



(51) International Patent Classification:
G06Q 30/06 (2012.01)

(21) International Application Number:
PCT/US2015/013714

(22) International Filing Date:
30 January 2015 (30.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).

(72) Inventor: **KOUTRIKA, Georgia**; 1501 Page Mill Road, Palo Alto, California 94304-1100 (US).

(74) Agents: **HEWLETT-PACKARD COMPANY** et al.; Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 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).

[Continued on next page]

(54) Title: PRODUCT, OPERATING SYSTEM AND TOPIC BASED RECOMMENDATIONS

300

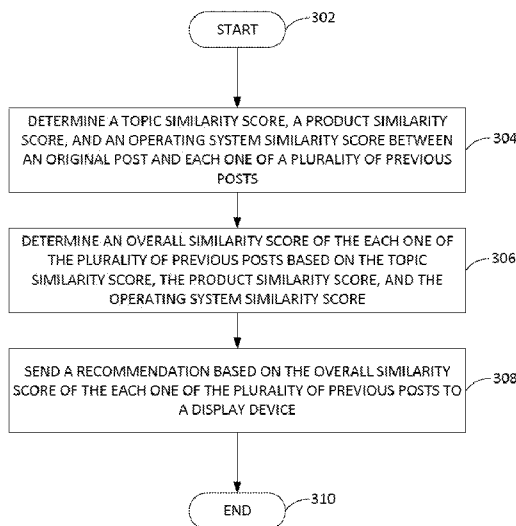


FIG. 3

(57) Abstract: A method is described in which a topic similarity score, a product similarity score and an operating system similarity score between an original post and each one of a plurality of previous posts are determined; an overall similarity score of the each one of the plurality of previous posts based on the topic similarity score, the product similarity score and the operating system similarity score is determined; and a recommendation of a top K number of the plurality of previous posts based on the overall similarity score of the each one of the plurality of previous posts is sent to a display device.

**Declarations under Rule 4.17:**

- *as to the identity of the inventor (Rule 4.17(i))*
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

PRODUCT, OPERATING SYSTEM AND TOPIC BASED RECOMMENDATIONS

BACKGROUND

[0001] Online product discussion services comprise a communication channel with and among customers on a company's products. In these sites, the customers ask questions about a company's product to seek solutions to problems with products. Various people can answer the questions creating a valuable repository of information about the company's products. Unfortunately, when a customer visits the forum, finding the right information may be tedious and often an unsuccessful process.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] FIG. 1 is a block diagram of an example system of the present disclosure;

[0003] FIG. 2 is an example of an original post and recommendations based on the original post;

[0004] FIG. 3 is an example flowchart of a method for providing product aware, operating system aware and topic based recommendations;

[0005] FIG. 4 is another example flowchart of a method for providing product aware, operating system aware and topic based recommendations; and

[0006] FIG. 5 is an example high-level block diagram of a computer suitable for use in performing the functions described herein.

DETAILED DESCRIPTION

[0007] The present disclosure broadly discloses a method and non-transitory computer-readable medium for providing product aware, operating system aware and topic based recommendations. As discussed above, finding the right information within an online product discussion service for a company's product can be a tedious and often an unsuccessful process for the customer. Basic keyword searching does not help because people can describe a problem or refer to a product in different ways. As a result, locating and bringing together

the relevant information within a few clicks in these forums can be difficult or the amount of search results based on a basic keyword search can be overwhelming for the customer to manually sift through them.

[0008] Examples of the present disclosure provide a novel method for providing product aware, operating system aware and topic based recommendations. For example, based on a question posted by a customer to an online product forum, or discussion service, the customer's question can be analyzed across several dimensions to provide high value recommendations to the customer. In one example, the question can be analyzed along the dimensions of topic, products and operating system. Based on the identified topic, product and operating system of the customer's environment, the examples of the present disclosure can provide a more accurate recommendation of previous posts within the online product forum service that answers the customer's question.

[0009] FIG. 1 illustrates an example system 100 of the present disclosure. In one implementation, the system 100 includes a communication network 102 in communication with one or more endpoint devices 124 and 126. It should be noted that although two endpoint devices 124 and 126 are illustrated in FIG. 1 that any number of endpoint devices may be deployed. In one example, the endpoint devices 124 and 126 may be any type of endpoint device, including for example, a desktop computer, a laptop computer, a mobile telephone, a smart phone, a tablet computer, and the like.

[0010] In one example, customers may use their respective endpoint device 124 or 126 to post questions about a company's product or post answers to questions posted about the company's product. The endpoint devices 124 and 126 may be in communication with the communication network 102 that includes an application server (AS) 104 and a database (DB) 106. It should be noted that the communication network 102 has been simplified for ease of explaining examples of the present disclosure. The communication network 102 may include additional network elements (not shown) such as a border element, gateways, firewalls, additional access networks, and the like.

[0011] In one example, previous posts 108 containing answers to previously

posted questions are stored in the DB 106. In addition, the DB 106 may store one or more models 128 used for identifying topics, products and operating systems (OS) contained in an original post 110 from a customer.

[0012] In one example, hand coded rules or models learnt from annotated data may be used to create a product recognition model and an OS recognition model for product recognition and OS recognition, respectively. In one example, the product recognition model and the OS recognition model may be referred to collectively as the product recognition model as the recognition of the OS may be a special case of product recognition. For example, annotated data may be a training set of posts that have been identified with positive and negative matches of products or OS that are marked. Then a classification model can be trained to identify products or OS in the original post 110.

[0013] In one implementation, a Latent Dirichlet Allocation (LDA) with Gibbs sampling may be used to create a model for topic recognition. For example, a number of s topics to be generated are given as an input to the algorithm based on a training document set. A small number of topics could provide a broad overview of the document structure, whereas a large number of topics could provide fine-grained topics at the cost of additional computational time.

[0014] In one implementation, the AS 104 and the DB 106 may be operated and maintained by a company that produces one or more products that the customers have questions about. In one example, the AS 104 may be deployed as a computer that includes a processor and is modified to perform the dedicated functions described herein. For example, the processor may execute the instructions and algorithms provided by the modules within the AS 104 as described below.

[0015] In one example, the AS 104 may receive the original post 110 from the endpoint device 124 or 126 of the customers. The AS 104 may process the original post 110 to provide one or more recommendations 122 back to the endpoint device 124 or 126 that submitted the original post 110. In one example, the recommendations 122 may include one or more of the previous posts 108 based on an overall similarity score that is determined or calculated by a processor (e.g., the processor 302 described below and illustrated in FIG.

3), as discussed in further detail below. In one example, the recommendations 122 may include the top k recommendations based on the top k previous posts 108 that have the highest overall similarity scores.

