

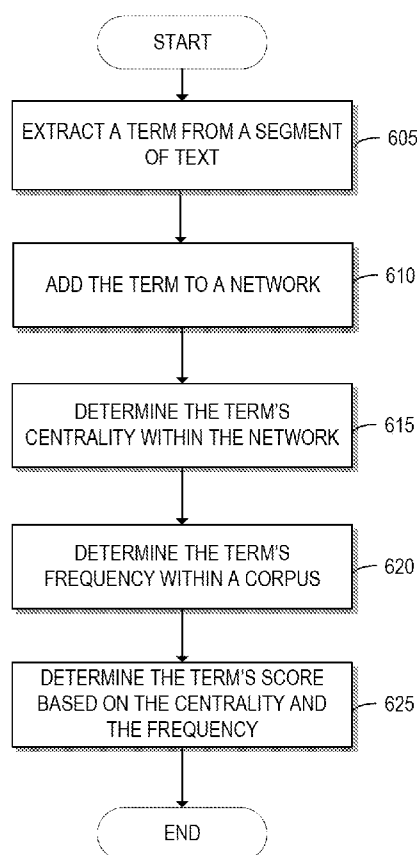


- (51) International Patent Classification: *G06F 17/27* (2006.01) *G06F 17/30* (2006.01)
- (21) International Application Number: PCT/US2015/034590
- (22) International Filing Date: 6 June 2015 (06.06.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P** [US/US]; Houston, Houston, TX 77070 (US).
- (72) Inventors: **WU, Shanchan**; 1501 Page Mill Rd, Palo Alto, CA 94304-1100 (US). **SIMSKE, Steven, J.**; 3404 E Harmony Rd, Ft. Collins, CO 80528-9544 (US).
- (74) Agents: **KOGAN, Daniel** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E Harmony Rd., Mail Stop 35, Fort Collins, CO 80528-9599 (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,

[Continued on next page]

(54) Title: TERM SCORES

600



(57) Abstract: Examples disclosed herein relate, among other things, to a method. The method may include extracting a term from a segment of a text, where the text includes a plurality of segments, adding the term to a network of terms by connecting the term to a set of other terms extracted from the segment, and determining a score of the term based on the terms centrality within the network and/or based on the terms frequency within a corpus of texts.

FIG. 6



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.

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))

- (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,

TERM SCORES

BACKGROUND

[0001] With today's ever growing availability of digital texts such as articles, textbooks, fiction or non-fiction books, etc., it may be useful to automatically obtain a set of key terms associated with each text. The key terms may allow users, for example, to quickly determine the main concepts and topics discussed in the text. Determining the set of representative key terms may also enable better indexing, classification, sequencing, and querying of digital texts.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0002] The following detailed description references the drawings, wherein:
- [0003] FIG. 1 is a block diagram of an example computing system;
- [0004] FIG. 2 is another block diagram of an example computing system;
- [0005] FIG. 3A illustrates an example arrangement of terms in segments;
- [0006] FIG. 3B illustrates an example network of terms;
- [0007] FIG. 4A is another illustration of an example arrangement of terms in segments;
- [0008] FIG. 4B is another illustration of an example network of terms;
- [0009] FIG. 5 illustrates an example display of an example computing system.
- [0010] FIG. 6 shows a flowchart of an example method; and
- [0011] FIG. 7 is a block diagram of an example computing device.

DETAILED DESCRIPTION

[0012] As mentioned above, determining a set of key terms that accurately represent or categorize a digital text may allow the readers to quickly determine the main concepts and topics discussed in the text. Such key terms may be extracted from the text itself. To determine whether a particular term (e.g., a word or a sequence of words) in the text should be used as a key term, the term's frequency of appearance in the text may be compared to the term's frequency of appearance in a larger body or "a corpus" of texts. For example, a term's frequency (TF) within a given text may be multiplied by the term's inverse document frequency (IDF), where IDF corresponds to the inverse of the term's frequency in a corpus of texts. A higher TF*IDF value may indicate that the term appears in the text unusually often, which may indicate that the term has an unusually high importance in the current text and could potentially be a good (descriptive) key term. Relying only on the TF*IDF may not produce the best key terms for some types of texts. For example, a textbook may often use a very particular and specialized vocabulary, making it difficult to determine and obtain a corpus of texts that would enable the TF*IDF measure to produce descriptive key terms.

[0013] Examples disclosed herein describe, among other things, a computing system. The computing system may include, among other things, a term extraction engine to extract a plurality of terms from a text, where the text includes a plurality of segments; a network construction engine to connect each term in the plurality of terms to each term included in any segment, in the plurality of segments, in which the term is also included; and a term ranking engine to calculate a score for each term in the plurality of terms based on the term's connections to other terms in the plurality of terms.

[0014] FIG. 1 is a block diagram of an example computing system 100. Computing system 100 may include one or more computing devices, where a computing device may include a smartphone, cell phone, tablet, laptop, desktop, server, application-specific computing device, any other processing device or equipment, or a combination thereof. Computing system 100 may include a term extraction engine 112, a network

construction engine 113, and a term ranking engine 114. FIG. 2 is a block diagram of another example computing system 100. As illustrated in FIG. 2, in some examples, computing system 100 may also include a network visualization engine 115, a memory 116, and a display 118.

[0015] Display 118 may be embedded into computing system 100 or communicatively coupled to computing system 100, and may be implemented using any suitable technology, such as LCD, LED, OLED, TFT, Plasma, etc. In some examples, display 118 may be a touch-sensitive display. Memory 116 may also be embedded in computing system 100 or communicatively coupled thereto, and may include any type of volatile or non-volatile memory, such as a random-access memory (RAM), flash memory, hard drive, memristor-based memory, and so forth. Engines 112, 113, 114, and 115 may each generally represent any combination of hardware and programming. Each of the engines is discussed in more detail below.

[0016] In some examples, term extraction engine 112 may obtain a text, where “text” may refer to any number of documents or portions thereof, any number of articles or portions thereof, any number of books (e.g., fiction or non-fiction books, textbooks, etc.) or portions thereof, or any other type of textual content. The text may be obtained from any type of machine-readable source, such as a file, a web page, an image of text, a data stream comprising text, etc. In some examples, the text may be obtained from a memory (e.g., 116) on computing system 100, or receive it from a remote device, e.g., via a local-area network, a wide-area network (e.g., the Internet), or any combination of these or other types of networks.

