Hybrid-Patent Classification Based on Patent-Network Analysis

Duen-Ren Liu and Meng-Jung Shih
Institute of Information Management, National Chiao Tung University, 1001 Ta Hseuh Road, Hsinchu 300, Taiwan. E-mail: dliu@iim.nctu.edu.tw

Effective patent management is essential for organizations to maintain their competitive advantage. The classification of patents is a critical part of patent management and industrial analysis. This study proposes a hybrid-patent-classification approach that combines a novel patent-network-based classification method with three conventional classification methods to analyze query patents and predict their classes. The novel patent network contains various types of nodes that represent different features extracted from patent documents. The nodes are connected based on the relationship metrics derived from the patent metadata. The proposed classification method predicts a query patent's class by analyzing all reachable nodes in the patent network and calculating their relevance to the query patent. It then classifies the query patent with a modified *k*-nearest neighbor classifier. To further improve the approach, we combine it with content-based, citation-based, and metadata-based classification methods to develop a hybrid- classification approach. We evaluate the performance of the hybrid approach on a test dataset of patent documents obtained from the U.S. Patent and Trademark Office, and compare its performance with that of the three conventional methods. The results demonstrate that the proposed patent-network-based approach yields more accurate class predictions than the patent network-based approach.

Introduction

Patents are valuable intellectual property and therefore require effective management to ensure that an organization maintains its competitive advantage (Guan & Gao, 2009; Su, Lai, Sharma, & Kuo, 2009). Because of developments in various technologies, the number of patents has increased rapidly in recent years. How to manage the constantly growing volume of patents is thus becoming an important issue. Patent classification is a key part of patent management; however, as the task is usually performed by patent analysts, categorizing new patent documents correctly is a laborious process. Hence, there is a pressing need for an effective patent-classification approach.

Basically, patent classification can be regarded as a text-categorization problem that involves assigning a patent document to a particular class. Most existing studies have considered information content to classify patent documents, and several classification algorithms have been developed based on different content features (e.g., He & Lo, 2008; Fall, Torcsvari, Benzineb, & Karetka, 2003, 2004; Kim & Choi, 2007; Larkey, 1999; Loh, He, & Shen, 2006; Trappey, Hsu, Trappey, & Lin, 2006). In addition, some approaches have utilized citation relationships to improve the performance of patent classification (Lai & Wu, 2005; Li, Chen, Zang, & Lie, 2007) while others have employed patent metadata (e.g., the inventor's name) to achieve improvements in the classification performance (Richter & MacFarlane, 2005).

Since patent metadata provides rich information that can be used to infer possible relationships between patent documents, there exists the potential to design effective patent-class prediction methods by utilizing patent metadata. To this end, we propose a novel patent-network-based classification method that utilizes patent metadata to construct a novel patent network for class prediction. The patent documents and metadata (e.g., the inventor and patent class) form, respectively, patent nodes and metadata nodes in the constructed network. In addition, the semantic relationships between the patent and metadata nodes are derived to link the nodes in the patent network. Based on the patent network, patent-network analysis is performed to identify the neighboring patents and metadata nodes of a query patent to predicting the patent's class. The concept of patent network analysis is based on social-network analysis (Alani, Dasmahapatra, O'Hara, & Shadbolt, 2003; O'Hara, Alani, & Shadbolt, 2002), which is used to determine the interactions between individuals in a social network. We adopt this concept in patent analysis by regarding patents as individuals in a patent "society," and propose a novel patent-network-based classification approach based on the patent network. The proposed approach involves two phases: (a) the patent

Received July 13, 2010; revised October 3, 2010; accepted October 4, 2010

 DOI: 10.1002/asi.21459

network construction phase, which identifies nodes and calculates the link weights based on the relationships provided by the patent metadata; and (b) the patent-class prediction phase, which predicts the class of a query patent by analyzing all reachable nodes in the patent network to calculate their relevance to the query patent and classifying the query patent with a modified k-nearest neighbor classifier.

Moreover, we propose a hybrid-patent-classification approach that combines a novel patent-network-based classification method with conventional content-based, citation-based, and metadata-based classification techniques to yield more accurate class predictions. Finally, we conduct experiments to assess the performance of the proposed approach with that of the conventional approaches on a real-world patent dataset. The experiment results show that the proposed patent-network-based classification method outperforms the aforementioned conventional patent-classification methods. The results also demonstrate that the proposed hybrid-classification approach yields more accurate class predictions than the patent network-based classification approach.

The remainder of this article is organized as follows. The next section contains a review of the literature on patent-classification methods and ontology-based network analysis. In the patent-network-based classification section, we present the patent-network-based classification methodology. In the hybrid-patent-classification section, we describe the proposed hybrid-patent-classification scheme, followed by the discussion of experiment results, and then our conclusions.

Literature Review

Patent-classification schemes classify patent documents. In recent years, a considerable number of such schemes have been proposed (e.g., He & Loh, 2010; He & Lo, 2008; Kim & Choi, 2007; Kohonen et al., 2000; Lai & Wu, 2005; Larkey, 1999; Richter & MacFarlane, 2005; Trappey et al., 2006). The features extracted from patent documents for classification purposes can be divided into three types: content features, citation information, and metadata.

Content-Based Patent Classification

Since patent classification is formulated as a text-categorization problem that involves assigning a patent document to the correct class, most studies have considered only patent content information to address the problem (e.g., Loh et al., 2006). In content-based patent-classification approaches, the content of a patent document d_p is represented by a vector of term weights, $\vec{d}_p = <w_{1p}, \ldots, w_{|T|p}>$, where T is the set of terms. The similarity of two patent documents is defined as the cosine value of their term vectors (Yang, 1994). The most popular term-weighting function is term frequency and inverse document frequency (*tfi–df*), developed by Salton and Buckley (1988). It is defined as follows:

$$tfidf(t_k, d_p) = tf(t_k, d_p) \times \log(N/n_{t_k}), \quad (1)$$

where $tf(t_k, d_p)$ denotes the number of times term t_k occurs in patent document d_p (the term frequency), and $\log(N/n_{tk})$ represents the total number of patent documents divided by those in which t_k occurs (the inverse document frequency).

The similarity of two patent documents is defined as the cosine value (Yang, 1994) of their respective term vectors, as shown in Equation 2:

$$Sim(p, q) = \frac{\vec{d}_p \cdot \vec{d}_q}{|\vec{d}_p||\vec{d}_q|}, \quad (2)$$

where q is the query-patent document to be classified, and p is a patent document in the training-patent dataset.

