



European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

LINK SPAM DETECTION USING SMOOTH CLASSIFICATION FUNCTION

BACKGROUND

5 [0001] Web search engines are currently in wide use, and are used to return a ranked list of web sites in response to a search query input by a user. It can be very valuable to have a web page returned high in the ranked list of web pages for a wide variety of different queries. This may
10 increase the likelihood that a user will view a given web page.

[0002] Therefore, in order to increase web traffic to a given site, the authors of certain sites have tried to artificially manipulate the ranked list returned by search
15 engines such that the web sites authored by those authors are ranked higher than they would normally be ranked. The particular manipulation techniques used by such authors depends on how a given web search engine ranks the pages for a given query. Any of the different manipulation techniques
20 used by such authors are referred to as "spamming" techniques.

[0003] Some search engines use link analysis algorithms in order to generate the ranked list of web pages returned in response to a query. In general, link analysis
25 algorithms identify an importance of a given web page, based upon the number of links that point to that web page. It is assumed that related web pages (those that have related content) have links to one another. Therefore, the more links that point to a web page, the more important the web
30 page may be regarded by the search engine.

[0004] In order to manipulate this type of search engine, web spammers (those employing spamming techniques) sometimes attempt to create a large number of links to their web pages by having unrelated web pages (web page with unrelated

content) linked to their web pages. This can be done using automated techniques to post links to their web sites onto other web pages, or simply by creating a large number of their own web pages and web sites, and then placing links in those web pages and web sites to all the other web pages and web sites which they created. This increases the number of links to any given web page or web site created by the author, regardless of whether it has related content. Similarly, some web sites reciprocally exchange links. When two unrelated web sites exchange links, at least one, and possibly both, of them are very likely to be spam (web sites that receive the benefit of spamming techniques).

[0005] It can be seen that spamming techniques can produce spam that misleads a search engine into returning low quality, or even entirely irrelevant, information to a user in response to a query. Therefore, a number of techniques have been developed in order to identify spam so that it can be removed from the ranked search results returned by a search engine. For instance, human experts can generally identify web spam in a very effective manner. However, it is quite easy for a spammer to create a large number of spam pages and to manipulate their link structure. It is thus impractical to detect web spam using only human judges. Therefore, some automatic approaches have been developed to identifying spam. One category of such approaches is referred to as a supervised approach in which some known examples of spam are provided to the system, and the system learns to recognize spam from those examples.

[0006] One such technique builds a ranking measure for web pages modeled on a user randomly following hyperlinks through the web pages. This ranking measure is well known as PageRank used by the Google search engine. At each web page, the modeled user either selects an outlink uniformly at random to follow with a certain probability, or jumps to

a new web page selected from the whole web uniformly at random with the remaining probability. The stationary probability of a web page in this "random walk" is regarded as the ranking score of the web page. The basic assumption behind such a technique is that a hyperlink from one page to another is a recommendation of the second page by the author of the first page. If this assumption is recursively applied, then a web page is considered to be important if many important web pages point to it.

10 **[0007]** By using random jumps to uniformly selected pages, this system accommodates the problem that some high quality pages have no out links, although they are pointed to by many other web pages.

[0008] This concept of random jumps has also been adopted, in another way, to address the problem of web spam. Basically, the random user described above is allowed to jump to a set of pages (seed pages) which have been judged as being high quality, normal pages, by human experts. Assuming this choice for the random jumps, the stationary probability of a web page is regarded as its trust score, and a web page with a trust score smaller than a given threshold value is considered to be spam.

20 **[0009]** This type of system can also be understood as follows: initially, only the selected good seed pages have trust scores equal to one, and the trust scores of other web pages are zero. Each seed page then iteratively propagates its trust score to its neighbors, and its neighbors further propagate their received scores to their neighbors. The underlying assumption in this algorithm is that web pages of high quality seldom point to spam pages.

30 **[0010]** A counterpart to this algorithm allows the random web user to either select an inlink uniformly at random to follow, in reverse, with a certain probability, or jump to a new web page randomly selected from a web page set which has

been judged as spam by human experts with the remaining probability. The stationary probability of a web page is, in this system, referred to as its antitrust rank, or antitrust score. A web page will be classified as spam if
5 its score is larger than a chosen threshold value. In terms of the propagation understanding, the scores in this system are propagated in the reverse direction along the inlinks. The basic underlying assumption of this type of system is that a web page pointing to spam pages is likely to be spam,
10 itself.

[0011] Another system is referred to as a functional ranking system. It considers a general ranking function that depends on incoming paths of various lengths weighted by some chosen damping function that decreases with
15 distance. In other words, links from pages that are a greater distance from the subject web page are weighted by weight that is damped less than links from closer web pages. That is, spam pages may gain an artificially high score under a system that simply ranks the pages based on the
20 number of links to it, because a spam page may be formed by using a spamming technique to have many incoming links from its immediate neighbor pages. However, spam pages of this type can be demoted using this system by choosing a damping function that ignores the direct contribution of links from
25 pages directly adjacent the given page, and only valuing links that start at least one link away from the subject page.

[0012] Yet another technology to be considered is general machine learning technology. In this technology, features
30 must be selected that are useful in detecting spam, and each web page is then represented as a vector having each element described by one type of spam feature. The features can be the number of inlinks, the number of outlinks, scores under any of the above-mentioned algorithms, etc. Then, a

classifier is chosen, such as a neural network, a decision tree, a support vector machine (SVM), etc., and it is trained with a set of examples of normal and spam web pages which have been judged by human experts. The trained
5 classifier is then used to predict a given web page as spam or not spam (i.e., as spam or a content page). One difficulty with this methodology is that the efficiency of a spam feature is generally validated only on the web pages which are not sampled from the entire web uniformly at
10 random, but instead from large websites and highly ranked web pages. Consequently, the trained classifier is biased to those selected pages, and it does not generalize well to the entire web.

[0013] The discussion above is merely provided for
15 general background information and is not intended to be used as an aid in determining the scope of the claimed subject matter.

SUMMARY

[0014] A collection of web pages is considered as a
20 directed graph in which the pages themselves are nodes and the hyperlinks between the pages are directed edges in the graph. A trusted entity identifies training examples for spam pages and normal pages. A random walk is conducted
25 through the directed graph. A classifier built on the random walk estimates a classification function that changes slowly on densely connected subgraphs within the directed graph. The classification function assigns a value to each of the nodes in the directed graph and identifies them as
30 spam or normal pages based upon whether the value meets a given function threshold value.

[0015] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not

intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter. The claimed subject matter is not limited to
5 implementations that solve any or all disadvantages noted in the background.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is a block diagram of one illustrative embodiment of a link spam detection system.

10 **[0017]** FIG. 2 is a flow diagram illustrating one embodiment of the overall operation of the system shown in FIG. 1.

[0018] FIG. 3 is a flow diagram illustrating one embodiment of obtaining stationary and transition
15 probabilities for a directed graph.

[0019] FIG. 4 is one embodiment of a strongly connected directed web graph.

[0020] FIG. 5 illustrates one embodiment of the web graph shown in FIG. 4 after classification.

20 **[0021]** FIG. 6 is a block diagram of one embodiment of an illustrative computing environment.