[0017] In some examples, after obtaining the text, engine 112 may pre-process the text which may include removing non-text content, removing stop words (e.g., prepositions, pronouns, etc.), stemming the words, or otherwise formatting and/or filtering the text. After optionally pre-processing the text, engine 112 may extract from the text a plurality of terms, i.e., two or more terms. In some examples, each extracted term may be either a single word (e.g., “software”) or a sequence of two or more words (e.g., “operating system”). For brevity, terms consisting of more than one

word may be referred to herein as “multi-word terms” and terms consisting of only one word may be referred to as “single-word terms.”

[0018] In some examples, engine 112 may extract from the text only labeled terms, i.e., only terms that have been at some point labeled by people as established concepts, terms of art, names of categories, etc. For example, engine 112 may extract from the text only terms that appear in a predefined set of taxonomies, indexes, encyclopedias, or other types of databases storing terms pre-labeled by people as concepts. For example, terms having dedicated Wikipedia articles may be considered by engine 112 as labeled terms. Thus, in some examples, engine 112 may disregard any unlabeled terms in the obtained text.

[0019] Alternatively or in addition, in some examples, engine 112 may disregard any single-word terms and extract only multi-word terms (e.g., labeled multi-word terms). Because a meaning of a multi-word term is less likely than a single-word term to change across various contexts, using only multi-word terms or as many multi-word terms as possible may help reduce any discrepancies associated with the same term meaning different things in different contexts.

[0020] In some examples, engine 112 may be configured to extract from the text at least M terms, where M may be a predefined number such as 10, 30, etc. In these examples, engine 112 may determine whether the text includes at least M labeled multi-word terms, and if so, it may not extract any additional terms. However, if the number of labeled multi-word terms in the text is less than M, engine 112 may extract additional terms.

[0021] To extract additional terms, engine 112 may first divide the text into a plurality of segments of fixed or variable sizes. For example, engine 112 may divide the text into segments such that each segment contains a certain number (e.g., 1, 10, etc.) of paragraphs, pages, sections, chapters, topics, or other units of text. After dividing the text into segments, engine 112 may apply a topic model such as a Latent Dirichlet Allocation (LDA) model on all segments of the text. In response, the topic model may generate a set of topics, where each topic may be described by a set of

terms referred to herein as “topic terms.” The topic model may also provide, for each topic term, a weight associated with that topic term.

[0022] In some examples, engine 112 may rank all topic terms by weight, and supplement the extracted labeled multi-word terms with highest ranking topic terms until the total number of obtained terms reaches M or until no topic terms remain. In some examples, some of the topic terms generated by the topic model may also be labeled terms, for example, labeled single-word terms. Such topic terms may be referred to herein as “labeled topic terms.” In these examples, the topic model may first supplement the extracted labeled multi-word terms with the highest ranking labeled topic terms. In some examples, if all labeled topic terms are insufficient to bring the total number of obtained terms to M, engine 112 may also add highest ranking unlabeled topic terms until the total number of terms reaches M, or until no topic terms remain.

[0023] After extracting the terms, term extraction engine 112 may pass the extracted terms to network construction engine 113. Engine 113 may obtain the extracted terms from engine 112, and construct, based on the extracted terms, a network of terms. In some examples, engine 113 may first determine which terms are found in which segments of the text. As discussed above, the text may be divided (e.g., virtually) by engine 112 or any other engine of computing system 100 into a plurality of segments, where each segment may correspond to one or more paragraphs, pages, sections, chapters, topics, or other units of text. Some extracted terms may be found in one segment, and some extracted terms may be found in a plurality of segments.

[0024] After determining which extracted terms are found in which segments, engine 113 may construct a network of terms by adding each term as a node and connecting any pair of terms found together in at least one segment in the text. The connections between the terms may also be referred to herein as “edges.” To illustrate, FIG. 3A shows an example arrangement 300 of five terms t1-t5 into four segments u1-u4. In this example, term t1 is found only in segment u1; term t2 is found in segments u1 and u2; term t3 is found in segments u1 and u2; term t4 is found in segments u1

and u3; and term t5 is found in segments u3 and u4. FIG. 3B illustrates an example network 310 constructed based on arrangement 300 of FIG. 3A. In network 310, any two terms found in the same segment are connected to each other. It is appreciated that a pair of terms may be found together in more than one segment. For example, terms t2 and t3 are found together in segment u1 and also in segment u2.

[0025] In some examples, each segment may be associated with a weight. A segment's weight may reflect, for example, a total number of terms extracted from that segment. For example, a segment's weight may be inversely proportional to the number of terms extracted from that segment. For example, FIG. 4A illustrates the example arrangement 300 of FIG. 3A with weights w1-w4 assigned to segments u1-u4, respectively. In this example, each segment is assigned a weight that is a reciprocal of the number of terms extracted from that segment. Accordingly, segment u1 is assigned a weight of $w1=0.25$ because segment u1 includes four extracted terms t1-t4; segment u2 is assigned a weight of $w2=0.5$ because segment u2 includes two extracted terms t2 and t3; segment u3 is assigned a weight of $w3=0.5$ because segment u3 includes two extracted terms t4 and t5; and segment u4 is assigned a weight of $w4=1$ because segment u4 includes only one extracted term t5.

[0026] In some examples, engine 113 may assign a weight to each edge in the network. The weight of an edge may be associated with (e.g., be a sum of) weights of the segment(s) that include both terms connected by the edge. To illustrate, FIG. 4B shows network 310 with weights assigned to all of the network's edges. In this example, the edge between terms t1 and t2 is assigned a weight of 0.25 because terms t1 and t2 appear together only in segment u1 whose weight is $w1=0.25$, and the edge between terms t2 and t3 is assigned a weight of 0.75 (i.e., the sum of $w1$ and $w2$) because terms t2 and t3 appear together in segments u1 and u2. Accordingly, if segments having many extracted terms weigh less, an edge between two terms found in a segment with few other terms may tend to have a higher weight, reflecting a potentially stronger correlation between the terms in such segments.