Based on the similarity of patent documents, the *kNN* classifier selects the k-nearest neighbors of a query patent to predict the class of the patent based on majority vote. The class that most of the neighboring patents belong to is taken as the class of the query patent. Instead of using the full text of a patent document as the basis for classification, some approaches classify patent documents by considering normative sections such as the abstract, background, and results (He & Lo, 2008; Fall et al., 2003, 2004; Kim & Choi, 2007; Larkey, 1999; Loh et al., 2006; Trappey et al., 2006). Several studies have regarded the patent document's abstract as the most informative feature (Larkey, 1999; Chen, Tokuda, & Adachi, 2003; Loh et al., 2006).

Citation-Based Patent Classification

In real-world applications, patent documents are linked through citations that imply the connections and relationships between the citer and the cited. Approaches that utilize citations have been proposed by Lai and Wu (2005) and Li et al. (2007). These studies have demonstrated that citation-based patent classification yields more accurate class predictions than the content-based classification approach. In our work, we also consider the citation relationships between patent documents when constructing the patent network.

The co-citation approach (Lai & Wu, 2005) classifies a query patent based on the majority vote of the classes of its cited patents. For example, suppose a query patent cites five documents in the basic patent set. If three of the cited patents belong to class $C1$ and the other two belong to class $C2$, the query patent will be assigned to class $C1$. Note that the co-citation approach uses the grouping result of patents, which are clustered according to the co-citation frequency and linkage strength of each pair of basic patents, as the classes rather than the well-known U.S. Patent Classification (UPC) codes or International Patent Classification (IPC) codes.

In the citation network-classification approach (Li et al., 2007), every patent has a citation network in which each cited node is labeled with its classification class. A patent's class is determined by evaluating the similarities between its citation networks and those of other patents already classified into UPC categories. The network similarity, or graph similarity, of two patents is calculated by comparing their random walk paths. This approach employs a three-stage,

DOI: 10.1002/asi

kernel-based technique for patent classification: data acquisition and parsing, kernel construction, and classifier training. Li et al. (2007) used support vector machine (*SVM*) as the kernel machine. In their approach, the kernel value (i.e., the patent similarity of a patent pair) is calculated as shown in Equation 3:

$$K(G_{pi}, G_{pj}) = \sum_h \sum_{h'} l(h, h') O(h|G) O(h'|G'), \quad (3)$$

where G_{pi} and G_{pj} represent the citation networks associated with two patents p_i and p_j, respectively; h and h' are the random walk paths in the respective graphs; $O(h|G)$ and $O(h'|G')$ denote the probability of random walk paths that exist in the citation networks. If h and h' are identical, $l(h,h') = 1$; otherwise, $l(h,h') = 0$.

For each class, the SVM classifier generates a classification model. The kernel matrix is an augmented matrix which contains the patent similarity vectors of all patents in the training set and their respective class labels. The class label of each patent is defined based on whether the patent belongs to a specific class. The label is 1 if the patent belongs to the specific class; otherwise, it is −1. This is the so-called *one-against-rest model* for the SVM, which is used to handle multiclass problems. For each specific class, a well-trained SVM model can be used to predict if a query patent belongs to the class. The final class is then determined by a applying a "winner-takes-all" strategy to the SVM models of all the classes.

Metadata-Based Patent Classification

The metadata in a patent document, such as the inventors' names and IPC codes, may correlate with the document's content and can be used for classification purposes. Richter and MacFarlane (2005) showed that patent classification based on a document's metadata can improve the accuracy of the results. Their approach uses metadata to help classify commercial intellectual property. Because the approach simultaneously considers text, inventor, and IPC metadata, it yields more accurate class predictions. Patent documents are mapped to vectors of important terms, inventors' names, and IPCs. For the text, the weights of terms are calculated by the *tf–idf* approach (Salton & Buckley, 1988); the weight of each inventor is calculated as $\sqrt{1/\#inv}$, where *#inv* is the total number of inventors of the patent; and the weight of each IPC code is calculated as $\sqrt{1/(\#ipc + 1)}$, where *#ipc* is the number of the IPC code assigned to the patent. Note that the primary IPC code is weighted twice as high as are other IPC codes assigned to the patent. After compiling the vectors, the similarity between two patent documents can be calculated. The *kNN* classifier is then used to identify the class of the query patent based on the similarity (cosine value) of the patent documents.

One limitation of the aforementioned method is that it works well only when the inventors of a query patent also exist in the training set. The method does not utilize indirect relationships to help classify patents developed by new inventors who are not included in the training set. In contrast, our method constructs a patent network; therefore, indirect relationships can be used to more flexibly and accurately classify patent documents.

Ontology-Based Network Analysis

A social network is a social structure made up of individuals (or organizations) connected by one or more specific types of interdependency (e.g., friendship) and common interest. The nodes are the individual actors in the network, and the connections (i.e., edges) are the relationships between the actors. Social networks have been used in various scenarios; for example, to examine how individuals interact with each other; to characterize the many informal connections that link executives, such as communities of practice (CoPs), which are groups of individuals interested in a particular job, procedure, or work domain; or to facilitate knowledge sharing (Alani et al., 2003; O'Hara et al., 2002; Yuan, Carboni, & Ehrlich, 2010).

O'Hara et al. (2002) developed an ontology-based network-analysis method to examine ontology-based social networks that help identify CoPs. The network is comprised of object instances (e.g., people, papers, or conferences) and the semantic relationships (e.g., author of, attend conference) between the instances. The rationale behind the method is that the relevance values of nodes increase with the number of semantic paths leading to the object of interest. The instances and their relationships in the ontology network are analyzed by a breadth-first, spreading-activation search algorithm that traverses the semantic relations between instances. In this approach, the relationships and their weights are selected manually and are predefined.

The purpose of social network analysis is to determine the interactions between a query node (e.g., a person) and the nodes (e.g., related persons) in a social network. Using a similar concept, we construct a patent network for patent-class prediction. Specifically, we modify O'Hara et al.'s (2002) ontology-based network-analysis method and use it in patent network analysis to measure the relevance of a query patent and the nodes in a patent network. The weights of relationships are generated automatically according to the semantic relevance of two nodes. Then, the k nodes with the highest relevance to the query patent are used to predict the class of the patent.