DETAILED DESCRIPTION

[0022] Link spam detection in the present specification is discussed in terms of a machine learning problem of
25 classification on directed graphs. FIG. 1 is a block diagram of one illustrative embodiment of a link spam detection system 100. System 100 shows trusted entity 102, directed graph 104, random walk component 106, classifier training component 108, spam classifier 110, and spam
30 detection system 112.

[0023] In system 100, a collection of web pages 114 is also shown. The collection of web pages 114 is considered a directed graph, in that the web pages themselves in collection 114 are nodes in the graph while hyperlinks

between those web pages are directed edges in the directed graph. Of course, it will be appreciated that the present system can be applied at the domain/host level as well, where the domains/hosts are nodes in the graph and
5 hyperlinks among the web pages in the domains/hosts are the directed edges. For purposes of the present discussion, however, and by way of example only, reference will be made to the web pages in collection 113 as the nodes and hyperlinks between those pages as the directed edges.

10 **[0024]** FIG. 2 is a flow diagram illustrating one illustrative embodiment of the operation of system 100 shown in FIG. 1. FIGS. 1 and 2 will now be described in conjunction with one another.

[0025] It will first be noted that, if web page
15 collection 114 does not form a strongly connected graph, then it is first decomposed into strongly connected components, and the present process proceeds with respect to each of the strongly directed components. The precise definition of what makes a strongly connected graph, or
20 strongly connected component, is set out below. Briefly, however, a graph can be considered strongly connected if each vertex (or node) in the graph is connected to every other vertex (or node) in the graph by some path of directed edges. Decomposing the directed graph into strongly
25 connected components is illustrated by block 150 in FIG. 2 and is shown in phantom indicating that it is only performed, if necessary.

[0026] The web page collection 114 is also provided to
30 trusted entity 102, such as a human expert, in identifying link spam. Trusted entity 102 then identifies some examples of spam web pages in web page collection 114 as spam training examples 116. Trusted entity 102 also identifies good web pages (or normal web pages) in web page collection

114 as normal training examples 118. Obtaining these examples is indicated by block 152 in FIG. 2.

[0027] Random walk component 106 then receives a definition of a random walk on directed graph 104 (or each strongly connected component in directed graph 104), the random walk being defined by transition probabilities (set out below in Eqs. 20-22). Receiving this definition is indicated by block 153 in FIG. 1. Based on the defined random walk, component 106 obtains stationary probabilities associated with each node in directed graph 104. The stationary probabilities are indicated by block 120 in FIG. 1. Obtaining these probabilities for directed graph 104 is indicated by block 154 in FIG. 2, and the stationary probabilities are obtained by conducting the defined random walk through the directed graph 104. This is discussed in greater detail below with respect to FIG. 3.

[0028] In any case, once examples 116 and 118 and probabilities 120 and 122 are obtained, classifier training component 108 trains a classifier that can be used in link spam detection. The classifier is shown in FIG. 1 as spam classifier 110. In one embodiment, training a classifier is performed by generating a smooth classification function based on the probabilities 120 and 122 over the detected graph 104. In generating the classification function, the values of the classification function are forced to be close to known values for examples 116 and 118. In other words, the values of the classification function are forced to be close to the values that indicate spam and normal pages at the nodes in the graph that are actually known to be spam and normal pages as identified by the trusted entity 102. For example, assume that the value of -1 indicates that the node is spam, while the value of 1 indicates that the node is a normal content page. Then, the classification function is forced to be at least close to the values of 1 or -1 at

the pages known to be normal pages and spam pages, respectively. This is indicated by block 156 in FIG. 2.

[0029] The closeness between the classification function and the known values can be measured in a variety of
5 different ways. For example, the closeness can be measured using least square loss, hinge loss, precision/recall measure, the F1-score, the ROC score, or the AUC score, as examples.

[0030] In accordance with one embodiment, the
10 classification function is not only close to known values at known nodes, but it is relatively smooth in that it changes relatively slowly on densely connected subgraphs. In other words, the nodes that reside close to one another on the subgraph may likely have values which are relatively close
15 to one another. However, if they are known to be one spam node and one normal node, respectively, then the classification function changes by a large amount between those nodes, but this lack of smoothness is penalized in the chosen cost function that is optimized.

[0031] This can provide significant advantages over prior
20 systems. In prior systems, for instance, those pages closely related to spam pages were deemed as spam while all other pages were deemed as normal pages. In another prior system, those pages close to normal pages were deemed normal
25 pages, while all other pages were deemed spam. The present system includes information related to both normal pages and spam pages in classifying a given page under consideration as content or spam. Also, because pages that are relatively close to one another on the directed graph are assumed to be
30 the same type (pages close to a known spam page are likely to be spam pages, while pages close to a known normal page are likely to be normal pages) by making the function smooth and relatively slow changing pages in the directed subgraph that are close to a known normal content page will have

classification function values that more likely indicate it to be a normal content page. Similarly, those pages in the directed subgraph that are close to a spam page will have classification function values that are likely to indicate that it will be a spam page. The classification function value can change abruptly, if necessary. Again, however, this is penalized.

[0032] In any case, the spam classifier is then used to assign values to all unlabeled nodes in the directed graph 104. A threshold can be set, and those pages that meet the classification threshold may be deemed to be normal pages, while those nodes having a value that does not meet the threshold value may be deemed to be spam pages. In one embodiment, simply the sign of the value calculated using the classification function is used to determine whether the associated node is spam or content. This effectively sets the classification function threshold value at 0. It may, however, be desirable to set the value at a level other than 0. For instance, if it is more desirable in a given application to error on the side of classifying spam pages as normal pages, then the threshold may be set below 0. On the other hand, if a given application deems it more desirable to error on the side of classifying normal pages as spam pages, then the threshold value can be set above 0, etc. Using the classification function embodied in spam classifier 110 to perform spam detection is indicated by block 158 in FIG. 2.

[0033] FIG. 3 is a flow diagram showing one illustrative embodiment of random walk component 106 in obtaining the stationary probabilities 120. In one embodiment, random walk component 106 simply selects, at random, a starting node in directed graph 104. This is indicated by block 180 in FIG. 3. Component 106 then randomly follows links in graph 104 starting at the selected starting node. This is

indicated by block 182. It will be noted that, at each step, component 106 can follow inlinks or outlinks from the given web page uniformly at random. If the outlinks are followed, component 106 simply follows links from the current page to another page to which it is linked through an outlink. However, if inlinks are used, then component 106 travels backward along the links that link to the current page, to the page at which the inlink originates. For the present description, following the outlink will be used, although either inlinks or outlinks could be used, as desired.

[0034] Component 106 continues to follow the links, uniformly at random, for a sufficient amount of time. This is indicated by block 184. The amount of time will depend upon the size of the collection of web pages 114. As component 106 is performing this random walk, it calculates the stationary probability distribution for the various nodes in graph 104. The "transition probabilities" are the probabilities of transitioning from any given node on graph 104 to another node. The "stationary probability distribution" assumes that component 106 starts from a randomly chosen node in graph 104, and jumps to an adjacent node by choosing an outlink. Assume for the sake of example that this is repeated infinitely many times for the various nodes in graph 104. Then, if graph 104 is connected (that is, using such a random walk, any point can be reached from any other point), then the fraction of time component 106 spends at a given node converges to a fixed number (where the corresponding numbers for all nodes sum to 1), and that fixed number is actually independent of the choice of starting nodes. In other words, the stationary probability distributions are the probabilities of being in any given node on directed graph 104.