[0027] After network constructions engine 113 constructs the network of terms, term ranking engine 114 may score and rank the network's terms. In some examples, a score of a term may be calculated based on the term's centrality within the network. The term's centrality within the network may be determined, for example, based on the sum of the weights of the edges connecting the term to its adjacent terms. For example, a centrality of the term t_i may be calculated by engine 113 based on the following formula:

$$[0028] \quad \text{Centrality}(t_i) = \frac{1-d}{N} + d \sum_j \left\{ \frac{w(t_j, t_i)}{\sum_k w(t_j, t_k)} \right\}$$

where d is a predefined constant (e.g., 0.85), N is the total number of terms in the network, index j runs on all terms adjacent to the term t_i , $w(t_j, t_i)$ is the weight from the edge t_j to the edge t_i , and $w(t_j, t_k)$ is the weight from the edge t_j to the edge t_k . In some examples, the centrality of the term t_i may also depend on centralities of its adjacent terms. For example, engine 113 may calculate the term's centrality using the following formula:

$$[0029] \quad \text{Centrality}(t_i) = \frac{1-d}{N} + d \sum_j \left\{ \frac{w(t_j, t_i)}{\sum_k w(t_j, t_k)} \cdot \text{Centrality}(t_j) \right\}$$

[0030] Alternatively or in addition to being dependent on the term's centrality, the term's ranking may in some examples depend on how frequently (or infrequently) the term appears in a corpus of texts. In some examples, engine 113 may determine such frequency by calculating the inverse document frequency (IDF) of a term t using, for example, the following formula:

$$[0031] \quad \text{IDF}(t) = \log\left(\frac{n - n_t + 0.5}{n_t + 0.5}\right)$$

where n is the total number of documents in the corpus and n_t is the number of documents in the corpus that contain the term t . In some examples, the corpus of texts may include only texts from the same type or category as the text being processed. For example, engine 114 may select a corpus from a plurality of corpuses, such that the selected corpus matches or is associated with the text. For

example, if the text being processed is a textbook, term ranking engine 114 may select from a plurality of corpuses a corpus containing only or mostly textbooks.

[0032] As discussed above, in some examples engine 114 may score and rank each term based on the term's centrality, the term's inverse document frequency, or both. In some examples, engine 114 may calculate a score R of each term t based on a linear combination of the term's centrality and the term's inverse document frequency, for example, using the following formula:

$$[0033] \quad R(t) = \alpha \cdot N \cdot \text{Centrality}(t) + \beta \cdot \text{IDF}(t)$$

where N is the number of the extracted terms, and α and β are two predetermined and calibrated constants. After calculating scores for all the extracted terms, engine 114 may rank all terms based on their score. For example, higher scoring terms may be ranked higher, and vice versa.

[0034] In some examples, instead of extracting a fixed number of terms M , engine 112 may extract a larger number of terms, engine 113 may construct a network from the larger number of terms, and engine 114 may calculate scores for those terms. Engine 114 may then select all terms whose scores are above a predefined threshold. Next, engine 113 may construct a new network of selected terms only, after which engine 114 may recalculate the scores of the selected terms (whose centrality values may have changed) based on the new network.

[0035] After scoring and ranking the terms, engine 114 may, for example, store the terms along with their scoring, ranking, and any other information, in a memory (e.g., memory 116), and/or send it to another (e.g., remote) device. In some examples, engine 114 may only store and/or send information about a predefined number of top-ranked (and top-scoring) terms while discarding the other terms.

[0036] In some examples, instead of or in addition to storing the terms, engine 114 may represent the ranked terms (or only a predefined number of top-ranked terms) on a display. For example, engine 114 may send the term information to

network visualization engine 115, which may process the term information and display the information on a display (e.g., display 118).

[0037] Network visualization engine 115 may represent the terms on a display, for example, in a form of a graph that may be similar to the network constructed by construction engine 113. Engine 115 may represent each term on the display using text and/or a predefined shape (e.g., a circle) of a certain size and color. In some examples, the size of the shape representing a particular term may be associated with (e.g., be proportional to) the term's score and/or rank. This may cause higher scoring and higher ranking terms appear larger than lower scoring and lower ranking terms. Thus, engine 115 may provide a visual indication of each term's relative score or rank, that is, a visual indication of the term's score or rank relative to other terms' scores or ranks. Alternatively or in addition, in some examples, engine 115 may provide a visual indication of the term's actual (absolute) score or rank, e.g., in a numerical form.

[0038] In some examples, engine 115 may also receive a user input associated with a particular term on the display, and in response to the user input, display to the user information associated with the particular term. For example, the user may click or touch on or around the displayed term, and engine 115 may, in response to the click or touch, display one or more portions of the text that contain the term. A portion may include, for example, any number of sentences, paragraphs, pages, sections, or other units of text containing the term. In some examples, to display a portion of text, engine 115 may open and display a document containing the text by launching an appropriate application, such as a word processor, a PDF viewer, a web browser, etc. In some examples, engine 115 may also present the text such that at least one portion containing the term is visible. In some examples, engine 115 may also select (e.g., highlight) the term within the visible portion.

[0039] In some examples, engine 115 may also display the connections between the terms, for example, using straight lines. In some examples, engine 115 may also visually indicate the connections' relative or absolute weights, because, as discussed

above, a connection's weight may reflect the level of correlation between the two connected terms. In some examples, engine 115 may represent each connection as with a straight line whose length, thickness, color, or another parameter may represent the connection's weight. For example, higher weighing connections may be represented by thicker lines, by shorter lines, by lines of a particular color, and so forth.

[0040] FIG. 5 illustrates some of the examples discussed above. In the example of FIG. 5, network visualization engine 115 provides to display 118 a graph 510 representing an example network constructed by construction engine 113. In this example, the processed text is a textbook about operating systems, and the graph contains various terms extracted from the text and their connections to each other. In this example, the larger circles may correspond to higher ranked terms (e.g., "operating system," "virtual memory," and "CPU") and the thicker connection lines may indicate stronger correlations (e.g., a very strong correlation between the terms "operating system" and "virtual memory," and a strong correlation between the terms "virtual memory" and "caching"). As also illustrated in the example of FIG. 5, the user may click on a term (e.g., "context switch"), and network visualization engine 115 may open, in response to the click, a window 510 displaying the portion of the textbook containing the selected term. Engine 115 may also highlight every occurrence of the selected term in the text.