Ontology-based network analysis examines ontology-based social networks to identify CoPs through traversing the semantic relations between instances. A CoP in a social network represents a group of relevant object instances, such as people who share common interests or professions. The social network analysis has the advantages of discovering the implicit and indirect relations between instances through link traversals. Patents are often written to obfuscate the idea of the patent, thereby providing insufficient patent content and making patent classification difficult. Motivated by the advantages of network analysis and the CoP aspect, we employ patent network analysis to obtain more patent-relevant data by identifying a group of relevant

DOI: 10.1002/asi

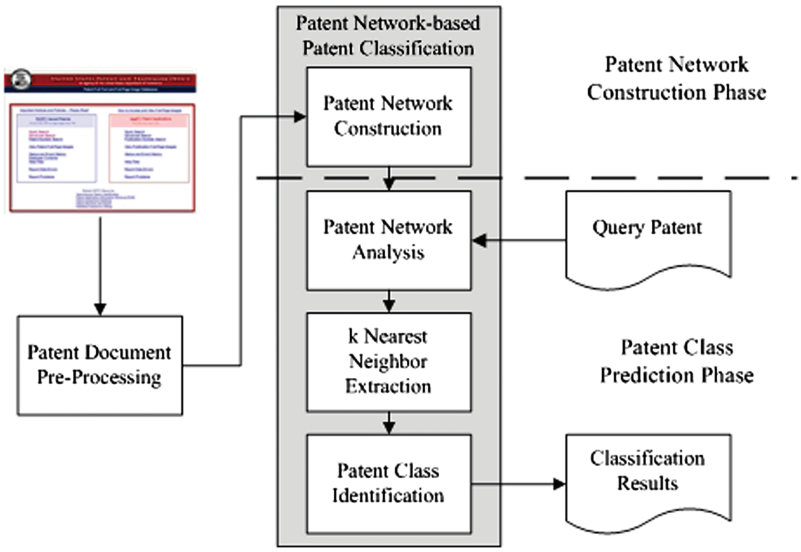

FIG. 1. The patent network-based classification process.

nodes, including patents, classes, assignees, and inventors, through discovering their implicit/indirect relations. Our proposed patent network-based classification approach utilizes the discovered patent-relevant data to compensate the insufficiency of patent content, and thus would yield more accurate class predictions.

Patent-Network-Based Classification

In this section, we introduce the proposed patent-network-based classification approach, as shown in Figure 1. The approach is implemented in two phases: (a) patent network construction and (b) patent class prediction, which includes patent network analysis, *k* nearest neighbor extraction, and patent-class identification.

Patent-Document Preprocessing

In this stage, we first collect patent documents from various sources on the Internet (e.g., the U.S. Patent and Trademark Office; USPTO). All the patent documents downloaded from the USPTO are in HTML format and semistructured. Therefore, we conduct data preprocessing to transform the raw patent document from the semistructured HTML format into a text format, filter out irrelevant content, and extract the required patent content. Previous studies have indicated that a patent's abstract is the most informative feature (Larkey, 1999; Chen et al., 2003; Loh et al., 2006). Thus, we extract the content features from the titles and abstracts of the patent documents. The processing of content features includes the removal of stop words and the extraction of *tf–idf* weighting for each term by the *tf–idf* approach (Salton & Buckley, 1988). We also extract the following information from the original documents for further analysis: the patent number, the UPC code, inventor and assignee names, and citation data.

Patent Network Construction

The first step of the patent-network-based classification process involves building a patent network, as shown in Figure 2. The relations between instances (nodes) are identified to construct the network. The weights of all the relationships among nodes are derived by the functions described in this section. Relationships (connections) of zero degree are dropped, and the network is trimmed to form the final patent network for classification. The proposed patent network contains four types of instances (nodes) and eight types of relations (edges). The node types are patent, UPC class, inventor, and assignee (e.g., a research institute). The weights of the relationships are calculated by the functions listed in Table 1.

$R_{PP}(p_1, p_2)$ denotes the relationship between two patents p_1 and p_2. Both citations and co-citations are considered active relations between two patents, as shown in Equation 4:

$$R_{pp}(p_1, p_2) = w_{cite} \times Cite(p_1, p_2) + w_{cocite} \times CoCite(p_1, p_2), \quad (4)$$

where $Cite(p_1, p_2)$ is the citation relation between p_1 and p_2; $CoCite(p_1, p_2)$ is the degree of co-citing between p_1 and p_2; and $w_{cite} + w_{cocite} = 1$. If the citation exists between p_1 and p_2 (either p_1 cites p_2 or p_2 cites p_1), $Cite(p_1, p_2) = 1$; otherwise, $Cite(p_1, p_2) = 0$. $CoCite(p_1, p_2) = |\ CitedBy(p_1) \cap CitedBy(p_2)| \ / \ |CitedBy(p_1) \cup CitedBy(p_2)|$, where $CitedBy(p_1)$ and $CitedBy(p_2)$ are the sets of patents cited by p_1 and p_2, respectively;.

$R_{II}(v_1, v_2)$ represents the ratio of patents that belong to two inventors v_1 and v_2, and is defined as Equation 5.

$$R_{II}(v_1, v_2) = \frac{|Patents(v_1) \cap Patents(v_2)|}{|Patents(v_1) \cup Patents(v_2)|}, \quad (5)$$

where $Patents(v_1)$ and $Patents(v_2)$ are the sets of patents belonging to v_1 and v_2, respectively.