[0035] Component 106 can use any given metric to determine whether it has performed the random walk sufficiently long enough. For example, where the stationary probabilities do not change by a given amount (that is they are changing very little or very slowly with each given iteration of the jump) then, component 106 may deem that it has performed the random walk long enough. In any case, once the random walk has been performed for sufficiently long time, component 106 calculates the final stationary probabilities 120 that are output to classifier training component 108. This is indicated by block 186 in FIG. 3.

[0036] It may seem at first that performing the classification over a large directed graph may take an inordinately large amount of time. It has been found that it can be done quite quickly, using a relatively small number of training examples. For instance, in a directed graph having 20 million web pages connected by directed edges (links) with 10,000 examples of spam web pages and 20,000 examples of content web pages, the classification can be performed in several minutes.

[0037] Having thus described transductive detection of spam pages in an intuitive sense, it will now be described in a more formal way. First, a discussion of some specific items of notation will be made.

[0038] Let $G = (V, E)$ denote a directed graph, where V is the set of vertices, and E the set of edges. For a given edge $e \in E$; denote the initial vertex of e by e^- , and the terminal vertex of e by e^+ . Also denote by (u, v) an edge from the vertex u to the vertex v . It is clear that an undirected graph can be regarded as a directed graph with each edge being double oriented. A graph G is weighted if it is associated with a function $w : E \rightarrow \mathbb{R}^+$ which assigns a positive number $w(e)$ to each edge e of G : Let $G = (V, E, w)$ denote a weighted directed graph. The function w is called

the weight function of G : The in-degree d^- and the out-degree d^+ of a vertex $v \in V$ are respectively defined as:

$$d^-(v) = \sum_{\{e|e^-=v\}} w(e), \text{ and } d^+(v) = \sum_{\{e|e^+=v\}} w(e)$$

Eq. 1

[0039] A path is a tuple of vertices (v_1, v_2, \dots, v_p) with the property that $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq p-1$. A directed graph is strongly connected when for every pair of vertices u and v there is a path in which $v_1 = u$ and $v_p = v$. For a strongly connected graph, there is an integer $k \geq 1$ and a unique partition $V = V_0 \cup V_1 \cup \dots \cup V_{k-1}$ such that for all $0 \leq r \leq k-1$ each edge $(u, v) \in E$ with $u \in V_r$ has $v \in V_{r+1}$, where $V_k = V_0$; and k is maximal, that is, there is no other such partition $V = V'_0 \cup \dots \cup V'_{k'-1}$ with $k' > k$.

[0040] When $k = 1$, the graph is aperiodic; otherwise the graph is periodic.

[0041] For a given weighted directed graph, there is a natural random walk on the graph with the transition probability function $p: V \times V \rightarrow \mathbb{R}^+$ defined by:

$$p(u, v) = \frac{w(u, v)}{d^+(u)}$$

Eq. 2

[0042] for all $(u, v) \in E$, and 0 otherwise. If the graph is strongly connected, there is a unique function $\pi: V \rightarrow \mathbb{R}^+$ which satisfies:

$$\sum_{u \in V} \pi(u) p(u, v) = \pi(v), \text{ and } \sum_v \pi(v) = 1$$

Eq. 3

[0043] The first equation in Equation 3 is called the balance equation, and π is called the Perron vector. For a general directed graph, there is no closed form solution for π . If the graph is both strongly connected and aperiodic, the random walk defined by Eq. 2 converges to the Perron vector π . Unless stated otherwise, the directed graphs

considered are always assumed to be strongly connected. One embodiment of a strongly connected graph is shown in FIG. 4. The nodes (or vertices) are labeled 1-9 while the edges are shown as arrows.

5 [0044] Now, a number of discrete operators on directed graphs are defined. The operators are discrete analogs of the corresponding differential operators on Riemannian manifolds. As discussed below, the discrete operators are then used to develop a discrete analog of classical
 10 regularization theory. Consequently, as in other regularization based machine learning algorithms in vectorial spaces (for instance, support vector machines (SVMs)) the present classification algorithm for directed graphs is derived from the discrete regularization.

15 [0045] In any case, let $F(V)$ denote the set of all real-valued functions on V ; and $F(E)$ the set of all real-valued functions on E . The function set $F(V, \mu)$ can be regarded as a Hilbert space $H(V, \mu)$ with the inner product defined by:

$$\langle \phi, \psi \rangle_{H(V)} = \sum_{v \in V} \phi(v) \psi(v) \pi(v) \tag{Eq. 4}$$

20 [0046] where $\phi, \psi \in F(V)$. Let $c(e) = \pi(e^-) p(e)$. The number $c(e)$ is called the *ergodic flow* on e . It is easy to check that the ergodic flow is a *circulation*, that is:

$$\sum_{\{e|e^- = v\}} c(e) = \sum_{\{e|e^+ = v\}} c(e), \forall v \in V \tag{Eq. 5}$$

[0047] A Hilbert space $H(E)$ over $F(E)$ can be constructed
 25 with the inner product defined by:

$$\langle \mathcal{G}, \psi \rangle_{H(E)} = \sum_{e \in E} \mathcal{G}(e) \psi(e) c(e) \tag{Eq. 6}$$

[0048] Where $\mathcal{G}, \psi \in F(E)$.

[0049] The discrete gradient $\nabla: H(V) \rightarrow H(E)$ is defined as an operator:

30 $(\nabla \phi)(e) := \phi(e^+) - \phi(e^-), \forall \phi \in H(V) \tag{Eq. 7}$

[0050] For simplicity, $(\nabla\phi)(e)$ is also denoted as $\nabla_e\phi$. For gaining an intuition of this definition, one may imagine a set of buckets, and some of them are connected by tubes. Assume a tube e which connects buckets e^- and e^+ , and the quantities of fluid in buckets e^- and e^+ to be $\phi(e^-)$ and $\phi(e^+)$. Then the flow through the tube should be proportional to the pressure difference and hence to $\phi(e^+) - \phi(e^-)$. When the fluid distributes itself uniformly among buckets, that is, ϕ is constant, the pressure differences will disappear and consequently there will be no flow in tubes any more, that is, $\nabla\phi$ vanishes everywhere.

[0051] As in the continuous case, the *discrete divergence* $\text{div} : H(E) \rightarrow H(V)$ can be defined as the dual of $-\nabla$ that is:

$$\langle \nabla\phi, \psi \rangle_{H(E)} = \langle \phi, -\text{div}\psi \rangle_{H(V)} \tag{Eq. 8}$$

[0052] where $\phi \in H(V), \psi \in H(E)$. By a straightforward computation, the following is obtained:

$$(\text{div}\psi)(v) = \frac{1}{\pi(v)} \left(\sum_{\{e|e^-=v\}} c(e)\psi(e) - \sum_{\{e|e^+=v\}} c(e)\psi(e) \right) \tag{Eq. 9}$$

[0053] By following the above fluid model, the divergence measures the net flows at buckets. Now the concept of circulation can be generalized in terms of divergence. A function $\psi \in H(E)$ is called a circulation if and only if $\text{div}\psi = 0$.