[0041] In the foregoing discussion, engines 112, 113, 114, and 115 were described as any combinations of hardware and programming. Such components may be implemented in a number of fashions. The programming may be processor executable instructions stored on a tangible, non-transitory computer-readable medium and the hardware may include a processing resource for executing those instructions. The processing resource, for example, may include one or multiple processors (e.g., central processing units (CPUs), semiconductor-based microprocessors, graphics processing units (GPUs), field-programmable gate arrays (FPGAs) configured to retrieve and execute instructions, or other electronic circuitry),

which may be integrated in a single device or distributed across devices. The computer-readable medium can be said to store program instructions that when executed by the processor resource implement the functionality of the respective component. The computer-readable medium may be integrated in the same device as the processor resource or it may be separate but accessible to that device and the processor resource. In one example, the program instructions can be part of an installation package that when installed can be executed by the processor resource to implement the corresponding component. In this case, the computer-readable medium may be a portable medium such as a CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed, and the computer-readable medium may include integrated memory such as a hard drive, solid state drive, or the like.

[0042] FIG. 6 is a flowchart of an example method 600. Method 600 may be described below as being executed or performed by a system or by a computing device such as computing system 100 of FIG. 1. Other suitable systems and/or computing devices may be used as well. Method 600 may be implemented in the form of executable instructions stored on at least one non-transitory machine-readable storage medium of the system and executed by at least one processor of the system. Alternatively or in addition, method 600 may be implemented in the form of electronic circuitry (e.g., hardware). In alternate examples of the present disclosure, one or more blocks of method 600 may be executed substantially concurrently or in a different order than shown in FIG. 6. In alternate examples of the present disclosure, method 600 may include more or less blocks than are shown in FIG. 6. In some examples, one or more of the blocks of method 600 may, at certain times, be ongoing and/or may repeat.

[0043] At block 605, method 600 may extract a term from a segment of a text, where the text may include a plurality of segments. At block 610, the method may add the term to a network of terms by connecting the term to a set of other terms extracted from the segment. As discussed above, extracting the term from the

segment of the text may include, for example, determining whether the term is included in a database (or a list) of labeled terms.

[0044] At block 615, the method may determine the term's centrality within the network of terms. As discussed above, each of the term's connections to the set of other terms extracted from the segment may be associated with a weight, and determining the term's centrality within the network may include calculating a sum of the weights of the term's connections to the set of other terms extracted from the segment. The weight associated with each of the term's connections may be calculated based on a number of segments in the plurality of segments that include both terms connected by the connection and/or based on a number of other terms extracted from those segments.

[0045] At block 620, the method may determine the term's frequency within a corpus of texts. At block 625, the method may determine the term's score based on the term's centrality and the term's frequency. As discussed above, in some examples, the method may display the term on a display (e.g., 118) and providing a visual indication of the term's score on the display. As also discussed above, in some examples, the method may also receive a user input associated with the term, and responsive to the user input displaying a portion of the text, where the portion includes the term.

[0046] FIG. 7 is a block diagram of an example computing system 700. Computing device 700 may be similar to computing system 100 of FIG. 1. In the example of FIG. 7, computing device 700 includes a processor 710 and a non-transitory machine-readable storage medium 720. Although the following descriptions refer to a single processor and a single machine-readable storage medium, it is appreciated that multiple processors and multiple machine-readable storage mediums may be anticipated in other examples. In such other examples, the instructions may be distributed (e.g., stored) across multiple machine-readable storage mediums and the instructions may be distributed (e.g., executed by) across multiple processors.

[0047] Processor 710 may be one or more central processing units (CPUs), microprocessors, and/or other hardware devices suitable for retrieval and execution of instructions stored in non-transitory machine-readable storage medium 420. In the particular example shown in FIG. 7, processor 710 may fetch, decode, and execute instructions 722, 724, 726, 728, or any other instructions (not shown for brevity). As an alternative or in addition to retrieving and executing instructions, processor 710 may include one or more electronic circuits comprising a number of electronic components for performing the functionality of one or more of the instructions in machine-readable storage medium 720. With respect to the executable instruction representations (e.g., boxes) described and shown herein, it should be understood that part or all of the executable instructions and/or electronic circuits included within one box may, in alternate examples, be included in a different box shown in the figures or in a different box not shown.

[0048] Non-transitory machine-readable storage medium 720 may be any electronic, magnetic, optical, or other physical storage device that stores executable instructions. Thus, medium 720 may be, for example, Random Access Memory (RAM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a storage drive, an optical disc, and the like. Medium 720 may be disposed within computing device 700, as shown in FIG. 7. In this situation, the executable instructions may be “installed” on computing device 700. Alternatively, medium 720 may be a portable, external or remote storage medium, for example, that allows computing device 700 to download the instructions from the portable/external/remote storage medium. In this situation, the executable instructions may be part of an “installation package”. As described herein, medium 720 may be encoded with executable instructions.

[0049] Referring to FIG. 7, instructions 722, when executed by a processor (e.g., 710), may cause a computing device (e.g., 700) to extract a term from a segment of a text, the text comprising a plurality of segments. Instructions 724, when executed by the processor, may cause the computing device to connect the term to a set of other

terms extracted from the segment via a set of connections. Instructions 726, when executed by the processor, may cause the computing device to calculate a weight for each connection in the set of connections based on a number of other terms extracted from the segment. Instructions 728, when executed by the processor, may cause the computing device to determine a score of the term based on the weights of the set of connections and based on a frequency at which the term appears in a corpus of texts. As discussed above, in some examples, additional instructions (not shown for brevity) may cause the computing device to display the term and a visual indication of the term's score.

CLAIMS

1. A method comprising:
 - extracting a term from a segment of a text, the text comprising a plurality of segments;
 - adding the term to a network of terms by connecting the term to a set of other terms extracted from the segment;
 - determining the term's centrality within the network of terms;
 - determining the term's frequency within a corpus of texts; and
 - based on the term's centrality and the term's frequency, determining a score of the term.
2. The method of claim 1, further comprising:
 - displaying the term on a display; and
 - providing a visual indication of the term's score on the display.
3. The method of claim 2, further comprising receiving a user input associated with the term, and responsive to the user input displaying a portion of the text, where the portion includes the term.
4. The method of claim 1, wherein extracting the term from the segment of the text comprises determining whether the term is included in a database of labeled terms.
5. The method of claim 1, wherein each of the term's connections to the set of other terms extracted from the segment is associated with a weight, and wherein determining the term's centrality within the network comprises calculating a sum of

the weights of the term's connections to the set of other terms extracted from the segment.