DOI: 10.1002/asi

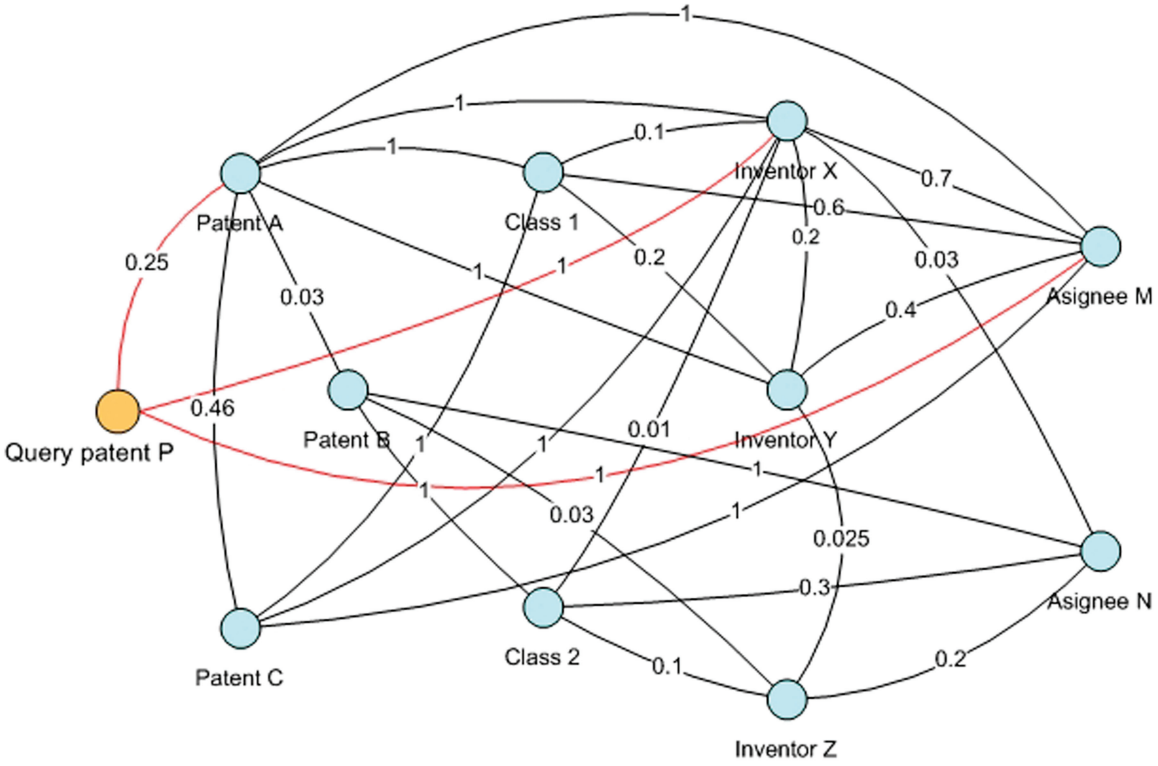

FIG. 2. An example of a patent network.

TABLE 1. The relationship metric in the patent network.

Relationship weights	Patent p_2	Class c_2	Inventor v_2	Assignee a
Patent p_1	$R_{PP}(p_1, p_2)$	$R_{PC} = \begin{cases} 1 : p_1 \in c_2 \\ 0 : p_1 \notin c_2 \end{cases}$	$R_{PI} = \begin{cases} 1 : p_1 \text{ invented by } v_2 \\ 0 : \text{not related} \end{cases}$	$R_{PA} = \begin{cases} 1 : p_1 \text{ belonging to } a \\ 0 : \text{not related} \end{cases}$
Class c_1		n.a.	$R_{CI}(v_2, c_1)$	$R_{CA}(c_1, a)$
Inventor: v_1			$R_{II}(v_1, v_2)$	$R_{IA} = \begin{cases} 1 : v_1 \text{ employed by } a \\ 0 : \text{not related} \end{cases}$

R_{CI} (v_2, c_1) represents the ratio of patents belonging to a specific inventor v_2 to the number of patents in a patent class c_1, and is defined as Equation 6.

$$R_{CI}(v_2, c_1) = \frac{|Patents(v_2) \cap Patents(c_1)|}{|Patents(c_1)|}, \qquad (6)$$

where $Patents(c_1)$ is the set of patents belonging to class c_1.

$R_{CA}(c_1,a)$ represents the importance and maturity of a technology of assignee a in a specific technology field (i.e., class c_1), as shown in Equation 7:

$$R_{CA}(c_1, a) = \frac{\sum_{p_i \in Patents(a) \cap Patents(c_1)} NumCitations(p_i, a, c_1)}{\sum_{p_j \in Patents(c_1)} NumCitations(p_j, c_1)}, \qquad (7)$$

where $NumCitations(p_i, a, c_1)$ is the number of patents in class c_1 that cite assignee a's patent p_i; and $NumCitations(p_j, c_1)$ is the number of patents in class c_1 that cite patent p_j.

Figure 2 shows a patent network that contains the four types of nodes: patent, class, inventor, and assignee. The weights of the relations are calculated by the equations listed in Table 1. The patent network is a base map for classifying unclassified patents. In the next subsection, we describe the classification process based on patent-network analysis. Classifying a patent to the most suitable class involves three steps: patent network analysis, k-nearest neighbor extraction, and patent-class identification.

Patent Network Analysis

To classify a patent document, we first search the patent network to find patent nodes, inventor nodes, and assignee nodes that have connections with the query patent. For example, in the network in Figure 2, X is the inventor of query patent P, and the assignee is M. Patent P also has citation relationships with other patents. These connections are therefore evaluated to derive their respective weights using the equations listed in Table 1.

After determining all the connections and weights between the query patent and the nodes in the patent network, we calculate the relevance of the query patent to each node in the

DOI: 10.1002/asi

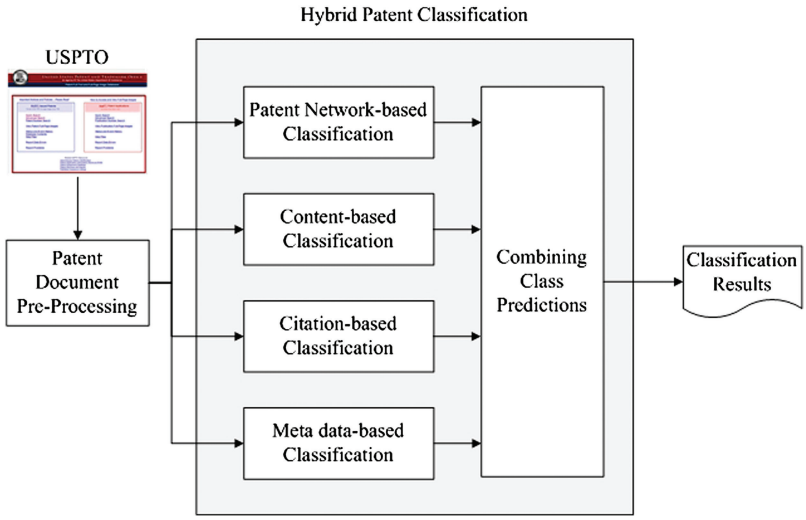

FIG. 3. The hybrid patent classification approach.

network. The algorithm used for patent network analysis is a modification of the ontology-based network-analysis algorithm developed by O'Hara et al. (2002) for identifying an individual's CoP. Our algorithm calculates the weights of the nodes and their relations to derive their relevance scores to the query patent. More specifically, it implements a breadth-first, spreading-activation search strategy and traverses the relations between the nodes until it reaches a link threshold, which is the maximum number of consecutive links between nodes that can be traversed. The steps of the patent network analysis algorithm are detailed in the Appendix.

K-Nearest Neighbor Extraction

After calculating the relevance of the query patent document to the nodes in the patent network, the k nodes with highest relevance scores to the query patent document are extracted and used to identify the most appropriate class for a patent.