[0016] In one example, to determine the overall similarity score, a product recognition module 112 may analyze the original post 110. The product recognition module 112 may apply the product recognition model and the OS recognition model 128 stored in the DB 106 to identify a product and an OS contained in the original post 110. The identified product may be received by a product similarity module 116 and the identified OS may be received by an OS similarity module 118. A topic similarity module 114 may receive the original post 110.

[0017] In one example, the topic similarity module 114 may include instructions for the processor to determine a topic similarity score between the original post 110 and the previous posts 108 stored in the DB 106. In one example, using latent topic models, a thread d_i can be represented by a set of estimated latent topic probabilities $P(z_j|d_i)$ over the set of different topics z_j in Z . Given a collection D of n threads and s topics, one can denote $F_{n \times s}$ the document-topic matrix that captures how the s topics are assigned to the n threads in D . That is $F_{ij} = P(z_j|d_i)$ with $i \leq n$ and $j \leq s$. Given a set of estimated latent topic probabilities for each thread, there are several measures that can be used to compare the similarity of two threads such as the cosine similarity, the unnormalized dot product and the symmetric Kullback-Leibler divergence measure. For example, a processor may communicate with the DB 106 to retrieve and execute a function to calculate the cosine similarity of two threads based on their topics. One example of a cosine similarity function that can be executed by the processor may be shown below:

$$\text{cossim}(d_i, d_j) = \frac{\vec{F}_i \cdot \vec{F}_j}{\|\vec{F}_i\| \|\vec{F}_j\|}.$$

[0018] In one example, a bagging model may be applied to the topic modeling used to determine the topic similarity score to provide a more accurate

topic similarity score. One problem with general topic modeling algorithms is that these algorithms only guarantee to converge to a locally optimal maximum likelihood solution and not the globally optimal solution. Thus, different initializations of a topic model will yield different final models. Consequently, document relationships will vary among these models. To tackle this problem and capture document relationships as accurately as possible, a bagging method can be used.

[0019] In one example, the bagging method runs the topic model a number of times over the input set of documents and evidence of the similarity of a pair of documents are combined from all the outputs. When using the bagging technique, the bagging method begins by assuming that N different latent topic models M_1 through M_N are trained from N different random model initializations. From each model M_k , an estimate of the topic distribution is produced for each thread. Once the topic models are generated, an aggregation step follows where for each pair of threads d_i and d_j the non-zero thread similarity measures between the threads produced by each of the N models are averaged. In one example of the present disclosure, a processor may communicate with the DB 106 to retrieve and execute a function to perform the bagging method. In one example, the bagging method uses an arithmetic mean as described by the function below:

$$sim_T(d_i, d_j) = \frac{1}{N'} \sum_{k=1}^N cossim(d_i, d_j, M_k)$$

where $N' \leq N$ is the number of non-zero topic similarity scores for the pair of documents d_i and d_j .

[0020] In one example, the product similarity module 116 may include instructions for the processor to determine a product similarity score between the product identified in the original post 110 and the products in the previous posts 108. The OS similarity module 118 may determine an OS similarity score between the OS identified in the original post 110 and the OS in the previous posts 108. In one example, the processor may determine the product similarity

score and the OS similarity score using the same algorithms.

[0021] Determining the product similarity score and the OS similarity score may not be straightforward. For example, a post may talk about a single product, but the author may refer to this product several times in the thread in different ways. In another example, the post may talk about more than one product (e.g., "We are currently running HP Officejet 6500 printers in our office and we also have one HP LaserJet 2605n."). In yet another example, authors in different posts may make different references to the same product.

[0022] In addition, different products within a same family may be mentioned in different posts. The product similarity score and the OS similarity score should account for similarity between the products in the same family or series. For example, all deskjet printers are more similar to each other than to laserjet printers.

[0023] To account for the factors described above, a processor may need to apply one or more algorithms to determine the product similarity score and the OS similarity score may involve applying one or more algorithms. In one example, two algorithms may be used to determine the product similarity score and the OS similarity score. An objective of the first algorithm is to find whether two strings refer to the same product, but vary slightly due to a typo, a different user writing style, or because the products are close "relatives" in a product family. In one example, a Levenshtein distance may be used as the first algorithm.

[0024] In one example, the Levenshtein distance is a string metric for measuring the difference between two strings. For example, the Levenshtein distance is the minimum number of single-character edits (e.g., insertions, deletions or substitutions) required to change one string into the other string.

[0025] In one example, the Levenshtein distance is modified to take into account a length of the two strings. For example, the smaller the two strings are, the fewer differences that are acceptable. Conversely, the longer the two strings are, the larger the amount of difference that is acceptable.

[0026] To illustrate by example, the string "officejet 6500" and "officejet 5500" should have a high product similarity score because there is enough information

to say that these products are quite similar. They are in the same family of “officejet” printers and only have a single character difference. On the other hand, the string “hp6500” and “hp5500” will have a lower score. Although the two strings “hp6500” and “hp5500” also only have a single character difference, the strings are shorter and contain less information. For example, the two strings do not contain enough information to determine if the products are within the same family. For example, “hp6500” could refer to a laserjet printer and “hp5500” could refer to a laptop computer.

[0027] In one example, given two product strings s_i and s_j with sizes or lengths $\text{len}(s_i)$ and $\text{len}(s_j)$ respectively, let $l = \min(\text{len}(s_i); \text{len}(s_j))$ and $\text{lev}(s_i; s_j)$ be their Levenshtein distance. In one example, a processor may communicate with the DB 106 to retrieve and execute a normalized Levenshtein distance to compute the similarity between the two product strings. An example of the normalized Levenshtein distance may be as follows:

$$\text{normlev}(s_i, s_j) = \begin{cases} \frac{l - \text{lev}(s_i, s_j)}{l + \text{lev}(s_i, s_j)} & \text{if } \text{lev}(s_i, s_j) > l/2 \\ \frac{l - \text{lev}(s_i, s_j)}{l} & \text{otherwise} \end{cases}.$$

[0028] The test $\text{lev}(s_i, s_j) > l/2$ determines when too many edits are required in order to transform one string to the other. This test may be used as a criterion, which has the meaning that if the number of changes would affect more than half of a string, then one should penalize the final score. This threshold works well for our product similarity computation problem. The threshold value may be set to a different value depending on the domain and the application. Using the example strings above and the normalized Levenshtein distance, the strings “hp6500” and “hp5500” would have a product similarity score of 0.83, while the strings “officejet 6500” and “officejet 5500” would have a product similarity score of 0.923.

[0029] As noted above, determining the product similarity score and the OS similarity score may involve applying one or more algorithms. The objective of a second algorithm is to identify when two strings that differ substantially actually

refer to the same product. For example, the strings “hp officejet 6310” and the string “6310” would have a low score based on the normalized Levenshtein distance function alone (e.g., 0.15). For example, the two strings are relatively short and have a large number of character differences between the two strings. However, the second algorithm would provide a higher score indicating that the two strings may refer to the same product because of the use of the same model number.