6. The method of claim 5, wherein the weight associated with each of the term's connections is calculated based on a number of segments in the plurality of segments that comprise both terms connected by the connection.

7. The method of claim 6, wherein the weight associated with each of the term's connection is calculated also based on a number of other terms extracted from the number of segments in the plurality of segments that comprise both terms connected by the connection.

8. A computing system comprising:

a term extraction engine to:

extract a plurality of terms from a text, wherein the text comprises a plurality of segments;

a network construction engine to:

connect each term in the plurality of terms to each term included in any segment, in the plurality of segments, in which the term is also included; and

a term ranking engine to:

calculate a score for each term in the plurality of terms based on the term's connections to other terms in the plurality of terms.

9. The computing system of claim 8, wherein the term ranking engine is to calculate the score for each term in the plurality of terms further based on an inverse document frequency associated with the term.

10. The computing system of claim 8, wherein the network construction engine is further to assign weights to all connections between the terms in the plurality of

terms, and wherein the term ranking engine is to calculate the score for each term in the plurality of terms based on weights of the term's connections.

11. The computing system of claim 10, wherein the network construction engine is to assign a weight to each connection based on a number of segments in which terms connected by the connection are included together and based on a number of other terms included in the number of segments.

12. The computing system of claim 8, further comprising:
a display; and
a network visualization engine to:
represent a set of terms from the plurality of terms on the display, and
represent scores of the set of terms on the display.

13. The computing system of claim 8, wherein extracting the plurality of terms comprises:
extracting from the text a set of multi-word terms included in a database of labeled terms; and
based on a determination that a size of the set of multi-word terms is less than a predefined size, extracting from the text a set of additional terms using a topic model.

14. A non-transitory machine-readable storage medium encoded with instructions executable by a processor of a computing device to cause the computing device to:
extract a term from a segment of a text, the text comprising a plurality of segments;
connect the term to a set of other terms extracted from the segment via a set of connections;
calculate a weight for each connection in the set of connections based on a number of other terms extracted from the segment; and

determine a score of the term based on the weights of the set of connections and based on a frequency at which the term appears in a corpus of texts.

15. The non-transitory machine-readable storage medium of claim 14, wherein the instructions further cause the computing device to display the term and a visual indication of the term's score.

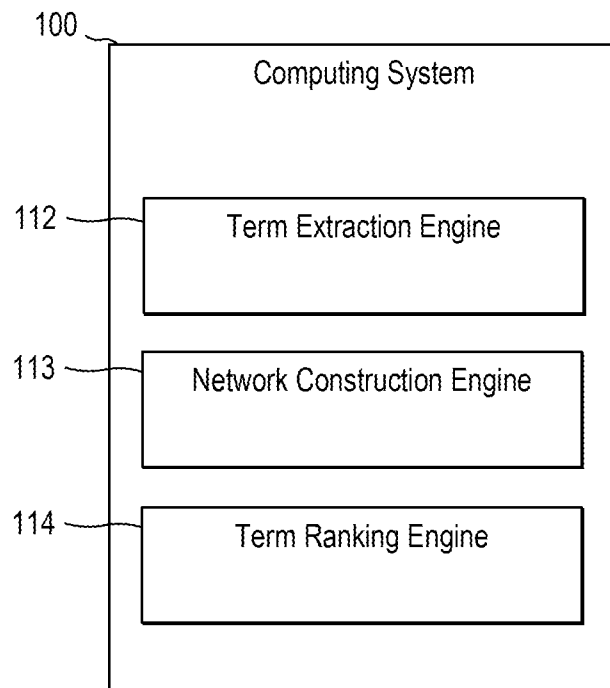
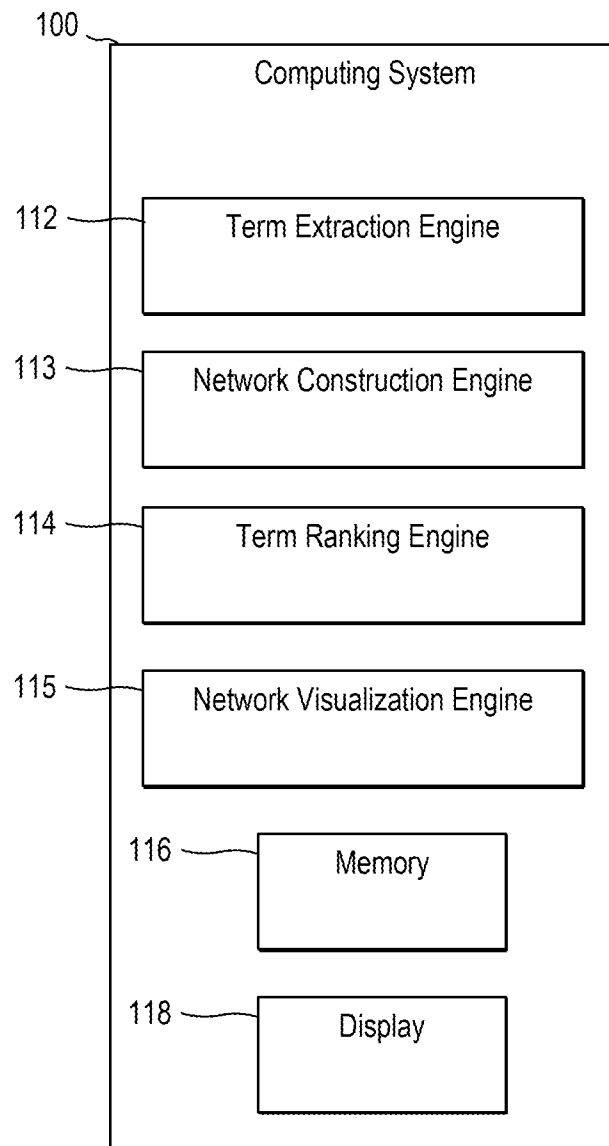