Patent-Class Identification

Let S_q be the set of neighboring nodes identified in the k-nearest neighbor extraction step. In this step, the nodes in S_q are used to determine the class of the query patent q. Unlike the classical *kNN* method, which can find only neighboring nodes of the same type, the proposed method can find k nodes of various types by using the result of patent network analysis. We only use patent and class nodes to calculate the scores of candidate classes because they are more suitable for interpreting patent classes. For "patent" nodes, the more relevant a patent node p is to the query patent, the greater the likelihood that the query patent belongs to the class of that patent node. In addition, for "class" nodes, the more relevant a class node c is to the query patent, the greater the likelihood that the query patent belongs to the class of that node. We denote the set of identified neighboring patent nodes and the set of identified neighboring class nodes as S_q^P and S_q^C, respectively. Note that S_q^P and $S_q^C \subset S_q$.

The next step evaluates the predicted scores of candidate classes, which are selected from the identified patent nodes and class nodes. The predicted score $F_{q,c}^{PNW}$ for a given query patent q belonging to class c is calculated by Equation 8:

$$F_{q,c}^{PNW} = \sum_{d \in S_q^P} w_d^{PNW} B_{d,c}^P + \sum_{d \in S_q^C} w_d^{PNW} B_{d,c}^C, \quad (8)$$

where w_d^{PNW} denotes the weight; that is, the relevance score of node d obtained by patent network analysis. If node d represents a patent belonging to class c, $B_{d,c}^P = 1$; otherwise, $B_{d,c}^P = 0$. If node d represents a class c, $B_{d,c}^C = 1$; otherwise, $B_{d,c}^C = 0$. After obtaining all the predicted scores of candidate classes, the class with the highest score is taken as the class of the query patent.

Hybrid-Patent Classification

In this section, we propose a hybrid approach that utilizes patent metadata and considers the semantic structure of the patent network. The approach involves two phases: implementing different patent-classification approaches and combining class predictions, as shown in Figure 3.

Patent Classification by Various Methods

In this phase, patent documents are classified by four methods: content-based patent classification, citation-based patent classification, metadata-based patent classification, and patent network-based classification. Next, we describe how the four methods are applied.

Content-based patent classification. Previous studies have posited that a patent's abstract is the most informative feature (Larkey, 1999; Chen et al., 2003; Loh et al., 2006). Thus, we extract the content features from the titles and abstracts of the patent documents in this work. The steps of the content-based approach were described earlier. After determining the similarity between the query patent and patents in

DOI: 10.1002/asi

the training-patent dataset, the k nodes with the highest similarity to the query patent document are extracted and used to identify the most appropriate class for the patent. Under the content-based classification method, for a given query patent q, $F_{q,c}^{content}$ denotes the prediction score of a query patent q belonging to class c. We choose the k-nearest neighbor patents, S_q^{Nbr}, as the references to calculate the prediction score, as shown in Equation 9:

$$F_{q,c}^{content} = \sum_{p \in S_q^{Nbr}} \frac{B_{p,c}}{\left|S_q^{Nbr}\right|},$$
$$\text{where } B_{p,c} = \begin{cases} 1, & \text{if patent } p \text{ belongs to class } c \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Citation-based patent classification. Citation-based patent-classification approaches include co-citation patent classification (Lai & Wu, 2005) and citation network patent classification (Li et al., 2007). The co-citation approach determines the class of a query patent by majority vote of the classes of its cited patents, as described in earlier. For a given query patent q, let $F_{q,c}^{cocitation}$ denote the prediction score of a query patent q belonging to class c under a citation-based classification method. The cited patents of q, S_q^{cite}, are taken as references for calculating the prediction score, as shown in Equation 10:

$$F_{q,c}^{cocitation} = \sum_{p \in S_q^{cite}} \frac{B_{p,c}}{|S_q^{cite}|}. \quad (10)$$

The steps of the citation-network approach (Li et al., 2007) were detailed earlier. This approach enables us to retrieve two levels of cited patents from each patent document to construct the citation network and train the classifier. The retrieved citation network of the set contains 25,348 patents in a citation network with 74 categories. Under the citation-network classification method, for a given query patent q, $F_{q,c}^{citeNW}$ denotes the prediction score of query patent q belonging to class c, as defined in Equation 11.

$$F_{q,c}^{citeNW} = SVM(q, sim_q, c), \quad (11)$$

where sim_q denotes the vector of patent similarity between q and patents in the training set; $sim_q = [K(G_1,G_q), K(G_2, G_q), \ldots, K(G_z,G_q)]$; and z is the number of patents in the training set. Note that G_{pi} and G_{pj} represent the citation networks associated with two patents p_i and p_j, respectively. $K(G_{pi},G_{pj})$ denotes their patent similarity (by Equation 3); and $SVM(q,sim_q, c)$ is the output of the SVM classifier for classifying q as belonging to class c.

Metadata-based patent classification. Richter and MacFarlane (2005) used metadata (e.g., inventors' names) to facilitate classification, as described earlier. In this study, every patent document is represented by a vector of terms and inventors. After constructing the vectors, the similarity of two patent documents is calculated, and the *kNN* classifier is used to identify the appropriate class for the query patent based on the similarity (cosine value) of the patent documents. Under the metadata-based classification method, for a given query patent q, $F_{q,c}^{metadata}$ denotes the prediction score of a query patent q belonging to class c. We choose the k-nearest neighbor patents, S_q^{Nbr}, as references to calculate the prediction score $F_{q,c}^{metadata}$, as shown in Equation 12.

$$F_{q,c}^{metadata} = \sum_{p \in S_q^{Nbr}} \frac{B_{p,c}}{|S_q^{Nbr}|}. \quad (12)$$

Patent-network-based classification. The proposed patent-network-based approach constructs a patent network by using the metadata of classified patents to represent the relationships among various field elements of the metadata. A query patent document can then be classified by searching for the "nearest" nodes in the patent network, ranking them by their relevance scores, and then predicting the most appropriate class for the query patent. The approach involves four steps: patent network construction, patent network analysis, k-nearest neighbor extraction, and patent-class identification, as described earlier. The predicted score $F_{q,c}^{PNM}$ for a given query patent q belonging to class c is calculated by Equation 8.