[0054] The discrete Laplacian $\Delta : H(V) \rightarrow H(V)$ is defined by:

$$\Delta := -\frac{1}{2} \text{div} \circ \nabla \tag{Eq. 10}$$

[0055] Compared with its counterpart in the continuous case, the additional factor in Eq. 10 is due to edges being oriented. From Eq. 10:

$$\langle \Delta\phi, \phi \rangle_{H(V)} = \frac{1}{2} \langle \nabla\phi, \nabla\phi \rangle_{H(E)} = \langle \phi, \Delta\phi \rangle_{H(V)} \tag{Eq. 11}$$

[0056] Note that the first equation in Eq. 11 is a discrete analog of Green's formula. In addition, Eq. 11 implies that Δ is self-adjoint. In particular, when $\varphi = \phi$, then:

$$5 \quad \langle \Delta\varphi, \varphi \rangle_{H(V)} = \frac{1}{2} \langle \nabla\varphi, \nabla\varphi \rangle_{H(E)} = \frac{1}{2} \|\nabla\varphi\|_{H(E)}^2 \quad \text{Eq. 12}$$

[0057] which implies that Δ is positive semi-definite. By substituting Eqs. 7 and 9 into Eq. 10:

$$(\Delta\varphi)(v) = \varphi(v) - \frac{1}{2\pi(v)} \left(\sum_{\{e|e^- = v\}} c(e)\varphi(e^-) + \sum_{\{e|e^+ = v\}} c(e)\varphi(e^+) \right) \quad \text{Eq. 13}$$

10 **[0058]** when the graph is undirected, that is, each edge being double oriented, Eq. 13 reduces to:

$$(\Delta\varphi)(v) = \varphi(v) - \frac{1}{d(v)} \sum_{u \sqcup v} w(u, v)\varphi(u) \quad \text{Eq. 14}$$

[0059] Eq. 14 has been widely used to define the Laplacian for an undirected graph. Now, define a family of
 15 functions $\{\delta_v\}_{v \in V}$ with $\delta_v(u) = I_{u=v}$, which is clearly a basis of $H(V)$. The matrix form of Δ with respect to this basis has the following components:

$$\Delta_{am}(u, v) = \begin{cases} -\frac{c(u, v) + c(v, u)}{2\pi(u)} & u \neq v, \\ 1 & u = v \end{cases} \quad \text{Eq. 15}$$

[0060] This matrix is not symmetric. However, if another
 20 basis $\{\pi^{-1/2}(v)\delta_v\}_{v \in V}$ is chosen, then Δ can be represented as a symmetric matrix:

$$\Delta_{sm}(u, v) = \begin{cases} -\frac{c(u, v) + c(v, u)}{2\sqrt{\pi(u)\pi(v)}} & u \neq v, \\ 1 & u = v \end{cases} \quad \text{Eq. 16}$$

[0061] This matrix has been used to define Laplacian for directed graphs.

[0062] Now, learning on directed graphs using the above analysis is discussed. Given a directed graph $G=(V,E,w)$, and a discrete label set $L=\{-1,1\}$, the vertices in a subset $S\subset V$ have labels in L . The task is to predict the labels of those unclassified vertices in S^c , the complement of S . The present link spam detection problem can be cast into classification on a directed graph. For instance, FIG. 5 shows the vertices (or nodes) in the graph shown in FIG. 4 classified as spam or normal pages. The solid nodes are classified as normal pages while those shown in phantom are spam pages.

[0063] Define a function y with $y(v)=1$ or -1 if $v\in S$, and 0 if $v\in S^c$. For classifying those unclassified vertices in S^c , define a discrete regularization:

$$\arg \min_{\varphi\in H(V)} \left\{ \|\nabla\varphi\|_{H(E)}^2 + C\|\varphi - y\|_{H(V)}^2 \right\} \quad \text{Eq. 17}$$

[0064] where $C>0$ is the regularization parameter. In the objective function, the first term forces the classification function to be relatively smooth, and perhaps as smooth as possible and the second term forces the classification function to fit the given labels as well as possible.

[0065] When choosing the basis $\{\delta_v\}_{v\in V}$, Eq. 17 can be written as:

$$\arg \min_{\varphi\in H(V)} \left\{ \sum_{e\in E} \pi(e^-)p(e)(\varphi(e^+) - \varphi(e^-))^2 + C\sum_{v\in V} \pi(v)(\varphi(v) - y(v))^2 \right\} \quad \text{Eq. 18}$$

[0066] Again, the first term makes the function relatively smooth over all nodes while the second term forces the function to fit the labeled nodes to a desired closeness. If each function in $H(V)$ is scaled with a factor $\pi^{-1/2}$ (in other words, choose another basis $\{\pi^{-1/2}(v)\delta_v\}_{v\in V}$), then Eq. 18 will be transformed into:

$$\begin{aligned} \arg \min_{\phi \in H(V)} \left\{ \sum_{e \in E} \pi(e^-) p(e) \left(\frac{\phi(e^+)}{\sqrt{\pi(e^+)}} - \frac{\phi(e^-)}{\sqrt{\pi(e^-)}} \right)^2 \right. \\ \left. + C \sum_{v \in V} (\phi(v) - y(v))^2 \right\} \end{aligned} \tag{Eq. 19}$$

[0067] However, it can be seen that Eq. 18 is much more natural than Eq. 19.

[0068] A random walk over a given directed graph can be defined in many different ways. Three exemplary types of random walk used in spam detection are:

1. Following outlinks uniformly at random.

Formally, define a random walk with:

$$p(u, v) = \frac{w(u, v)}{d^+(u)} \tag{Eq. 20}$$

10

This is the one discussed above with respect to FIG. 3.

2. Following links uniformly at random regardless of directionality. Formally, define a random walk with:

$$p(u, v) = \frac{w(u, v) + w(v, u)}{d^+(u) + d^-(u)} \tag{Eq. 21}$$

15

3. Following inlinks uniformly at random.

Formally, define a random walk with:

$$p(u, v) = \frac{w(v, u)}{d^-(u)} \tag{Eq. 22}$$

[0069] Other choices of random walks can be used as well.

[0070] Assigning values to the nodes in directed graph basically requires selection of a random walk definition (transition probabilities) and solving Eq. 18 above for each of the nodes. This is set out above with respect to FIG. 2. Solving for Eq. 18 is set out more formally in pseudocode in Table 1 below for the random walk that inversely follows the links. To solve the optimization problem in Eq. 18, differentiate the objective function with respect to ϕ and then obtain:

25

$$\Delta_{am}\varphi + C(\varphi - y) = 0 \tag{Eq. 23}$$

[0071] where the first term on the left hand side is derived from Eq. 11 via the differential rule on inner products. The above equation can be written as:

$$(CI + \Delta_{am})\varphi = Cy \tag{Eq. 24}$$

[0072] where I is the identity matrix. This linear system has the closed-form solution:

$$\varphi = C(CI + \Delta_{am})^{-1}y \tag{Eq. 25}$$

[0073] although it may be more efficient to solve the linear system directly, rather than computing the inverse.

[0074] In the algorithm in Table 1 below, a parameter $\alpha \in]0,1[$ is used instead of $C \in]0,\infty[$. The relationship between α and C can be expressed as:

$$\alpha = \frac{1}{1+C} \tag{Eq. 26}$$

[0075] In the last step in Table 1, the classification is based on the sign of the function value on each vertex. As mentioned above with respect to FIG. 2, this is equivalent to setting the classification threshold to 0.