FIG. 1

2/7

**FIG. 2**

3/7

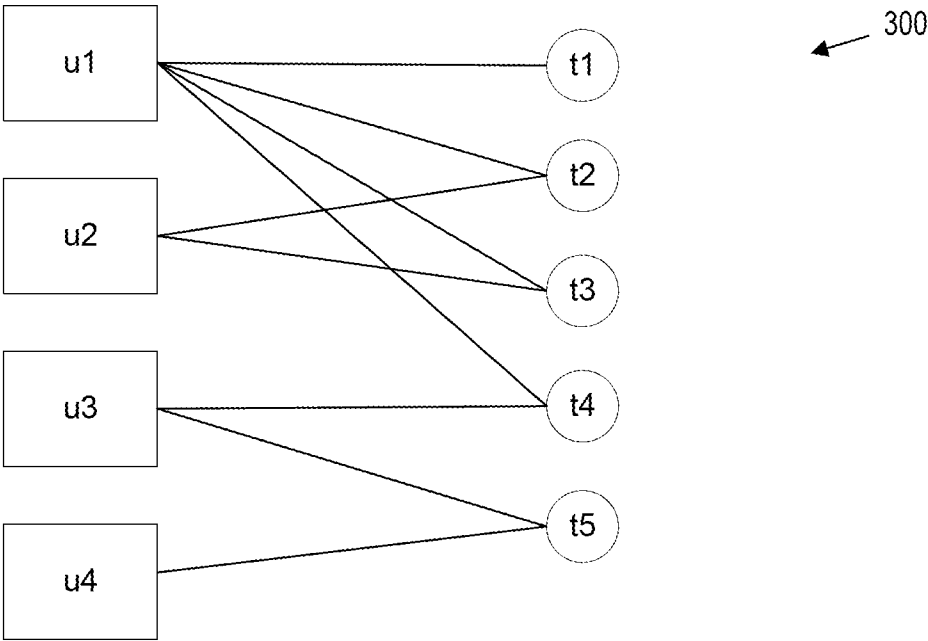


FIG. 3A

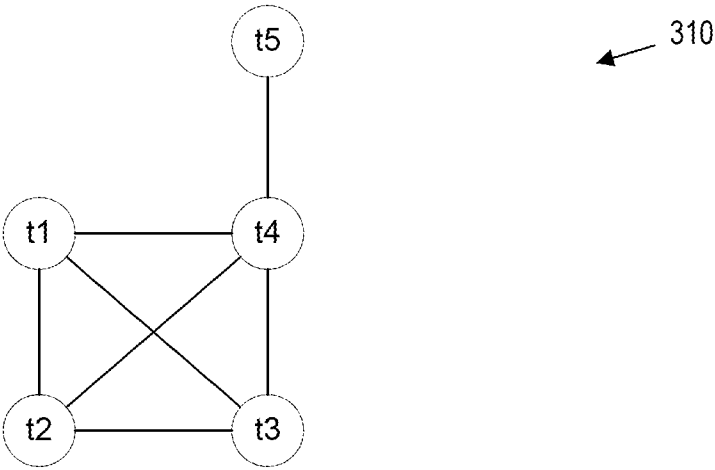


FIG. 3B

