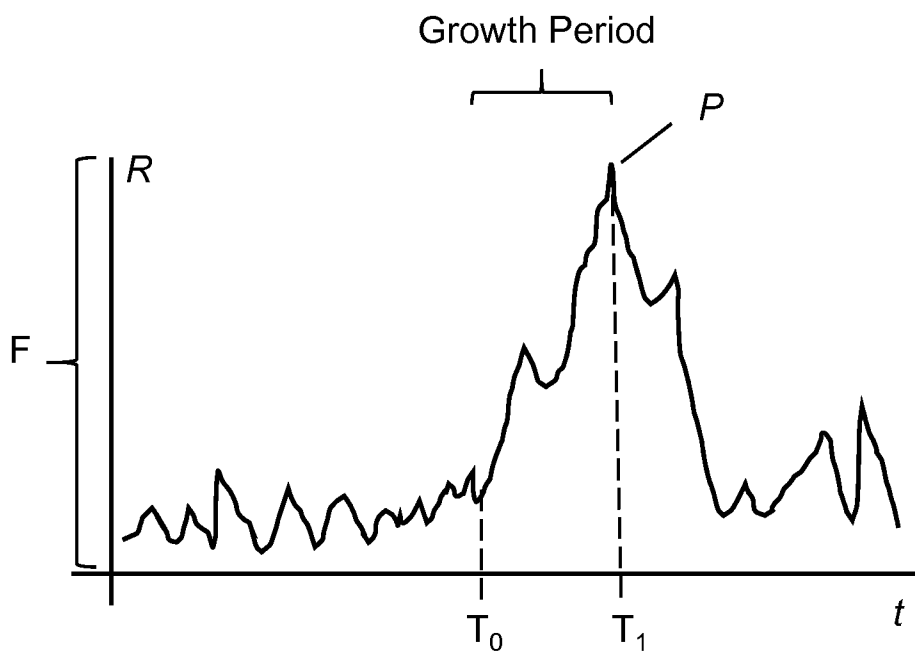
1 / 4

**Fig. 1****Fig. 2**

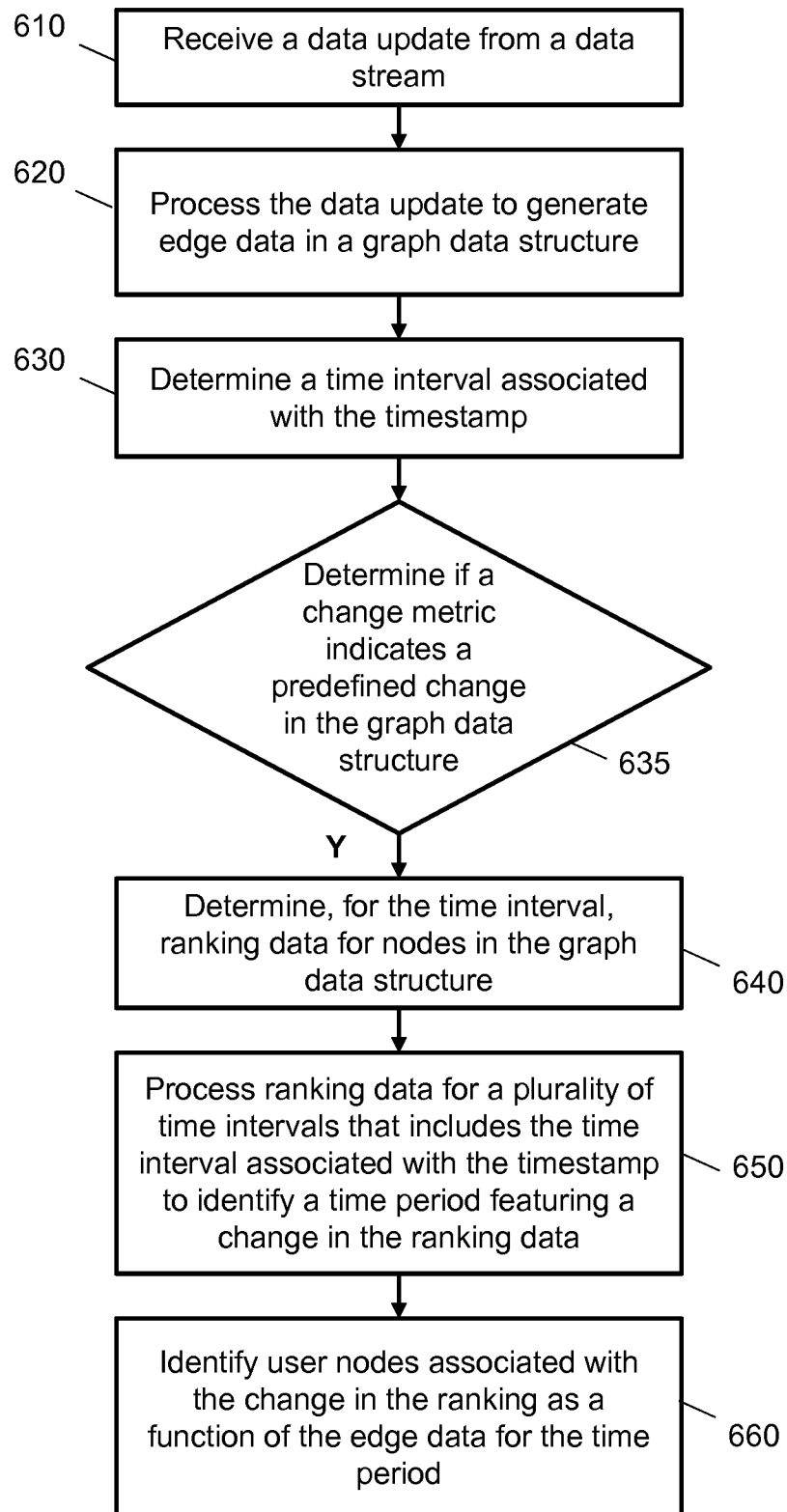
2 / 4

**Fig. 3****Fig. 4**

3 / 4

**Fig. 5****Fig. 7**

4 / 4

**Fig. 6**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013554**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/30(2006.01)i, G06F 17/00(2006.01)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 17/30; G06F 3/0481; H04L 12/18; G06F 17/00

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: graph data structure, sliding time window, temporal snapshot, data update, topic, ranking, and similar terms.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2013-149381 A1 (MICROSOFT CORPORATION) 10 October 2013 See paragraphs [0001], [0017]-[0020], [0031]-[0034], and [0046]-[0047]; and figures 1, 3, and 5.	1-15
A	US 2013-0339357 A1 (AHMED S. ELDAWY et al.) 19 December 2013 See paragraphs [0021]-[0022] and [0027]-[0030]; claim 1; and figures 2A-2B and 4-5.	1-15
A	US 2015-0026158 A1 (KENT STATE UNIVERSITY) 22 January 2015 See paragraphs [0108]-[0116].	1-15
A	US 2014-0344718 A1 (JEFFREY ALAN RAPAPORT) 20 November 2014 See paragraph [0141] and figure 1A.	1-15
A	US 2013-0232174 A1 (CONCURIX CORPORATION) 05 September 2013 See paragraphs [0113]-[0120] and figure 5.	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 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

10 September 2015 (10.09.2015)

Date of mailing of the international search report

10 September 2015 (10.09.2015)

Name and mailing address of the ISA/KR

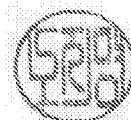
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/013554

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2013-149381 A1	10/10/2013	CN 104205095 A EP 2834755 A1 US 2015-0154262 A1	10/12/2014 11/02/2015 04/06/2015
US 2013-0339357 A1	19/12/2013	US 2013-0339355 A1 US 8635224 B2 US 8768929 B2	19/12/2013 21/01/2014 01/07/2014
US 2015-0026158 A1	22/01/2015	WO 2013-116308 A1	08/08/2013
US 2014-0344718 A1	20/11/2014	US 2012-0290950 A1 US 8676937 B2	15/11/2012 18/03/2014
US 2013-0232174 A1	05/09/2013	WO 2014-120263 A1	07/08/2014