20 TABLE 1
TRANSDUCTIVE LINK SPAM DETECTION

Given a web graph $G=(V,E)$, some web pages $S \subset V$ have been manually labeled as content or spam. The graph is strongly connected. Otherwise, it is decomposed into strongly connected components. The remaining unclassified web pages in V may be classified as follows:

1. Define a random walk which chooses an inlink uniformly at random to follow. Formally, this random walk has the transition probabilities:

$$p(u,v) = \frac{w(v,u)}{d^-(u)},$$

for any u, v in V . Let π denote the vector which satisfies:

$$\sum_{u \in V} \pi(u) p(u, v) = \pi(v).$$

2. Denote by P the matrix with the elements $p(u, v)$, and Π the diagonal matrix with the diagonal elements being stationary probabilities $\pi(u)$ and zeros everywhere else. Form the matrix:

$$L = \Pi - \alpha \frac{\Pi P + P^T \Pi}{2}$$

where α is a parameter in $]0, 1[$.

3. Define a function y on V with $y(v) = 1$ or -1 if the web page v is labeled as content or spam, and 0 if v is unlabeled. Solve the linear system:

$$L\varphi = \Pi y,$$

and classify each unlabeled web page v as sign $\varphi(v)$.

[0076] FIG. 6 illustrates an example of a suitable computing system environment 300 on which embodiments may be implemented. The computing system environment 300 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the claimed subject matter. Neither should the computing environment 300 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 300.

[0077] Embodiments are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that

may be suitable for use with various embodiments include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, telephony systems, distributed computing environments that include any of the above systems or devices, and the like.

[0078] Embodiments may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Some embodiments are designed to be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules are located in both local and remote computer storage media including memory storage devices.

[0079] With reference to FIG. 6, an exemplary system for implementing some embodiments includes a general-purpose computing device in the form of a computer 310. Components of computer 310 may include, but are not limited to, a processing unit 320, a system memory 330, and a system bus 321 that couples various system components including the system memory to the processing unit 320. The system bus 321 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus,

and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0080] Computer 310 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 310 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 310. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0081] The system memory 330 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 331 and random access memory (RAM) 332. A basic input/output system 333 (BIOS), containing the
5 basic routines that help to transfer information between elements within computer 310, such as during start-up, is typically stored in ROM 331. RAM 332 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit
10 320. By way of example, and not limitation, FIG. 6 illustrates operating system 334, application programs 335, other program modules 336, and program data 337. Any part of system 100 can be in programs 335, modules 336, or anywhere else, as desired.

[0082] The computer 310 may also include other
15 removable/non-removable volatile/nonvolatile computer storage media. By way of example only, FIG. 6 illustrates a hard disk drive 341 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive
20 351 that reads from or writes to a removable, nonvolatile magnetic disk 352, and an optical disk drive 355 that reads from or writes to a removable, nonvolatile optical disk 356 such as a CD ROM or other optical media. Other
removable/non-removable, volatile/nonvolatile computer
25 storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 341 is typically connected to
30 the system bus 321 through a non-removable memory interface such as interface 340, and magnetic disk drive 351 and optical disk drive 355 are typically connected to the system bus 321 by a removable memory interface, such as interface 350.

[0083] The drives and their associated computer storage media discussed above and illustrated in FIG. 6, provide storage of computer readable instructions, data structures, program modules and other data for the computer 310. In
5 FIG. 6, for example, hard disk drive 341 is illustrated as storing operating system 344, application programs 345, other program modules 346, and program data 347. Note that these components can either be the same as or different from
10 operating system 334, application programs 335, other program modules 336, and program data 337. Operating system 344, application programs 345, other program modules 346, and program data 347 are given different numbers here to illustrate that, at a minimum, they are different copies.

[0084] A user may enter commands and information into the
15 computer 310 through input devices such as a keyboard 362, a microphone 363, and a pointing device 361, such as a mouse, trackball or touch pad. Other input devices (not shown) may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected
20 to the processing unit 320 through a user input interface 360 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 391 or other type of display device is also connected to the
25 system bus 321 via an interface, such as a video interface 390. In addition to the monitor, computers may also include other peripheral output devices such as speakers 397 and printer 396, which may be connected through an output peripheral interface 395.

30 [0085] The computer 310 is operated in a networked environment using logical connections to one or more remote computers, such as a remote computer 380. The remote computer 380 may be a personal computer, a hand-held device, a server, a router, a network PC, a peer device or other

common network node, and typically includes many or all of the elements described above relative to the computer 310. The logical connections depicted in FIG. 6 include a local area network (LAN) 371 and a wide area network (WAN) 373, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0086] When used in a LAN networking environment, the computer 310 is connected to the LAN 371 through a network interface or adapter 370. When used in a WAN networking environment, the computer 310 typically includes a modem 372 or other means for establishing communications over the WAN 373, such as the Internet. The modem 372, which may be internal or external, may be connected to the system bus 321 via the user input interface 360, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 310, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 6 illustrates remote application programs 385 as residing on remote computer 380. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0087] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

320757.03WO

WHAT IS CLAIMED IS:

1. A method of analyzing web pages, comprising:
accessing a plurality of web pages (108);
5 generating (150) a plurality of graphical representations
(110-114) of the web pages (108), each graphical
representation (110-114) having nodes (A-G) that represent
the web pages (108) and links between the nodes (A-G), the
links representing different relationships between the
10 nodes (A-G) in each graphical representation (110-114);
generating (156) a model (122) that models a random walk
through all of the graphical representations (110-114);
receiving training pages, wherein training nodes, in the
graphical representations (110-114), corresponding to the
15 training pages have a target function value indicative of
the training pages belonging to one of the groups (124-
128);
generating (158) a classifier based on the model (122),
based on classifier function values of nodes in the
20 graphical representations (110-114), and based on the
target function values of the training pages; and
grouping the web pages (108) into groups with the
classifier.

25 2. The method of claim 1 wherein generating a
classifier comprises:
selecting (158) a classifier function by optimizing a cost
function that penalizes differences between the target
function value and a classifier function value for the
30 training nodes.

320757.03WO

3. The method of claim 2 wherein generating a classifier comprises:

5 selecting (158) the classifier function by optimizing the cost function, wherein the cost function also penalizes differences between classifier function values calculated for different nodes (A-G) in the graphical representations (110-114).

4. The method of claim 1 wherein generating a plurality of graphical representations comprises:

10 generating a first graphical representation (110) having the links between the nodes (A-G) being representative of hyperlinks between the web pages (108).

5. The method of claim 4 wherein generating a plurality of graphical representations comprises:

15 generating a second graphical representation (112) having the links between the nodes (A-G) being representative of a similarity of the web pages (108).

20 6. The method of claim 5 wherein each of the links in the second graphical representation (112) are weighted by a weight indicating similarity between two web pages (108) connected by the links.

25 7. The method of claim 5 wherein generating a model comprises:

30 individually selecting (154) a random walk definition (116-120) defining a random walk for each of the graphical representations (110-114).

320757.03WO

8. The method of claim 7 wherein generating a model further comprises:

generating the model (122) to collectively model the random walks defined for each of the graphical representations
5 (110-114).

9. The method of claim 8 wherein generating a model comprises:

generating (156) a Markov mixture (122) of the random walks
10 defined for each of the graphical representations (110-114).

10. The method of claim 1 wherein grouping the web pages into groups comprises:

15 grouping (164) the web pages (108) into a first group (128) indicative of a spam web page and a second group (128) indicative of a content web page.