Combination of Multiple Class Predictions

Under the proposed hybrid approach, each method generates a classification result based on the scores of the query patent in all candidate classes. The results generated by the four methods are then combined to yield the final patent classes as the output of this phase. Let $F_{q,c}^{citation}$ denote the prediction score of the citation-based patent classification, including the co-citation approach (Equation 10) and the citation-network approach (Equation 11). The joint result, $F_{q,c}$ is generated by the linear combination of $F_{q,c}^{content}$, $F_{q,c}^{citation}$, $F_{q,c}^{metadata}$, and $F_{q,c}^{PNM}$, as shown in Equation 13:

$$F_{q,c} = \alpha \times F_{q,c}^{content} + \beta \times F_{q,c}^{citation} + \gamma \times F_{q,c}^{metadata} + \delta \times F_{q,c}^{PNW}, \quad (13)$$

where α, β, γ, and δ are the respective weights of the four classification methods. The weights are determined empirically based on the most accurate class prediction in experiments. The class with the highest prediction score is then taken as the class of the query patent.

Experiments

To evaluate the proposed approach, we conducted experiments on the collection of patent documents obtained from the USPTO. We use a patent's UPC to denote its class. We selected five classes (i.e., UPCs) to distinguish the classification effect, and randomly selected patent documents from each selected class. Some selected patent documents have missing field values, and thus were deleted from the dataset. The final dataset contained 1,231 patent documents divided into five UPCs, as shown in Table 2. The documents in the database records were divided into two sets: (a) a training set (70% of the collected dataset) containing the patent

DOI: 10.1002/asi

TABLE 2. The U.S. Patent and Trademark Office patent dataset.

Class no.	Class title	Data instances
29	Metal Working	246
257	Active Solid-State Devices	273
324	Electricity: Measuring and Testing	221
438	Semiconductor Device Manufacturing Process	286
709	Electrical Computers and Digital Processing Systems: Multicomputer Data Transferring	205

documents whose classes were known and (b) a test set (30% of the collected dataset) containing patent documents whose classes were to be determined.

We used four standard classification-performance metrics—*accuracy*, *precision*, *recall*, and *F-measure* (Salton & Buckley, 1988; van Rijsbergen, 1979)—to evaluate the performance of the classifiers. The metrics have been widely used in information retrieval and machine learning studies. Classification accuracy was used to assess the overall performance, as shown in Equation 14:

$$\text{Accuracy} = \frac{\text{\# of correctly classified patents}}{\text{total \# of patents}}. \quad (14)$$

Precision, recall, and F-measure were used to assess the classification performance. For instances of class i:

$$\text{Precision}(i) = \frac{\text{\# of correctly identified patents for class } i}{\text{total \# of patents identified as class } i} \quad (15)$$

$$\text{Recall}(i) = \frac{\text{\# of correctly identified patents for class } i}{\text{total \# of patents in class } i}. \quad (16)$$

Finally, to obtain a single performance measure, we used a simple F-measure to balance the precision and recall scores, *a*s shown in Equation 17:

$$\text{F-measure}(i) = \frac{2 \times \text{precision}(i) \times \text{recall}(i)}{\text{precision}(i) + \text{recall}(i)}. \quad (17)$$

Precision and recall evaluate whether a classification is successful. If both parameters yield high scores in a classification experiment, the approach's performance is considered ideal. However, precision and recall usually conflict with each other, so the F-measure is used to balance the two results.

Experiments on Patent-Network-Based Classification

Link threshold of relevance calculation. The *k*-nearest neighbor extraction step attempts to identify the nodes that are most similar to the query patent document within the boundary defined by the given link threshold. The number of links used to expand the patent network has a significant effect on the results. If we limit expansion to only one link, all identified nodes have a direct relation to the query patent document. However, as the number of links increases, the number of nodes that have an indirect link to the query patent also will increase. Table 3 shows the performance of the patent-network-based classification module under different link thresholds. The most accurate class prediction is achieved when the link threshold = 3. Hence, we set the link threshold = 3 in the following experiments. Moreover, the k nodes with highest relevance scores to the query patent document are extracted and used to identify the most appropriate class for a target patent. The k value is determined empirically from the experiments, and we set $k = 10$.

TABLE 3. The performance of the patent-network-based classification under different link thresholds.

Link threshold	Accuracy	Precision	Recall	F-measure
1	33.2	31.4	31.8	31.6
2	57.6	58.1	55.4	56.7
3	74.9	77.6	74.9	76.2
4	67.8	66.3	64.7	65.5

TABLE 4. The performance of the patent network with different combinations of nodes.

Node types	Accuracy	Precision	Recall	F-measure
Patent/class/inventor	61.9	68.8	65.3	67.0
Patent/class/assignee	68.5	66.1	71.4	68.6
Patent/class/inventor/assignee	74.9	77.6	74.9	76.2

Types of Nodes in the Patent Network (link threshold = 3). The types of nodes for patent network analysis also affect the results. We tried various combinations of the types of nodes via experiments. As shown in Table 4, patent network analysis using with four types of nodes (patent, class, inventor, and assignee nodes) yields the most accurate class prediction.

Comparison of the Patent-Network-Based Approach With Other Methods

We compare four patent-classification methods: content-based, citation-based, metadata-based, and the proposed patent-network-based classification methods. The content-based method uses the similarity of content (title and abstract), and adopts the *kNN* classifier to predict the class of a query patent based on the similarity measures of the patents. The co-citation approach determines the class of a query patent by majority vote of the classes of its cited patents. The citation-network approach uses the patent similarity in the citation network and employs an SVM classifier to predict the class of a query patent. We retrieve two levels of cited patents from each patent document to construct the citation network. The retrieved citation network of the set contains 25,348 patents. For the metadata-based approach, the neighbors are chosen based on the similarities of the content (title and abstract), inventor, and IPC. This approach also uses the *kNN* classifier to predict the class of a query patent. Note that our proposed patent-network-based approach uses the

DOI: 10.1002/asi

TABLE 5. Comparison of the paten- network-based approach with other methods.

Patent-classification methods	Accuracy	Precision	Recall	F-measure	p
Patent-network-based approach	74.9	77.6	74.9	76.2	
Content-based (title + abstract)	45.2	47.8	45.4	46.6	0.00000***
Citation-based (co-citation)	57.6	54.2	62.8	58.2	0.00000***
Citation-based (citation network)	69.5	71.4	73.5	72.4	0.00994**
Metadata-based (text + inventor + IPC)	71.3	75.6	68.7	72.0	0.10464
Metadata-based* (text + inventor)	52.6	71.6	56.5	63.2	0.00000***

* $p < 0.05$. ** $p < 0.01$. *** $p < 0.001$.

relevance of nodes in the patent network. A particular feature of the *kNN* classifier applied in our proposed patent-network-based approach is that the neighbors can be of different types such as patents and classes whereas the other three methods only search for neighbors among patents.