[0030] In one example, the second algorithm may be a Jaccard similarity function. In one example, the Jaccard similarity function may be right ordered (e.g., *r-ordjacc*). The Jaccard index measures the size of the intersection of the two sets divided by the size of their union. The ordered Jaccard similarity takes into account the position of elements in the two strings as well as the length of each element. There are two possibilities when considering the position of elements in two strings. The function may give more importance to larger elements in top positions that coincide resulting in a higher similarity score (e.g., this is called a left-ordered Jaccard). For example, applied to products, products of the same family with different model information would have a higher score using the Jaccard similarity function than products of different families with similar model information. Another possibility when considering the position of the elements of the two strings is to have the function give more importance to larger elements in bottom positions that coincide resulting in a higher similarity score (this is called the right-ordered Jaccard).

[0031] In one example, the left-ordered Jaccard index function is described as follows. A string s_i (e.g., a product description) can be represented as an ordered set S_i of tokens such that $S_i[k]$ is the k -th token in s_i , n_i is the number of tokens in s_i and hence the size of S_i . Furthermore, $len(S_i[k])$ is the length of the token $S_i[k]$. Given two strings s_i and s_j , their respective ordered sets are S_i and S_j with n_i and n_j being their corresponding sizes. Sets may have different sizes. Let $n = \max(n_i, n_j)$. To compute the left-ordered Jaccard similarity, one may ‘left-align’ the two sets by ‘padding’ the smaller set with empty tokens at its end. Then, the left-ordered Jaccard similarity returns the product similarity score giving priority to the left positions in the strings. In one example, the left-ordered

Jaccard index function may be stored in the DB 106 and retrieved and executed by a processor. In one example, the left-ordered Jaccard index function may be expressed as follows:

$$l - ordjacc(S_i, S_j) = \frac{\sum_{k=1, n} w_k * 1/k * (len(S_i|k|) + len(S_j|k|))}{\sum_{k=1, n} 1/k * (len(S_i|k|) + len(S_j|k|))}$$

where $w_k = 1$ if $S_i|k| == S_j|k|$ and $w_k = 0$ otherwise.

[0032] The left-ordered Jaccard index function can be also expressed in a form that gives priority to the last positions. In this case, the Jaccard similarity would right-align s_i and s_j and pad the smaller set at the left positions. The right-ordered Jaccard similarity is denoted $r-ordjacc$. Using the $r-ordjacc$ based on the left-ordered Jaccard index function described above for the example strings above, the similarity score of the “hp officejet 6310” and “6310” would only be 0.15 using *normlev* of normalized Levenshtein distance, but would be 0.42 with the $r-ordjacc$.

[0033] In one example, combining together the scores determined from Levenshtein distance and the left-ordered Jaccard index function, a processor may then compute the similarity of two products s_x and s_y by applying a combination of the functions as shown below:

$$sim(s_x, s_y) = max(normlev(s_x, s_y), r - ordjacc(s_x, s_y))$$

That is, the similarity of the two products is the maximum of the partial similarity scores of the products computed by the different algorithms. A different combining function could also be used, such as the minimum or average.

[0034] A processor may compute the similarity of two posts or threads d_i and d_j based on the lists P_i and P_j of products mentioned in the posts or threads by applying a function below:

$$sim_p(d_i, d_j) = max\{sim(s_x, s_y) | \forall s_x \in P_i, \forall s_y \in P_j\}$$

$$\text{where } \text{sim}(s_x, s_y) = \max(\text{normlev}(s_x, s_y), r - \text{ordjacc}(s_x, s_y)).$$

That is the similarity of the two posts is computed by considering the maximum similarity of the products mentioned in the two posts. A different combining function could be also used that computes the similarity of the two posts by considering for example the minimum or average similarity of their product lists. Similarly, a processor may compute the OS similarity score as $\text{sim}_{\text{OS}}(d_i, d_j)$ using the normalized Levenshtein distance function, the left or right ordered Jaccard index function and the combination of functions as described above.

[0035] After the topic similarity score, the product similarity score and the OS similarity score are each determined, each score will produce a different ranking list of recommendations for previous posts 108 with respect to the original post. However, the different ranking lists may not coincide with one another. For example, the original post 110 and a previous post 108 may have a high topic similarity score referring to the same topic, but may not refer to the same product and have a low product similarity score. In another example, the original post 110 and a previous post may have high OS similarity score, but may refer to different problems and have a low topic similarity score.

[0036] As a result, the topic similarity score, the product similarity score and the OS similarity score may be fed to a multi-aspect recommendation module 120. The multi-aspect recommendation module 120 may include instructions for the processor to determine the top k recommendations. In one example, the top k recommendations may be based on an overall similarity score determined by the multi-aspect recommendation module 120. In one example, the overall similarity score may be based on the topic similarity score, the product similarity score and the OS similarity score. In one example, the topic similarity score, the product similarity score and the OS similarity score can each be weighted and summed to obtain the overall similarity score. In one example, a processor may communicate with the DB 106 to retrieve and execute a function to calculate the overall similarity score. In one implementation, the processor may calculate the overall similarity score using a function as described below:

$$s = a_1 * sim_T + a_2 * sim_P + a_3 * sim_{OS},$$

where a_1 , a_2 and a_3 are different weights (e.g., a value between 0 and 1) for the topic similarity score sim_T , the product similarity score sim_P and the OS similarity score sim_{OS} .

[0037] In another example, the top k recommendations may be based on other score methods that use the topic similarity score, the product similarity score and the OS similarity score. For example, the top k recommendations may be based on the topic similarity score and then re-ordered based on a combination of the product similarity score and the OS similarity score.

[0038] Thus, after all of the scores s are determined for each one of the previous posts 108, the top k previous posts 108 can be filtered and selected as the top k recommendations. The top k recommendations may be sent to a display device of the customer that submitted the original post 110. For example, the display device may be part of the endpoint device 124 or 126 (e.g., a monitor). In one example, if the original post 110 included multiple products or operating systems, the overall similarity score may be determined for each product and OS identified in the original post 110.

[0039] In one example, the topic similarity scores, the product similarity scores, the operating system similarity scores and overall similarity scores may be pre-determined between the plurality of previous posts and stored in the DB 106. In one example, the original post 110 may be one of the plurality of previous posts selected by the user. As a result, the topic similarity scores, the product similarity scores, the operating system similarity scores and overall similarity scores may be pre-determined as noted above and the top k recommendations may be quickly provided to the user.

[0040] FIG. 2 illustrates an example of an original post 110 and recommendations 122 of previous posts 108 based on an overall similarity score that assumes an equal weighting for each one of the topic similarity score, the product similarity score and the OS similarity score. For example, the original post 110 may be "my HP OfficeJet 6500 will not connect to my PC running Windows 7". A first previous post 108 may be "troubleshooting HP

OfficeJet 6500 connection issues on Windows 7” may have a highest overall similarity score of 2.980. For example, the topic similarity score for the first previous post 108 may be determined to be 0.980 as both the first previous post 108 have a similar topic of connection issues. The product similarity score may be a perfect 1.000 as the product is an exact match of “HP OfficeJet 6500” and the OS similarity score may be a perfect 1.000 as the OS is an exact match of “Windows 7.”