11. The method of claim 1 wherein grouping the web
20 pages into groups comprises:

grouping (162) the web pages (108) into groups (126) based on similarity of content.

12. The method of claim 1 wherein grouping the web
25 pages into groups comprises:

identifying (160) a community (124) of web pages (108) based on usage of the web pages (108).

13. A system for analyzing a collection of web pages
30 (108), comprising:

a graph generator (102) generating a plurality of graphs (110-114) representing the plurality of web pages (108),

320757.03WO

each graph (110-114) having a plurality of nodes (A-G) and links linking the nodes (A-G), each node representing a web page (108) in the collection and each link representing a relationship between web pages (108) linked by the link;

5 a random walk component (104) configured to generate a mixture model (122) modeling a collection of random walks performed on the plurality of graphs (110-114); and
a web page analysis component (106) configured to select an analysis function based on the mixture model (122), based
10 on how closely analysis function values for the nodes (A-G) conform to known values, and based on how much the analysis function values for the nodes (A-G) change over the graphs (110-114), the web page analysis component (106) being further configured to group the web pages (108) into groups
15 (124-128) based on the selected analysis function.

14. The system of claim 13 wherein the web page analysis component (106) groups the web pages (108) into groups that are likely to be viewed as groups in each graph
20 (110-114), given the random walk associated with each graph (110-114).

15. The system of claim 14 wherein the web page analysis component (106) groups the web pages (108) into
25 groups that are likely to be viewed as a group given all graphs (110-114), and all random walks defined for the graphs (110-114).

16. The system of claim 13 wherein the random walk component (104) generates the mixture (122) by following a
30 random walk defined for each graph (110-114).

320757.03WO

17. The system of claim 16 wherein the random walk component (104) generates the mixture (122) model by assigning stationary probabilities to the nodes (A-G) in each graph (110-114).

5

18. The system of claim 17 wherein the random walk component (104) generates the mixture model (122) as a Markov mixture of each of the random walks.

10 19. A method, implemented on a computer, of identifying groups of nodes in a collection of web pages, comprising:

generating (150) a plurality of directed graphs (110-114), each directed graph having, as its nodes (A-G), the web pages (108) linked by directed edges, each directed graph (110-114) having edges that represent a different relationship between nodes (A-G) connected by the edges than edges in other directed graphs (110-114);

15 20 defining (114) a random walk for each of the directed graphs (110-114), by defining transition probabilities for each pair of nodes (A-G);

performing (156) the random walks to generate stationary probabilities indicative of a probability of being on a given node (A-G) by selecting a starting node in a starting directed graph (110-114) and repeatedly selecting uniformly, at random, whether to follow an edge from the starting node (A-G) to another node (A-G) in the starting directed graph (110-114), or to a node (A-G) in another directed graph (110-114) or to jump, without following a link, to another node (A-G) in any of the directed graphs (110-114);

25
30

320757.03WO

identifying (158) a classifier function based on the stationary probabilities and based on classifier function values for nodes (A-G) satisfying a cost function that simultaneously considers differences between classifier function values for nodes (A-G) and training data and differences among classifier function values for the nodes (A-G); and
5 storing the groups (124-128) for use in a web page analysis system (106).

10

20. The method of claim 19 wherein performing the random walks comprises:

generating (156) a mixture model (122) modeling a mixture of the stationary and probabilities on each directed graph
15 (110-114).