Table 5 shows the performances of the compared patent-classification approaches. The proposed patent-network-based approach achieves the best performance in terms of accuracy (74.9%) and the F-score (76.2%). The second-best approach, the metadata-based approach, considers the IPC codes when deciding the class of a query patent. The IPC code denotes a kind of classification and may correlate with the UPC code, which represents the class of a patent. Thus, it is not reasonable to consider the IPC codes when making UPC predictions of class. The metadata-based (text + inventor + IPC) method may be affected by the correlation between the IPC and the UPC and thus yields a good result. Accordingly, we also compared the metadata-based approach without considering the IPC codes. The citation-network approach yields more accurate class predictions than does the metadata-based (text + inventor) method.

We performed pairwise and one-tailed t tests to examine the significance differences between the patent-network-based methods and the traditional methods. T tests were conducted by using the prediction results from different methods, where *1* represents a successful prediction and *0* represents a false prediction. The p values for comparing the patent-network-based approach with other classification methods are listed in Table 5. The results show that the differences are statistically significant at the 0.01 or 0.001 level, except for the comparison with the metadata-based (text + inventor + IPC) method. It is clear that the proposed patent-network-based approach yields more accurate class predictions than the content-based, co-citation, citation-network, and modified metadata-based methods. The difference between the patent-network-based method and the metadata-based (text + inventor + IPC) method is not significant. The metadata-based (text + inventor + IPC) method may be affected by the correlation between the IPC and the UPC and thus yields a good result.

Experiments on Hybrid-Patent Classification

In the proposed hybrid approach, each method generates a classification result, and the joint result is derived by linear combination, as shown in Equation 13. The parameters α, β, γ, and δ are the respective weights of the classification methods, which are determined empirically based on the most accurate class prediction in experiments. We chose the citation-network method as the citation-based part of the proposed hybrid approach because it outperforms the co-citation method in the experiments. The metadata-based (text + inventor) method is used as the metadata-based part of the hybrid approach.

Table 6 shows the combinations of different patent-classification methods and their weights. The goal of this experiment is to determine which combination of the content-, citation-, metadata-, and patent-network-based methods yields the most accurate class prediction. The weights are determined according to the best class-prediction quality (e.g., accuracy or F-measure) that can be achieved under different combinations of weight assignments. To find the best weight combination of the hybrid approach, which combines four patent-classification methods, we tested various combinations of the α, β, γ, and δ parameters by enumerating their values systematically in increments of 0.1 ranging from 0 to 1. The best class-prediction quality (accuracy: 84,1% and F-measure: 86.4%) of the proposed hybrid approach is achieved when $(\alpha, \beta, \gamma, \delta) = (0.1, 0.3, 0.1,$ and $0.5)$. Thus, we use these weights as the weight ratios of the hybrid approach.

Table 6 also shows different combinations of patent-classification methods, including the combinations of two or three patent-classification methods. Similarly, the best weight setting of each of the combined approaches is determined by systematically adjusting the weight values in increments of 0.1. For example, the best class-prediction quality of combining the content-based and citation-network methods is achieved when $(\alpha, \beta, \gamma, \delta) = (0.2, 0.8, 0,$ and $0)$. The result shows that the combination of all four methods achieves the best performance in terms of accuracy (84.1%) and the F-measure (86.4%). The weights of the four methods are 0.1, 0.3, 0.1, and 0.5, respectively. In terms of the hybrid effect, the results show that the patent-network-based method (with the highest weight of 0.5) enhances the classification performance the most; and the citation network method (with weight of 0.3) is more effective than are the content-based and metadata-based methods.

Table 7 shows the performances of the proposed hybrid approach and other patent-classification methods. The proposed hybrid approach with weights $\alpha = 0.1$, $\beta = 0.3$, $\gamma = 0.1$, and $\delta = 0.5$ achieves the best performance in

DOI: 10.1002/asi

TABLE 6. The results of experiments using different combinations of patent-classification approaches.

Hybrid-patent classification	α	β	γ	δ	Accuracy	F-measure
$C_{ontent} + C_{itationNW} + M_{etadata}{}^{*} + P_{atentNW}$	0.1	0.3	0.1	0.5	84.1	86.4
$C_{ontent} + C_{itationNW} + M_{etadata}{}^{*}$	0.1	0.6	0.3	0	73.8	75.4
$C_{ontent} + C_{itationNW} + P_{atentNW}$	0.1	0.3	0	0.6	78.4	80.3
$C_{ontent} + M_{etadata}{}^{*} + P_{atentNW}$	0.1	0	0.2	0.7	77.0	78.8
$C_{itation} + M_{etadata}{}^{*} + P_{atentNW}$	0	0.3	0.2	0.5	83.2	86.2
$C_{ontent} + C_{itationNW}$	0.2	0.8	0	0	71.9	74.2
$C_{ontent} + M_{etadata}{}^{*}$	0.1	0.9	0	0	53.0	63.5
$C_{ontent} + P_{atentNW}$	0.1	0	0	0.9	75.5	78.5
$C_{itation} + M_{etadata}{}^{*}$	0	0.7	0.3	0	73.5	75.2
$C_{itation} + P_{atentNW}$	0	0.4	0	0.6	76.4	79.4
$M_{etadata}{}^{*} + P_{atentNW}$	0	0	0.2	0.8	76.5	78.7

TABLE 7. Comparison of the hybrid approach with different patent-classification methods.

Patent-classification methods	Accuracy	Precision	Recall	F-measure	p
Hybrid ($C_{ontent} + C_{itationNW} + M_{etadata}{}^{*} + P_{atentNW}$)	84.1	85.2	87.7	86.4	
Content-based (Title + Abstract)	45.2	47.8	45.4	46.6	0.00000***
Citation-based (Co-citation)	57.6	54.2	62.8	58.2	0.00000***
Citation-based (citation network)	69.5	71.4	73.5	72.4	0.00000***
Metadata-based (text + inventor + IPC)	71.3	75.6	68.7	72.0	0.00000***
Metadata-based* (text + inventor)	52.6	71.6	56.5	63.2	0.00000***
Patent Network-based Approach	74.9	77.6	74.9	76.2	0.00000***

* $p < 0.05$. ** $p < 0.01$. *** $p < 0.001$.

terms of accuracy (84.1%) and the F-measure (86.4%). The second-best approach is the proposed patent-network-based method. The content-based method yields less accurate class predictions than other methods. We also conducted pairwise and one-tailed *t*-tests to examine the differences in performance between the proposed hybrid approach and other methods. T-he *p* values for comparing the proposed hybrid approach with other classification methods are listed in Table 7. The results show that the differences are statistically significant at the .001 level. From these results, it is clear that our proposed hybrid approach yields more accurate class predictions than the other classification methods.