[0041] Similarly, the second previous post 108 may be “problems with HP Office Jet 6500 on Windows” and have an overall similarity score of 2.730. The topic similarity score for the second previous post 108 may be determined to be 0.750. Both are related to a problem but the second previous post 108 is more general to problems than a specific connection issue. The product similarity score may be a perfect 1.000 as the product is an exact match of “HP OfficeJet 6500”. The OS similarity score may be determined to be 0.980 as the OS is identified generally as “Windows” in the second previous post 108, rather than specifically “Windows 7” identified in the original post 110.

[0042] The overall similarity scores for the third, fourth and fifth previous posts 108 may be calculated in a similar way. In the example illustrated in FIG. 2, the recommendations may be provided based on a descending order of the overall similarity score. It should be noted that although five recommendations are illustrated in FIG. 2 any number of recommendations may be provided (e.g., one or more).

[0043] As a result, examples of the present disclosure provide product aware, OS aware and topic based recommendations to customers in response to questions posted by the customer in an online product discussion or forum service. The examples of the present disclosure provide previous posts to the customer to answer the question of the customer that accurately addresses the topic, the product and the OS contained in the customer’s question.

[0044] FIG. 3 illustrates a flowchart of a method 300 for providing product aware, operating system aware and topic based recommendations. In one implementation, the method 300 may be performed by the AS 104 or a computer as illustrated in FIG. 5 and discussed below.

[0045] At block 302, the method 300 begins. At block 304, the method 300 determines a topic similarity score, a product similarity score and an operating system similarity score between an original post and each one of a plurality of previous posts. For example, a processor may identify a topic, a product and an operating system in the original post based on pre-established models. Based on the topic, the product and the operating system that is identified, the processor may compare the topic, the product and the operating system to a topic, a product and an operating system identified in each one of the plurality of previous posts.

[0046] At block 306, the method 300 determines an overall similarity score of the each one of the plurality of previous posts based on the topic similarity score, the product similarity score and the operating system similarity score. In one implementation, the processor may sum the topic similarity score, the product similarity score and the operating system similarity score to determine the overall similarity score. In another implementation, a weight may be applied to the topic similarity score, the product similarity score and the operating system similarity score.

[0047] At block 308, the method 300 sends a recommendation based on the overall similarity score of the each one of the plurality of previous posts to a display device. For example, the recommendations may include a top k number of previous posts based on the overall similarity scores. At block 310, the method 300 ends.

[0048] FIG. 4 illustrates a flowchart of a method 400 for providing product aware, operating system aware and topic based recommendations. In one example, the method 400 may be performed by the AS 104 or a computer as illustrated in FIG. 5 and discussed below.

[0049] At block 402 the method 400 begins. At optional block 404, the method 400 creates a topic model, a product recognition model and an OS recognition model. In one example, the models may be created using the methods described above. In another example, the models may be obtained from third parties that have created the topic model, the product recognition model and the OS recognition model externally.

[0050] At block 406, the method 400 receives an original post. For example, the original post may be a new post to an online product discussion service or forum that includes a question about a company's product. The service or forum may be operated and maintained by the company that produces the product or products to provide quick answers to questions that customers may have. The forum may allow a user to write a post that includes a question about a product and operating system. The post may then be processed to identify a topic, the product and the operating system.

[0051] For example, the user may post on the forum "what does XYZ error message mean on my HP 6150 laptop running OS version 8?" The topic may be identified as "XYZ error message," the product may be identified as "HP 6150 laptop," and the OS may be identified as "OS version 8." The identified topic, product and OS may then be used to determine similarity scores between the original post and each one of the previous posts stored in a database. For example, a similar previous post would most likely have answers already posted that would help the customer with the same problem.

[0052] At block 408, the method 400 determines a topic similarity score, a product similarity score, and an OS similarity score between the original post and each one of a plurality of previous posts. In one example, the topic similarity score may be determined using a cosine similarity score determined using the cosine similarity function and a bagging method described above. In one example, the product similarity score and the OS similarity score may be determined using a combination of a Levenshtein distance and a Jaccard index using the normalized Levenshtein distance function, the left or right ordered Jaccard index function and the combination of functions as described above.

[0053] At block 410, the method 400 determines an overall similarity score of the each one of the plurality of previous posts based on the topic similarity score, the product similarity score and the OS similarity score. In one example, the overall similarity score may be a weighted sum of the topic similarity score, the product similarity score and the OS similarity score. In one example, the overall similarity score may be determined using the function to calculate the overall similarity score described above.

[0054] At block 412, the method 400 sends a recommendation of a top k number of the plurality of previous posts based on the overall similarity score of the each one of the plurality of previous posts to a display device. For example, the top k recommendations may be sent to an endpoint device of the customer having a monitor or display. The customer or user may then review the top k recommendations of the previous posts on his or her endpoint device.

[0055] At block 414, the method 400 determines if there are any additional original posts. For example, if the customer submits another original post or the original post contained multiple products, the method 400 may return to block 406. The method 400 may then repeat blocks 406-414.

[0056] However, if there are no additional original posts, the method 400 may proceed to block 416. At block 416, the method 400 ends.

[0057] As a result, the examples of the present disclosure improve the functioning of an application server or a computer. For example, the AS 104 may provide more accurate recommendations of previous posts in response to an original post by a customer based on a product, OS and topic identified in the original post that could not otherwise be verified without the improvements provided by the present disclosure. In other words, the technological art of matching previous posts to an original post is improved by providing a computer that is modified with the ability to provide accurate recommendations based on the product, the OS and the topic contained in the previous posts and the original post, as disclosed by the present disclosure.

[0058] It should be noted that although not explicitly specified, one or more blocks, functions, or operations of the methods 300 and 400 described above may include a storing, displaying and/or outputting step as required for a particular application. In other words, any data, records, fields, and/or intermediate results discussed in the methods can be stored, displayed, and/or outputted to another device as required for a particular application. Furthermore, blocks, functions, or operations in FIGs. 3 and 4 that recite a determining operation, or involve a decision, do not necessarily require that both branches of the determining operation be practiced. In other words, one of the branches of the determining operation can be deemed as an optional step.

[0059] FIG. 5 depicts a high-level block diagram of a computer that can be transformed to into a machine that is dedicated to perform the functions described herein. Notably, no computer or machine currently exists that performs the functions as described herein. As a result, the examples of the present disclosure improve the operation and functioning of the computer to validate a hard disk drive, as disclosed herein.