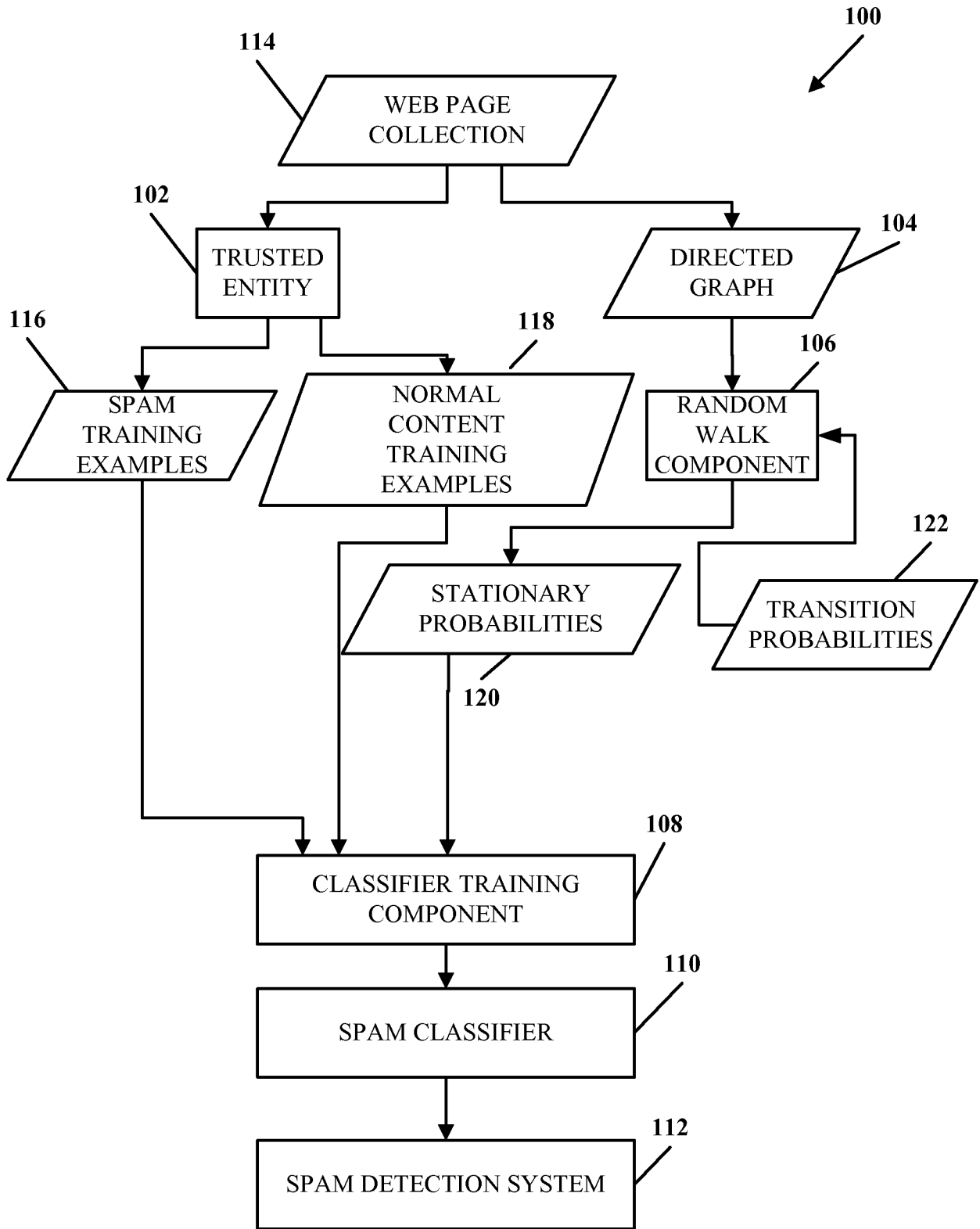


FIG. 1

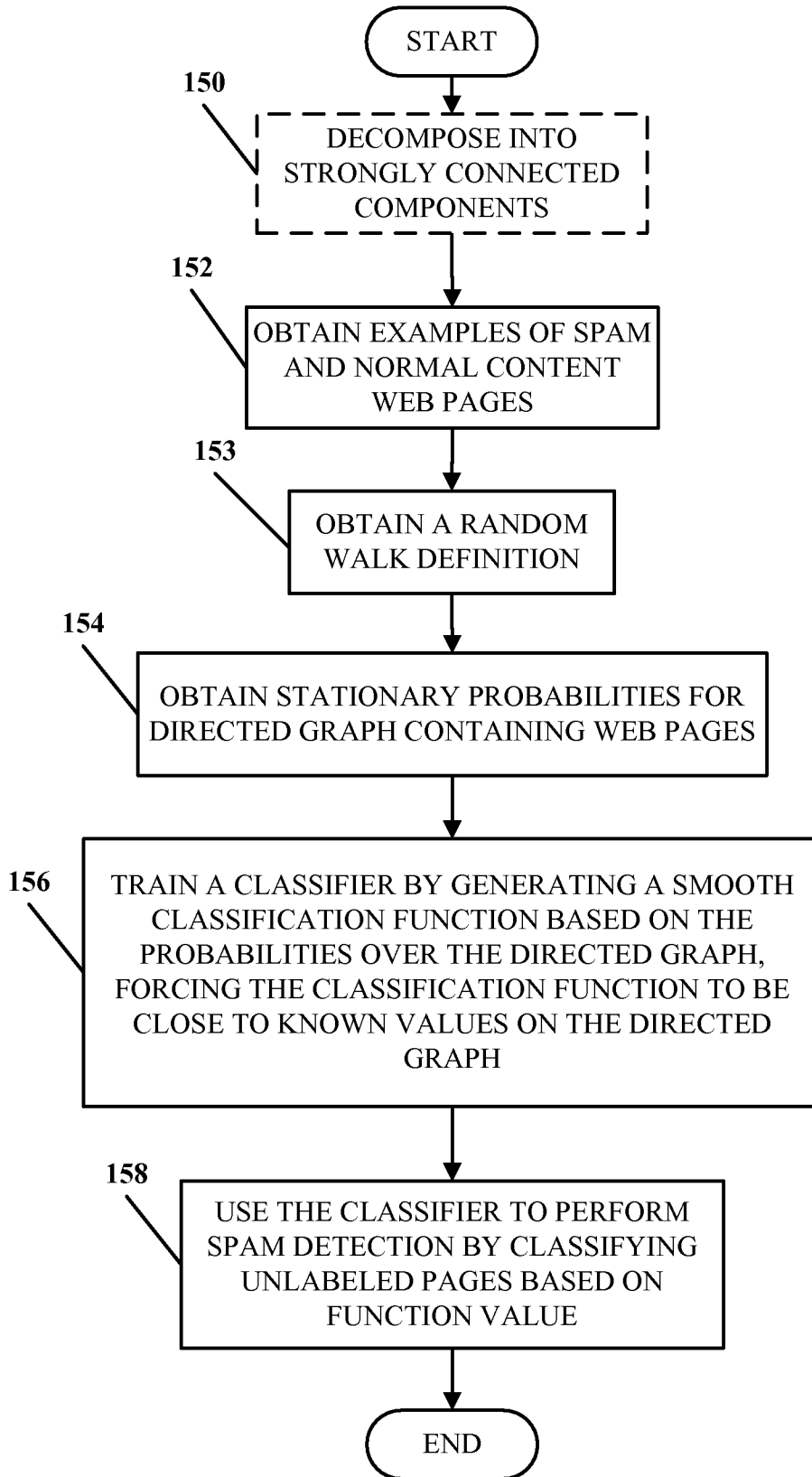


FIG. 2

3/5

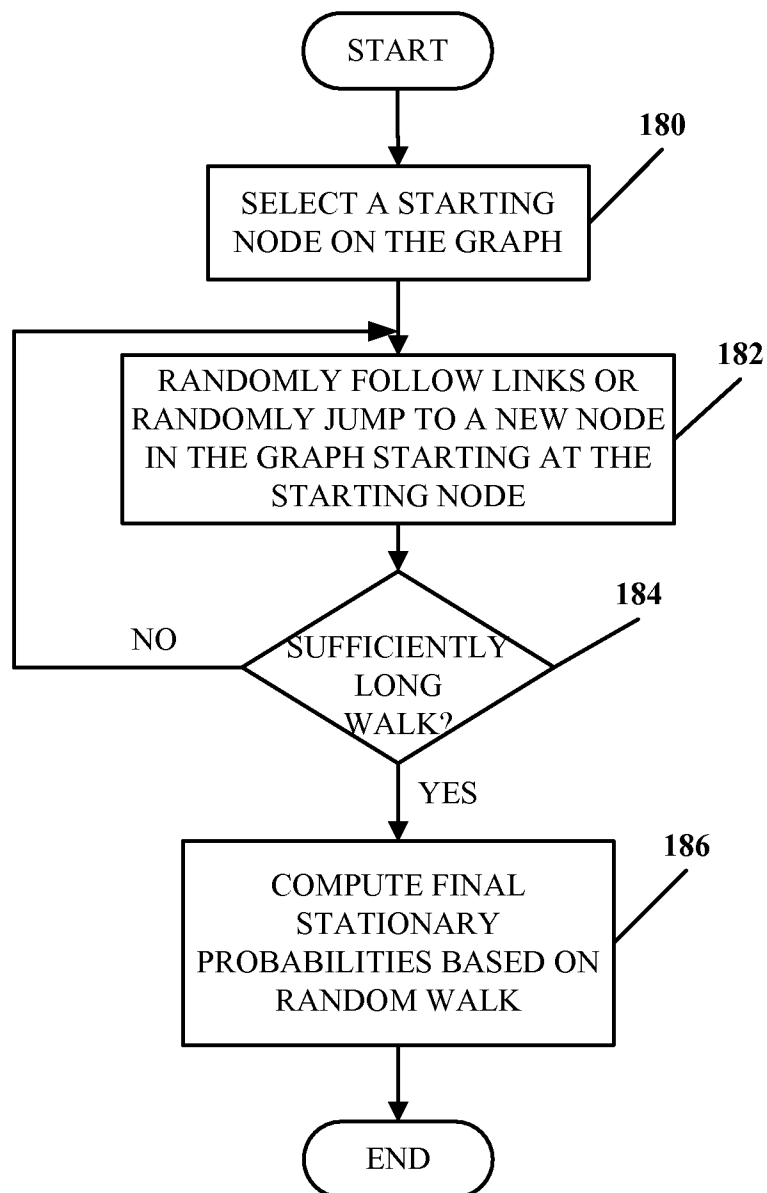


FIG. 3

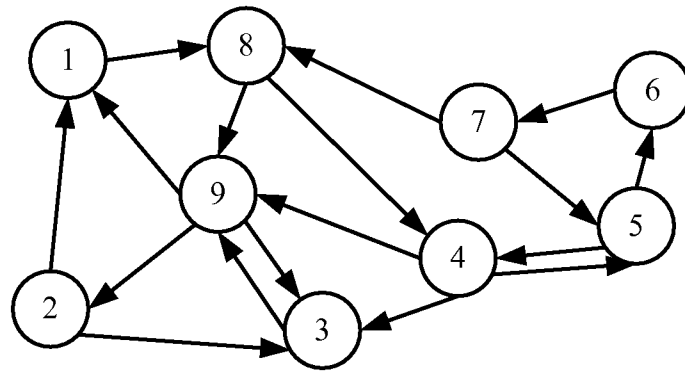


FIG. 4

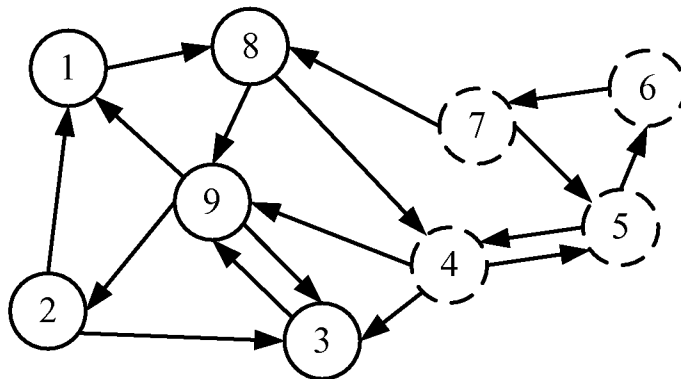


FIG. 5

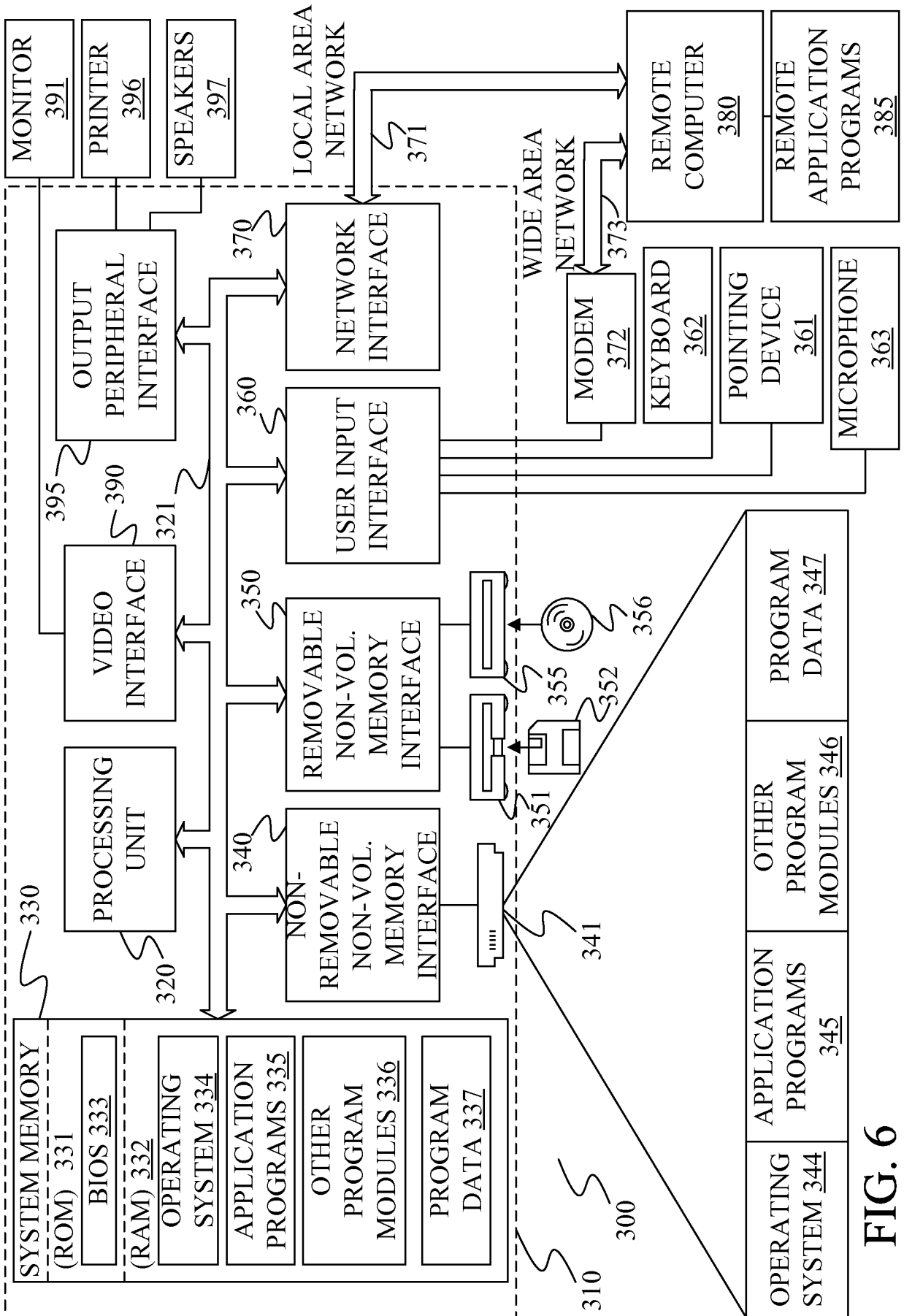




FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/061637

A. CLASSIFICATION OF SUBJECT MATTER		
<i>G06F 13/00(2006.01)i, H04L 12/66(2006.01)i</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) IPC 8 G06F, H04L		
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 since 1975 Japanese Utility models and applications for Utility models since 1975		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKIPASS (LIPO internal) & keywords: "web page, web site, spam, link, analyze, detection"		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 20060095416 A1 (PAVEL BARKHIN et al.) 04 May 2006 See the abstract; figures 2, 3A, 3B; paragraphs [0045]-[0080].	1 - 20
A	US 20050060297 A1 (MARC A. NAJORK) 17 Mar. 2005 See the abstract; figures 3A-3G; paragraphs [0058]-[0074].	1 - 20
A	US 20060069667 A1 (MARK STEVEN MANASSE et al.) 30 Mar. 2006 See the abstract; figure 3; paragraphs [0023]-[0024].	1 - 20