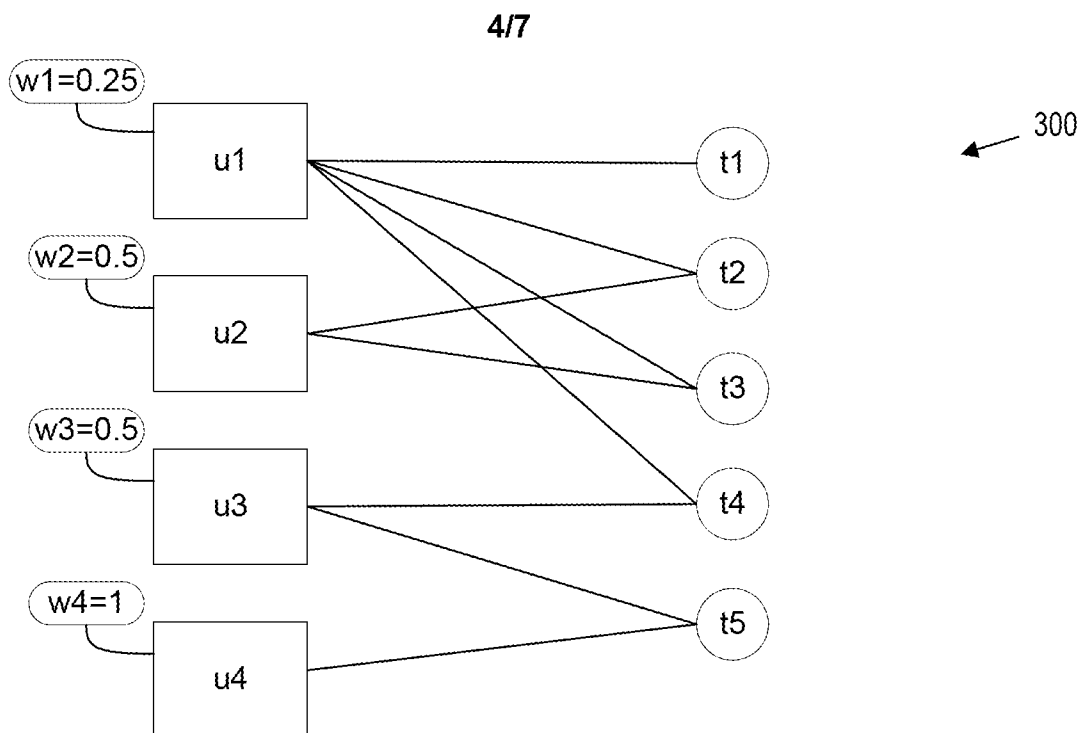


FIG. 4A

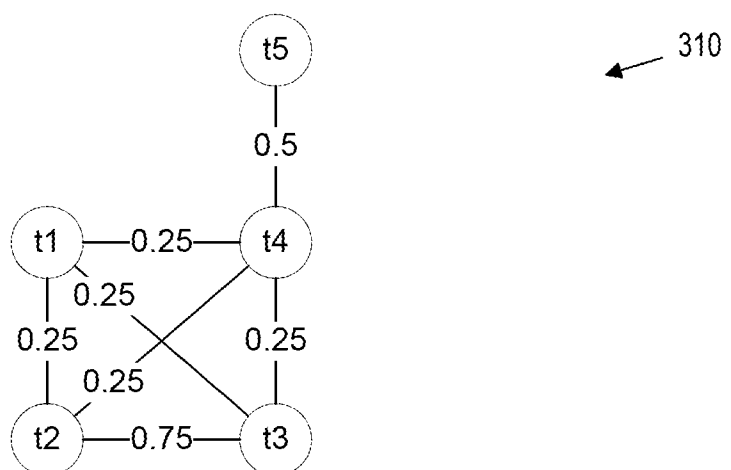
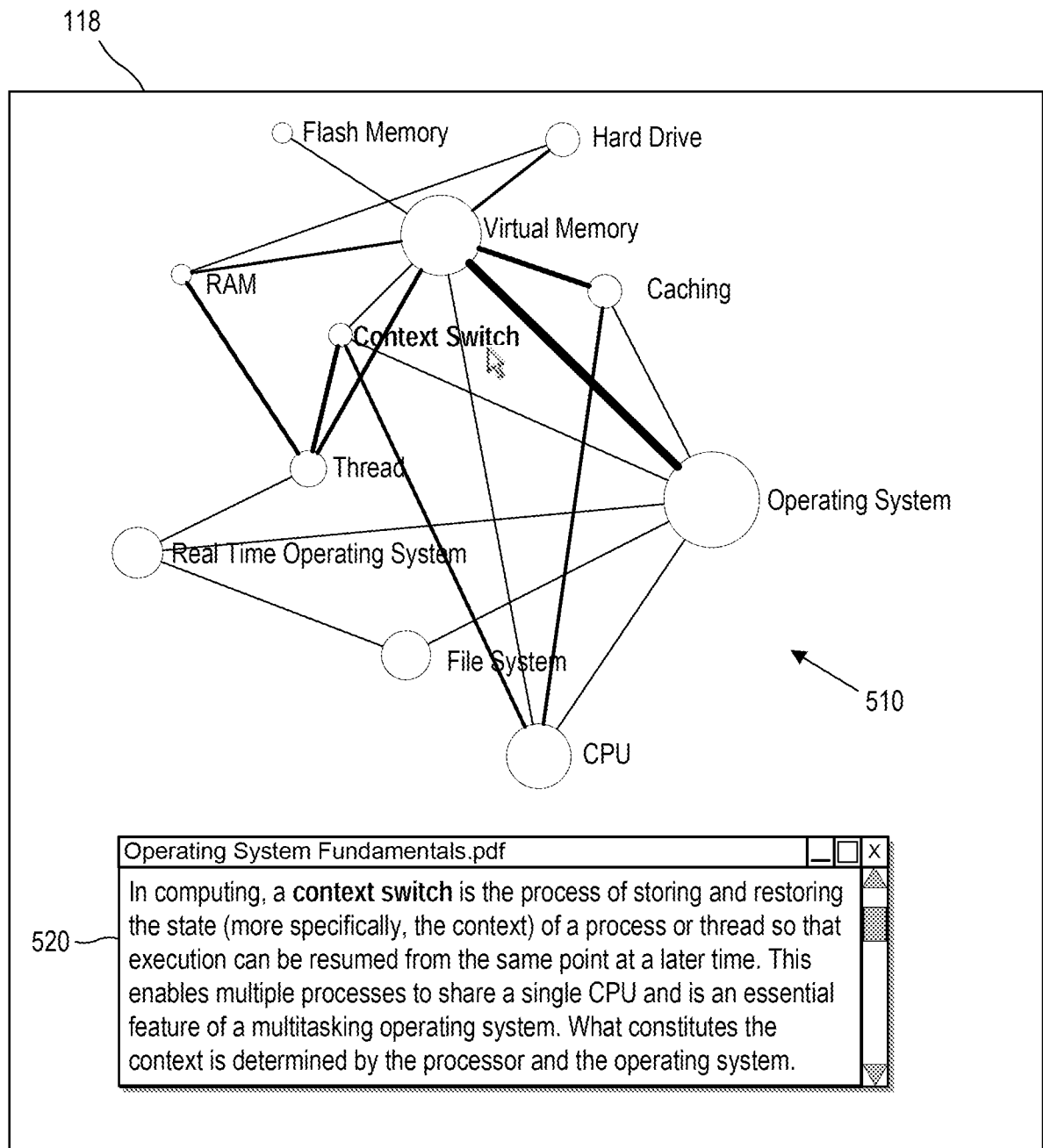
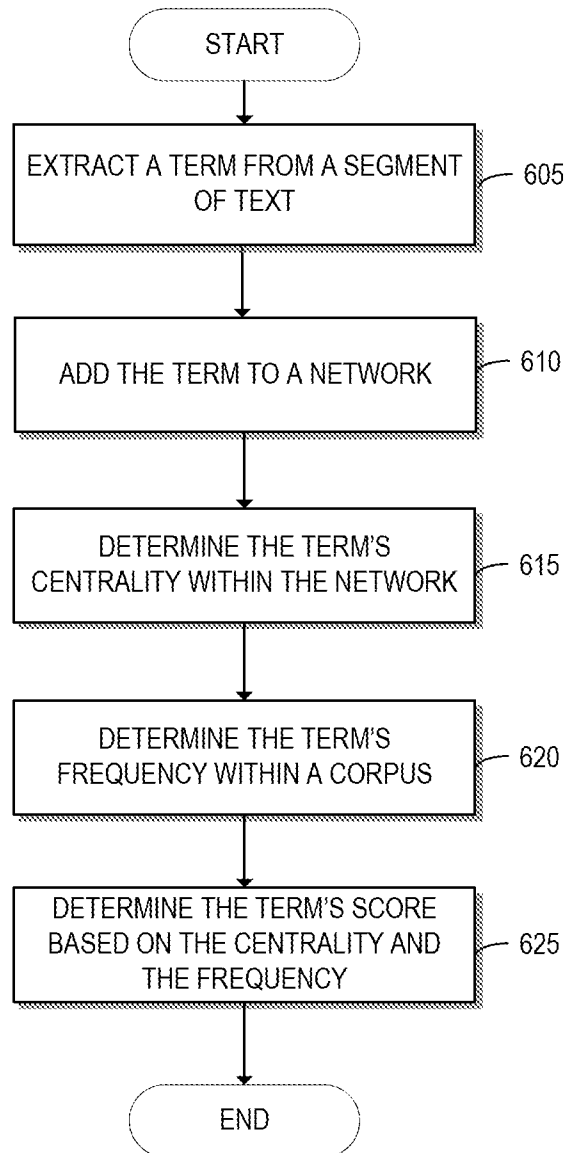