[0060] As depicted in FIG. 5, the computer 500 comprises a hardware processor element 502, e.g., a central processing unit (CPU), a microprocessor, or a multi-core processor, a memory or storage 504, e.g., random access memory (RAM) and/or read only memory (ROM), a module 505 for providing product aware, operating system aware and topic based recommendations, and various input/output user interface devices 506 to receive input from a user and present information to the user in human perceptible form, e.g., storage devices, including but not limited to, a tape drive, a floppy drive, a hard disk drive or a compact disk drive, a receiver, a transmitter, a speaker, a display, a speech synthesizer, an output port, an input port and a user input device, such as a keyboard, a keypad, a mouse, a microphone, and the like. Although only one processor element is shown, it should be noted that the computer may employ a plurality of processor elements. Furthermore, although only one computer is shown in the figure, if the method(s) as discussed above is implemented in a distributed or parallel manner for a particular illustrative example, i.e., the blocks of the above method(s) or the entire method(s) are implemented across multiple or parallel computers, then the computer of this figure is intended to represent each of those multiple computers. Furthermore, one or more hardware processors can be utilized in supporting a virtualized or shared computing environment. The virtualized computing environment may support one or more virtual machines representing computers, servers, or other computing devices. In such virtualized virtual machines, hardware components such as hardware processors and computer-readable storage devices may be virtualized or logically represented.

[0061] It should be noted that the present disclosure can be implemented by machine readable instructions and/or in a combination of machine readable

instructions and hardware, e.g., using application specific integrated circuits (ASIC), a programmable logic array (PLA), including a field-programmable gate array (FPGA), or a state machine deployed on a hardware device, a computer or any other hardware equivalents, e.g., computer readable instructions pertaining to the method(s) discussed above can be used to configure a hardware processor to perform the blocks, functions and/or operations of the above disclosed methods. In one example, instructions and data for the present module or process 505 for providing product aware, operating system aware and topic based recommendations, e.g., machine readable instructions can be loaded into memory 504 and executed by hardware processor element 502 to implement the blocks, functions or operations as discussed above in connection with the exemplary methods 300 and 400. Furthermore, when a hardware processor executes instructions to perform “operations”, this could include the hardware processor performing the operations directly and/or facilitating, directing, or cooperating with another hardware device or component, e.g., a co-processor and the like, to perform the operations.

[0062] The processor executing the machine readable instructions relating to the above described method(s) can be perceived as a programmed processor or a specialized processor. As such, the present module 505 for providing product aware, operating system aware and topic based recommendations, including associated data structures, of the present disclosure can be stored on a tangible or physical (broadly non-transitory) computer-readable storage device or medium, e.g., volatile memory, non-volatile memory, ROM memory, RAM memory, magnetic or optical drive, device or diskette and the like. More specifically, the computer-readable storage device may comprise any physical devices that provide the ability to store information such as data and/or instructions to be accessed by a processor or a computing device such as a computer or an application server.

[0063] It will be appreciated that variants of the above-disclosed and other features and functions, or alternatives thereof, may be combined into many other different systems or applications. Various presently unforeseen or unanticipated alternatives, modifications, variations, or improvements therein

may be subsequently made by those skilled in the art which are also intended to be encompassed by the following claims.

What is claimed is:

1. A method, comprising:
 - determining, by a processor, a topic similarity score, a product similarity score and an operating system similarity score between an original post and each one of a plurality of previous posts;
 - determining, by the processor, an overall similarity score of the each one of the plurality of previous posts based on the topic similarity score, the product similarity score and the operating system similarity score; and
 - sending, by the processor, a recommendation based on the overall similarity score of the each one of the plurality of previous posts to a display device.
2. The method of claim 1, wherein the original post comprises a question about a product.
3. The method of claim 1, wherein the original post and the plurality of previous posts are within an online product discussion service.
4. The method of claim 1, wherein the determining the topic similarity score comprises:
 - creating, by the processor, a latent topic model using a machine learning algorithm; and
 - calculating, by the processor, a similarity between the original post and the each one of the plurality of previous posts based upon the latent topic model.
5. The method of claim 4, wherein the latent topic model comprises a plurality of different latent topic models and further comprising, performing a calculation of the similarity for each one of the plurality of different latent topic models and averaging the similarity for the each one of the plurality of different latent topic models.

6. The method of claim 1, wherein the determining the product similarity score and the operating system similarity score each comprises:

calculating, by the processor, a first similarity score based on a difference between a first string and a first length of a product or a operating system in the original post and a second string and a second length of the product or the operating system in one of the plurality of previous posts;

calculating, by the processor, a second similarity score based on the first string and the second string, the first length of the first string and the second length of the second string and a position of each element within the first string and the second string; and

combining, by the processor, the first similarity score and the second similarity score.

7. The method of claim 6, wherein the first similarity score is determined using a Levenshtein distance function.

8. The method of claim 6, wherein the second similarity score is determined using an ordered Jaccard function.

9. The method of claim 1, wherein the topic similarity score, the product similarity score and the operating system similarity score each has a different weight for calculating the overall similarity score.

10. An apparatus comprising:

a processor;

a storage coupled to the processor, wherein the storage is configured to store a plurality of previous posts;

a topic similarity module in communication with the processor, wherein the topic similarity module is configured to determine a topic similarity score between an original post and each one of the plurality of previous posts;

a product similarity module in communication with the processor, wherein

the product similarity module is configured to determine a product similarity score between the original post and each one of the plurality of previous posts;

an operating system similarity module in communication with the processor, wherein the operating system similarity module is configured to determine an operating system similarity score between the original post and each one of the plurality of previous posts; and

a multi-aspect recommendations module in communication with the topic similarity module, the product similarity module, the operating system module and the processor, wherein the multi-aspect recommendations module is configured to determine an overall similarity score of the each one of the plurality of previous posts based on the on the topic similarity score, the product similarity score and the operating system similarity score and providing a recommendation based on the overall similarity score of the each one of the plurality of previous posts to a display device.

11. The apparatus of claim 10, further comprising:

a product recognition module in communication with the processor, wherein the product recognition module is configured to apply one or more recognition models to the original post to identify a topic, a product and an operating system contained in the original post.

12. The apparatus of claim 10, wherein processor is further configured to:

create a latent topic model using a machine learning algorithm; and
determine a similarity between the original post and the each one of the plurality of previous posts based upon the latent topic model.

13. The apparatus of claim 10, wherein product similarity module and the operating system similarity module are each further configured to:

determine a first similarity score based on a difference between a first string and a first length of a product or a operating system in the original post and a second string and a second length of the product or the operating system in one of the plurality of previous posts;

determine a second similarity score based on the first string and the second string, the first length of the first string and the second length of the second string and a position of each element within the first string and the second string; and

combine the first similarity score and the second similarity score.

14. The apparatus of claim 10, wherein the topic similarity score, the product similarity score and the operating system similarity score each has a different weight for calculating the overall similarity score.