A	KR 0462292 B1 (NHN CO.) 17 Dec. 2004 See the abstract; figures 3A, 3B; claim 5.	1 - 20
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 26 AUGUST 2008 (26.08.2008)		Date of mailing of the international search report 26 AUGUST 2008 (26.08.2008)
Name and mailing address of the ISA/KR  Korean Intellectual Property Office Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140		Authorized officer BAE, Kyung Hwan Telephone No. 82-42-481-5768 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2008/061637

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 20060095416 A1	04.05.2006	EP 1817697 A2	15.08.2007
		KR 10-2007-0085477 A	27.08.2007
		WO 2006-049996 A2	11.05.2006
		WO 2006-049996 A3	27.09.2007
US 20050060297 A1	17.03.2005	AU 2004-205331 A1	07.04.2005
		BR 200403304 A	31.05.2005
		CA 2475328 A1	16.03.2005
		CN 1601532 A	30.03.2005
		EP 1517250 A1	23.03.2005
		JP 2005-092881 A	07.04.2005
		KR 10-2005-0027944 A	21.03.2005
		PA 04008383 A	31.03.2005
		RU 2004127646 A	20.02.2006
US 20060069667 A1	30.03.2006	CN 1770158 A	10.05.2006
		EP 1643392 A1	05.04.2006
		JP 2006-146882 A	08.06.2006
		KR 10-2006-0051939 A	19.05.2006
KR 0462292 B1	17.12.2004	JP 2007-524172 A	23.08.2007
		WO 2005-083593 A1	09.09.2005