Conclusion

In this article, we have proposed a novel patent-network-based classification method that uses patent metadata to derive the weights of the relationships between different types of nodes in a patent network. Based on patent-network analysis, the classification result can be improved by considering the neighboring patent nodes and class nodes of a query patent when making class predictions. The contributions of the proposed method include novel designs for (a) patent-network construction based on the proposed relationship metrics between different types of patent nodes and (b) patent-class prediction based on patent-network analysis and the modified *kNN* classifier.

Our results show that the proposed patent-network-based method outperformed the content-based, citation-based, and modified metadata-based methods with statistically significant differences. The difference between the patent-network-based method and the metadata-based (text + inventor + IPC) method was not significant. We also combined the patent-network-based method with three conventional classification methods to develop a hybrid-patent-classification approach. The experiment results demonstrated that the hybrid approach yields more accurate class predictions than the patent network-based method. The *t*-test results show that our proposed hybrid approach yields more accurate class predictions other classification methods with statistically significant differences. It enhances the classification performance by using a hybrid of multiple classifiers. In terms of the hybrid effect, the results show that the patent-network-based method is more effective than other methods in enhancing the classification performance.

Acknowledgment

This research was supported in part by National Science Council of the Taiwan Grant NSC 96–2416-H-009–007-MY3.

References

Alani, H., Dasmahapatra, S., O'Hara, K., & Shadbolt, N. (2003). Identifying communities of practice through ontology network analysis. IEEE Intelligent Systems, 18(2), 18–25.

Chen, L., Tokuda, N., & Adachi, H. (2003). A patent document retrieval system addressing both semantic and syntactic properties. In Proceedings of the Association for Computational Linguistics Workshop on Patent Corpus Proceeding (PATENT '03) (pp. 1–6). Morristown, NJ: Association for Computational Linguistics.

DOI: 10.1002/asi

Fall, C.J., Torcsvari, A., Benzineb, K., & Karetka, G. (2003). Automated categorization in the international patent classification. SIGIR Forum, 37(1), 10–25.

Fall, C.J., Torcsvari, A., Benzineb, K., & Karetka, G. (2004). Automated categorization of German-language patent documents. Expert Systems with Applications, 26(2), 269–277.

Guan, J.C., & Gao, X. (2009). Exploring the h-index at patent level. Journal of the American Society for Information Science and Technology, 60(1), 35–40.

He, C., & Loh, H.T. (2008). Grouping of TRIZ Inventive Principles to facilitate automatic patent classification. Expert Systems with Applications, 34(1), 788–795.

He, C., & Loh, H.T. (2010). Pattern-oriented associative rule-based patent classification. Expert Systems with Applications, 37(3), 2395–2404.

Kim, J.H., & Choi, K.S. (2007). Patent document categorization based on semantic structural information. Information Processing & Management, 43(5), 1200–1215.

Kohonen, T., Kaski, S., Lagus, K., Salojavi, J., Honkela, J., & Paatetro, V., et al. (2000). Self organization of a massive document collection. IEEE Transactions on Neural Networks, 11(3), 574–585.

Lai, K.K., & Wu, S.J. (2005). Using the patent co-citation approach to establish a new patent classification system. Information Processing & Management, 41(2), 313–330.

Larkey, L.S. (1999). A patent search and classification system. In Proceedings of the Fourth ACM Conference on Digital Libraries (pp. 179–183). New York: ACM Press.

Li, X., Chen, H.C., Zhang, Z., & Li, J. (2007). Automatic patent classification using citation network information: An experimental study in nanotechnology. In Proceedings of the Seventh ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 419–427). New York: ACM Press.

Loh, H.T., He, C., & Shen, L. (2006). Automatic classification of patent documents for TRIZ users. World Patent Information, 28(1), 6–13.

O'Hara, K., Alani, H., & Shadbolt, N. (2002). Identifying communities of practice: Analysing ontologies as network to support community recognition. In Proceeding of the International Federation for Information Processing Conference, World Computer Congress on Information Systems: The E-Business Challenge (IFIP) (pp. 89–102). Laxenburg, Austria: International Federation for Information Processing.

Richter, G., & MacFarlane, A. (2005). The impact of metadata on the accuracy of automated patent classification. World Patent Information, 27(1), 13–26.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

Su, F.P., Lai, K.K., Sharma, R.R.K., & Kuo, T.H. (2009). Patent priority network: Linking patent portfolio to strategic goals. Journal of the American Society for Information Science and Technology, 60(11), 2353–2361.

Trappey, A.J.C., Hsu, F.C., Trappey, C.V., & Lin, C.-I. (2006). Development of a patent document classification and search platform using a back-propagation network. Expert Systems with Applications, 31(4), 755–765.

van Rijsbergen, C.J. (1979). Information retrieval (2nd ed.). London: Butterworths.

Yang, Y. (1994). Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In W.B. Croft & C.J. van Rijsbergen (Eds.), Proceedings of the 17th annual International Association for Computing Machinery's Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval (pp. 13–22). New York: ACM Press.

Yuan, Y.C., Carboni, I., & Ehrlich, K. (2010). The impact of awareness and accessibility on expertise retrieval: A multilevel network perspective. Journal of the American Society for Information Science and Technology, 61(4), 700–714.

Appendix

This appendix presents the patent-network-analysis algorithm, which is adopted and modified from the ontology-based network-analysis algorithm (Alani et al., 2003; O'Hara, et al., 2002).

```
Initialize the weights of all nodes to 1.
Create a relationship array of the relationships and weights.
Set the query patent document as the active node.
Mark the current node as unlocked and add it to the node array.
Loop to the maximum number of links to traverse the network.
  Search for the current node in node array.
  If found:

    Mark the node as locked.
        Set the node as the active node.
    Find all nodes connected to the current node in the relationship array.
    Loop to number of connected nodes.

      If a node is not in the node array (new node):
        Weight of node = initial weight + current node weight × weight of connecting relation.
        Mark the node as unlocked and add it to the node array.

      If the node is already in the node array:
        Weight of node = node weight + current node weight ×  weight of connecting relation.
    End loop.

  If not found, then exit.
End loop.
Relevance of node = Weight of node raised to the power of 1/n.
(n = the minimum number of the links traversed to reach the node starting from the query node).
```


DOI: 10.1002/asi