15. A non-transitory machine-readable storage medium storing instructions executable by a processor, the machine-readable storage medium comprising:

instructions to determine a topic similarity score, a product similarity score and an operating system similarity score between an original post and each one of a plurality of previous posts;

instructions to determine an overall similarity score of the each one of the plurality of previous posts based on the topic similarity score, the product similarity score and the operating system similarity score; and

instructions to send a recommendation of a top K number of the plurality of previous posts based on the overall similarity score of the each one of the plurality of previous posts to a display device.

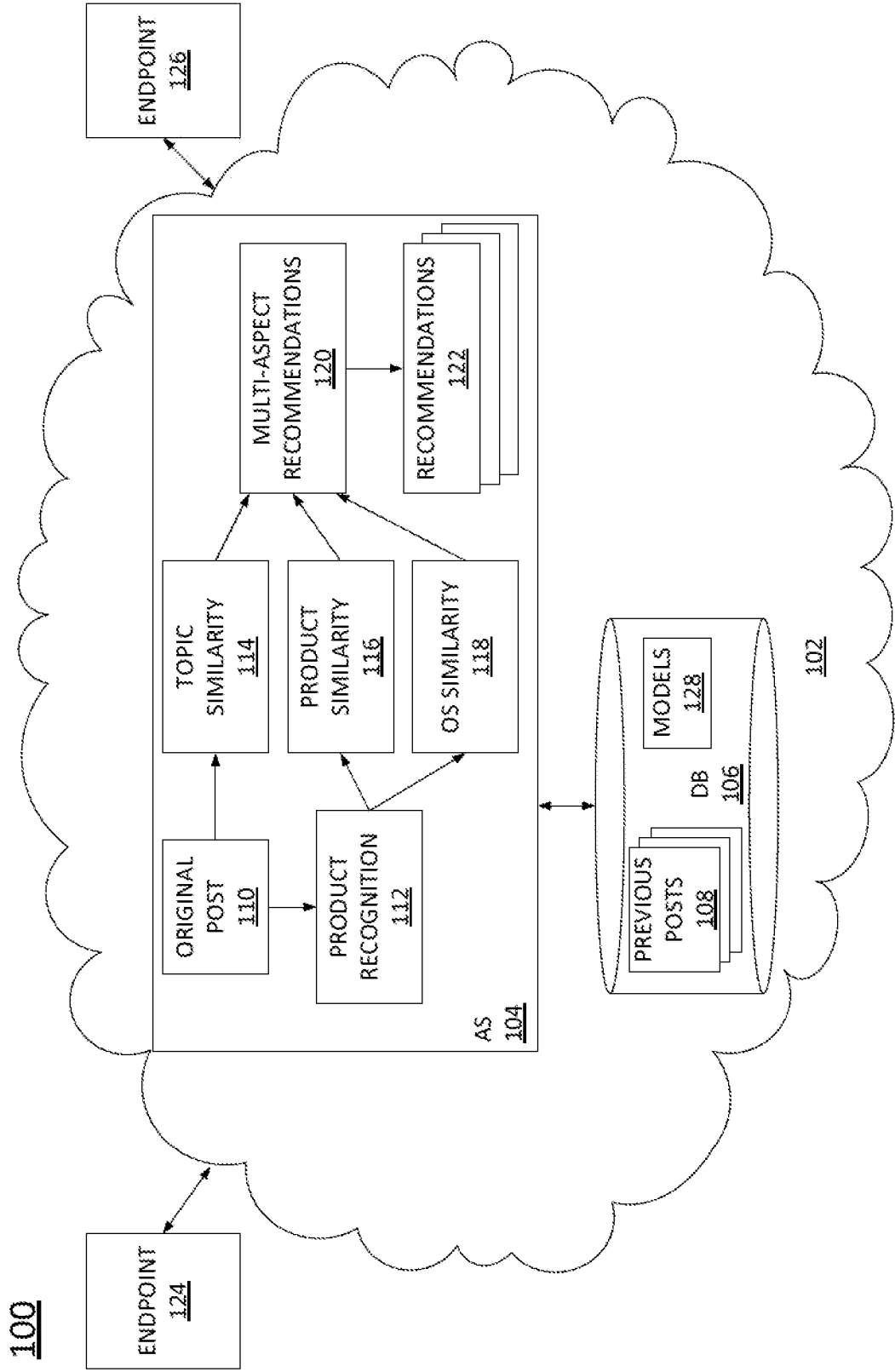


FIG. 1

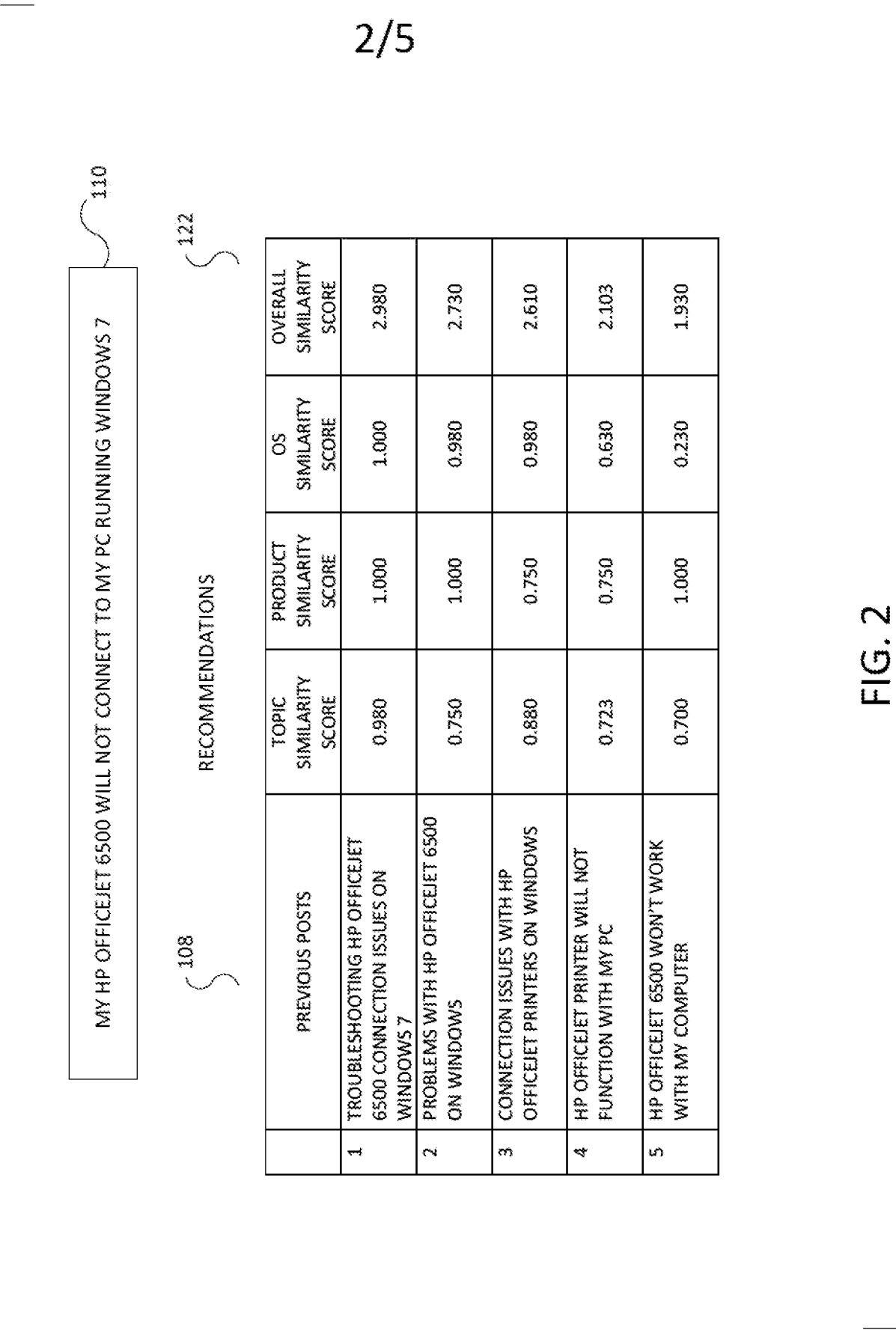


FIG. 2

300

3/5

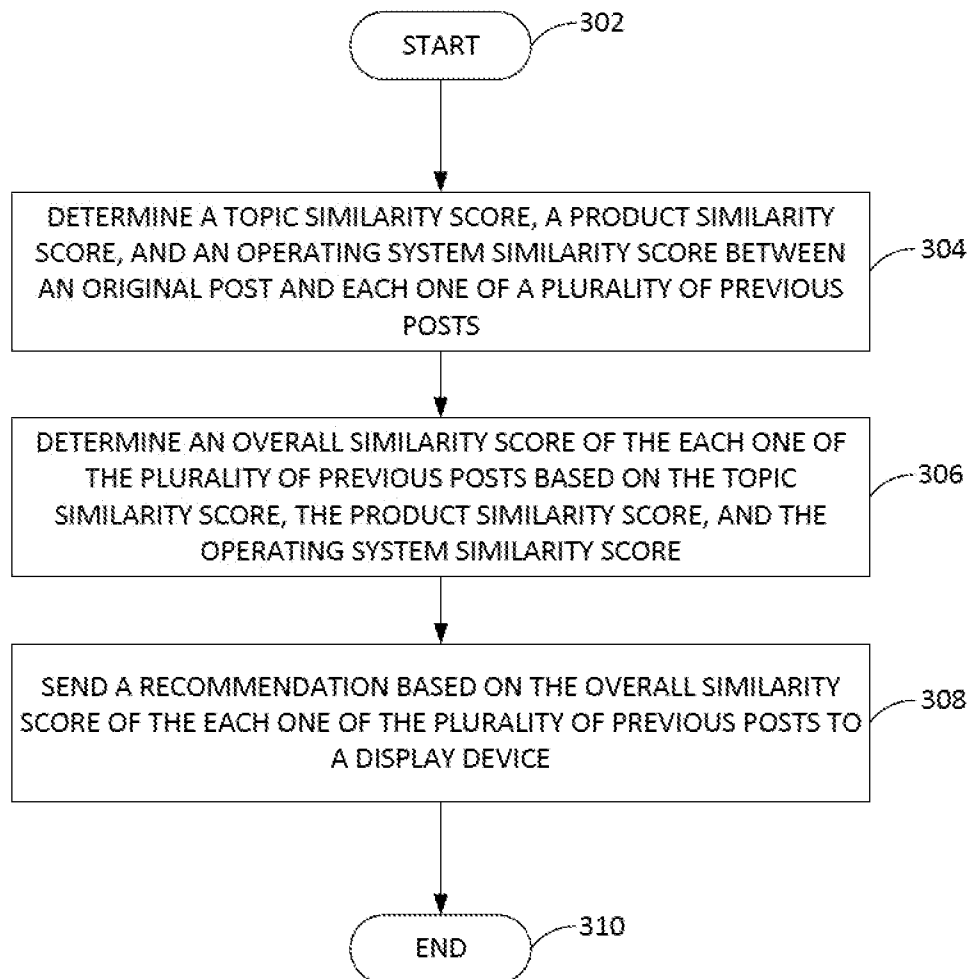


FIG. 3

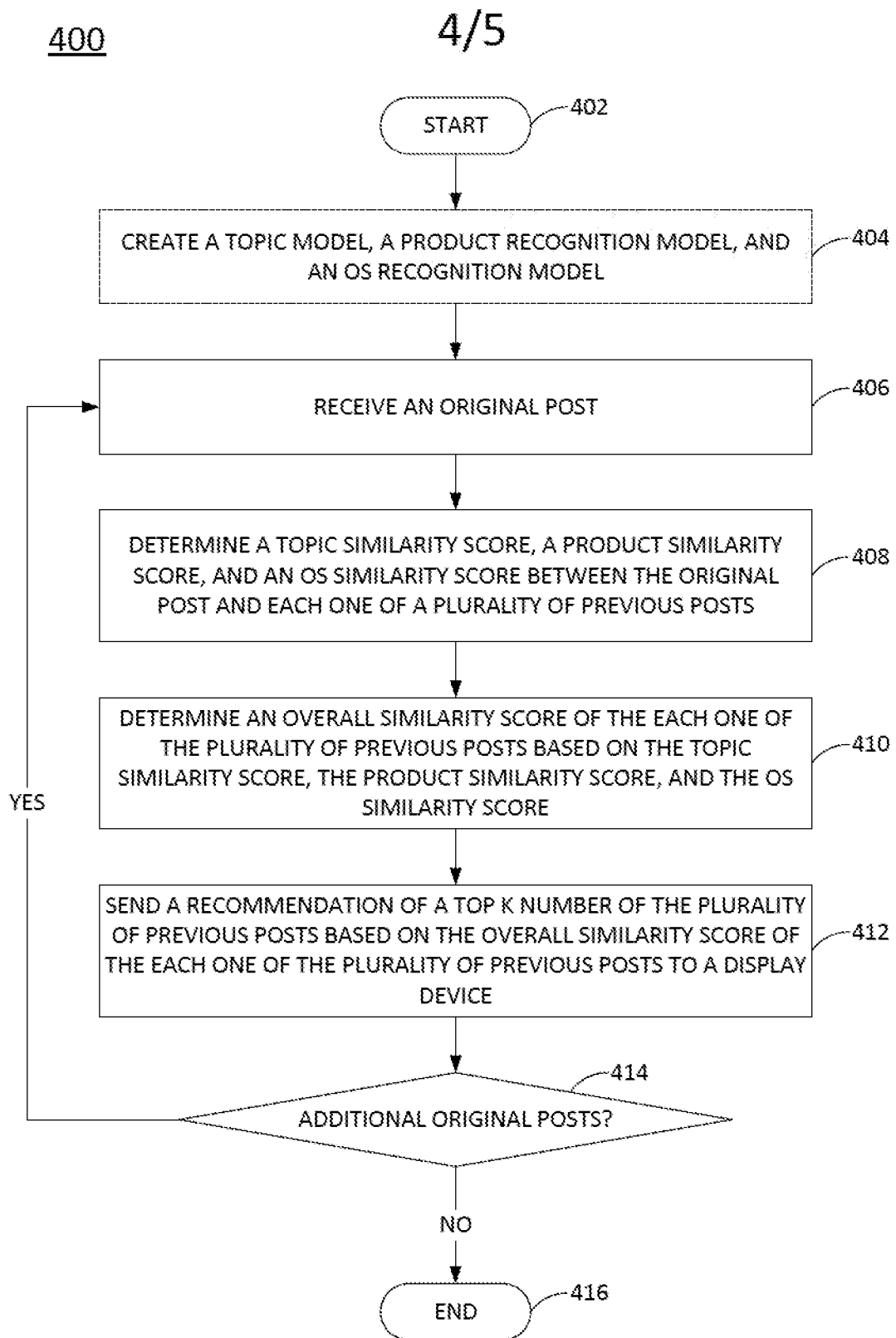


FIG. 4

5/5

500

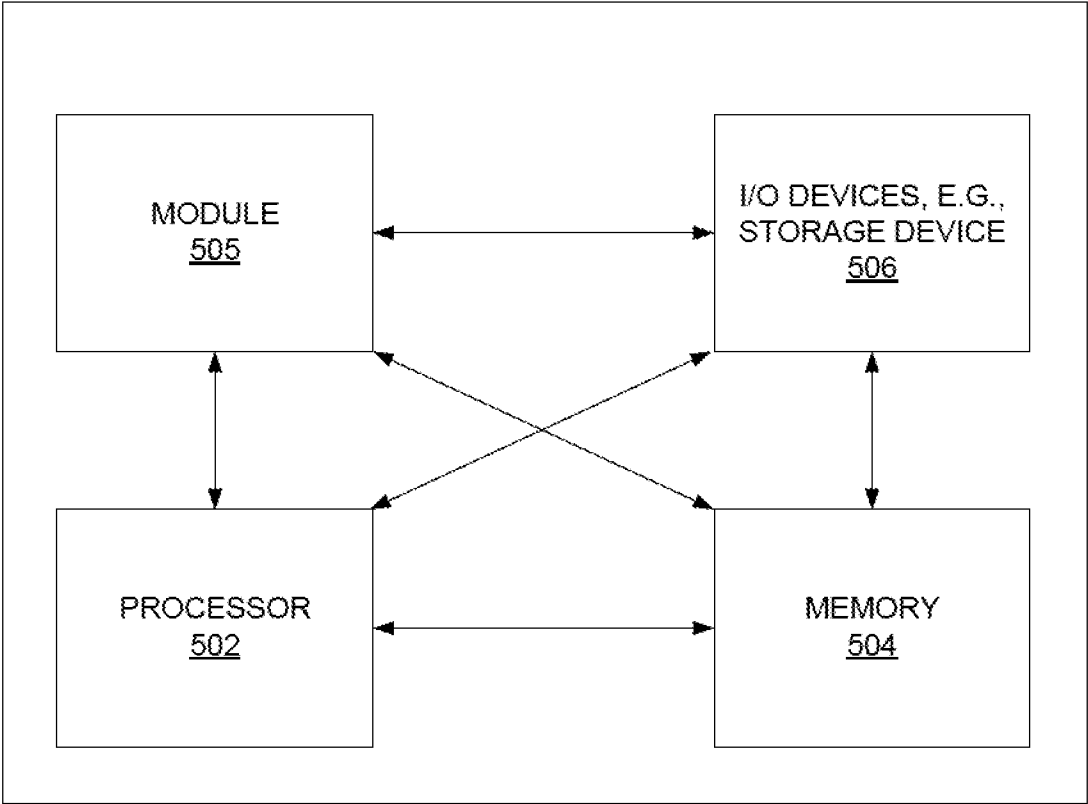