FIG. 4B

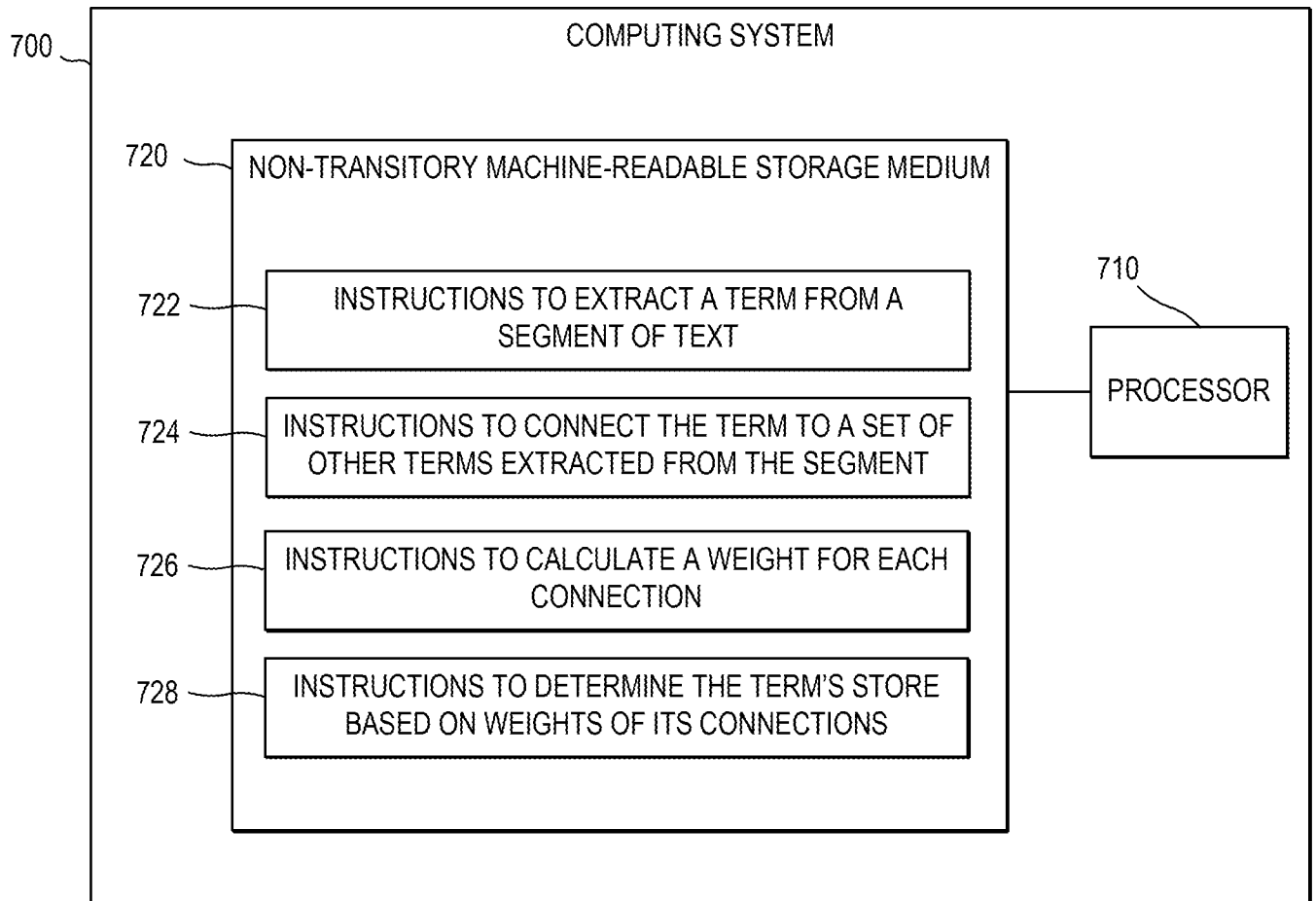
5/7

**FIG. 5**

6/7

600**FIG. 6**

7/7

**FIG. 7**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/034590**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/27(2006.01)i, G06F 17/30(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 17/27; G06F 3/14; G06F 17/30; G06F 15/18

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: segment, graph, centrality, TF(Term Frequency), IDF(Inverse Document Frequency), weight, 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.
X	US 2010-0145678 A1 (ANDRAS CSOMAI et al.) 10 June 2010 See paragraphs [0002], [0034], [0059]-[0061], [0064], [0078], and [0101]-[0104]; claims 33 and 35; and figures 1, 4-5, and 9.	1,4-11,13-14
Y		2-3,12,15
Y	US 2011-0060983 A1 (WEI JIA CAI et al.) 10 March 2011 See paragraphs [0020]-[0021], [0078], and [0081]; and figures 2, 4, and 7.	2-3,12,15
A	US 2011-0270845 A1 (CHING-YUNG LIN et al.) 03 November 2011 See paragraphs [0030]-[0031] and [0054]-[0058]; and figures 2 and 4.	1-15
A	WO 2006-001906 A2 (UNIVERSITY OF NORTH TEXAS) 05 January 2006 See page 24, line 12 - page 25, line 18; and figures 5-8.	1-15
A	WILLIYAN D. ABILHOA et al., "A keyword extraction method from twitter messages represented as graphs," Applied Mathematics and Computation, Vol. 240, pp. 308-325, 2014 See page 309, line 25 - page 311, line 11; and figure 1.	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

25 February 2016 (25.02.2016)

Date of mailing of the international search report

26 February 2016 (26.02.2016)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-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/US2015/034590

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0145678 A1	10/06/2010	US 8346534 B2	01/01/2013
US 2011-0060983 A1	10/03/2011	US 8245135 B2	14/08/2012
US 2011-0270845 A1	03/11/2011	None	
WO 2006-001906 A2	05/01/2006	US 2005-0278325 A1	15/12/2005
		US 7809548 B2	05/10/2010
		WO 2006-001906 A3	08/09/2006