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/013714**A. CLASSIFICATION OF SUBJECT MATTER****G06Q 30/06(2012.01)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)

G06Q 30/06; G06Q 50/30; G06F 17/30; G06N 5/02

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: product, similarity, score, topic, operating system, post, question, answer, recommendation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2010-0235311 A1 (YUNBO CAO et al.) 16 September 2010 See abstract, paragraphs [0030]-[0031], [0037]-[0038], [0042]-[0046], [0065]-[0066], [0071], claims 1-9, 11-14, 19 and figures 3-4.	1-15
A	JP 2012-168653 A (M-WARP INC.) 06 September 2012 See paragraphs [0012]-[0013], [0020]-[0026], [0037]-[0038], claims 1-2 and figures 1, 3-5.	1-15
A	US 2007-0073683 A1 (KENJI KOBAYASHI et al.) 29 March 2007 See abstract, paragraphs [0020]-[0030], claims 1-3, 6-8 and figures 1-5, 8.	1-15
A	JP 2009-043263 A (NHN CORP.) 26 February 2009 See paragraphs [0012]-[0020], [0035]-[0040], claims 1-8, 12 and figures 1-4.	1-15
A	KR 10-2009-0114338 A (KT CORP. et al.) 03 November 2009 See abstract, claims 1-4, 6-7, 9-10 and figures 1-4.	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

16 October 2015 (16.10.2015)

Date of mailing of the international search report

19 October 2015 (19.10.2015)

Name and mailing address of the ISA/KR

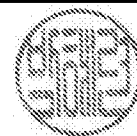
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

PARK, Hye Lyun

Telephone No. +82-42-481-3463



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/013714

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0235311 A1	16/09/2010	US 2010-0235343 A1	16/09/2010
JP 2012-168653 A	06/09/2012	None	
US 2007-0073683 A1	29/03/2007	CN 100535898 C	02/09/2009
		CN 1871605 A	29/11/2006
		JP 2005-128873 A	19/05/2005
		JP 3820242 B2	13/09/2006
		US 7587420 B2	08/09/2009
		WO 2005-041068 A1	06/05/2005
JP 2009-043263 A	26/02/2009	JP 4729607 B2	20/07/2011
		KR 0-2009-0016329 A	13/02/2009
		KR 10-0898459 B1	21/05/2009
KR 10-2009-0114338 A	03/11/2009	KR 10-1199474 B1	09/11/2012