PatentLine: Analyzing Technology Evolution on Multi-View Patent Graphs

Longhui Zhang[†] Lei Li[†] Tao Li[†] Qi Zhang[‡]

[†]School of Computing and Information Sciences
Florida International University
11200 S.W. 8th Street
Miami, FL 33199
{lzhan015,lli003,taoli}@cs.fiu.edu

[‡]School of Computer Science
Fudan University
220 Handan Rd, Yangpu
Shanghai, China 200433
qi_zhang@fudan.edu.cn

ABSTRACT

The fast growth of technologies has driven the advancement of our society. It is often necessary to quickly grab the evolution of technologies in order to better understand the technology trend. The availability of huge volumes of granted patent documents provides a reasonable basis for analyzing technology evolution. In this paper, we propose a unified framework, named `PatentLine`, to generate a technology evolution tree for a given topic or a classification code related to granted patents. The framework integrates different types of patent information, including patent content, citations of patents, temporal relations, etc., and provides a concise yet comprehensive evolution summary. The generated summary enables a variety of patent-related analyses such as identifying relevant prior art and detecting technology gap. A case study on a collection of US patents demonstrates the efficacy of our proposed framework.

Categories and Subject Descriptors: H.3.3[Information Storage and Retrieval]: Information Filtering

Keywords: Patent Evolution; Dominating Set; Steiner Tree

1. INTRODUCTION

Technological innovation is becoming one of the important factors that stimulate the development of our society. Granted patents, as the major carrier for technology documentation, have great potential to provide valuable insights of technologies. Analyzing patent documents enables us to effectively understand technological progress, comprehend the evolution of technologies and grab the emergence of new technologies [3]. One representative application of patent analysis involves that enterprises evaluate and understand the prior art or patent evolution of a specific technical field in the development of new products [15].

In this paper, we study the problem of generating patent evolution tree. The evolutionary analytic result is able to facilitate enterprises to understand technological trend, discover invention hot spots and predict research directions. Given a collection of patent documents, a key question is what are the useful resources contained in these patent documents that can be adopted for generating an evolution tree. In the domain of patent analysis, a wide selection of information is available for analysis, including the content of patent documents, the citation relations, and the temporal orders of different patents. Patent documents are often lengthy with rich content. In addition, citation relations are good indicators for patent trend, which explicitly organize patent documents using citation links [7]. Further, temporal information, e.g., the publication date of patents, is another factor that enables the analysis of patent evolution. In our work, we integrate these types of information in providing reasonable patent evolution tree.

In general, changes in patent trends represent the evolution of technological innovation. It is important for enterprises to obtain an overview of patent trends. There have been a number of research publications and applications that delve into the problem of analyzing patent evolution [6, 10, 12]. For example, Shih et al. assume that a patent trend can be represented by the frequent patents in a specific period, and propose to explore patent trend using association rule mining [12]. However in their work, only citation relations of patents are considered; the trend might be disconnected if there are citation gaps between frequent patents.

To address the aforementioned limitation, in our work, we propose a unified framework, named `PatentLine`, to generate a technology evolution tree for a given set of granted patents. The input to our system is a topic or a classification code relevant to a specific technical field. Our system first retrieves all the patent documents related to the topic/code from a patent database. We then construct a multi-view patent graph in which patent content, citation relations and temporal orders are integrated. The system then selects a set of nodes (patents) using an approximation algorithm for the minimum dominating set problem and creates a patentline by virtue of a directed Steiner tree algorithm. Finally, we summarize the content of each patent on the generated tree and present the tree-based summary to the analysts. Figure 1 depicts an overview of the proposed framework.

Our major contributions are two-fold: (1) The proposed framework combines multiple types of information in patent data to improve the understanding of patent trend by providing an integrated summary of patent documents; and (2) We formulate the problem as a graph-based problem customized by various characteristics of the patent domain.

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609518.

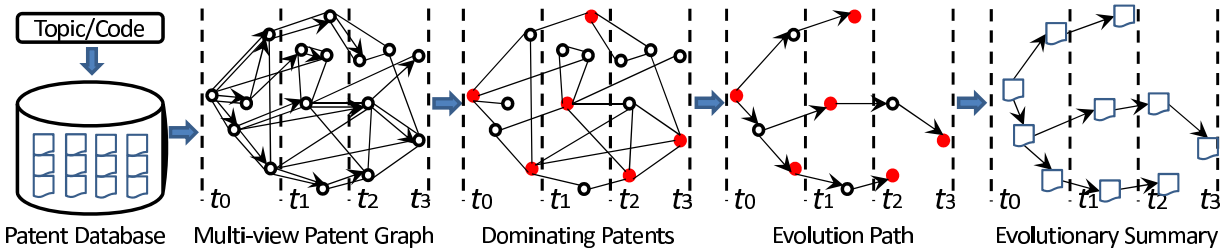

Figure 1: An overview of the framework.

2. ANALYZING PATENT EVOLUTION

The problem of detecting topic evolution has recently attracted increasing interest in the information retrieval community. Most existing approaches focus on identifying evolutionary topics in scientific literatures [1, 2] by making use of vector space model or LDA-alike topic models. Some recent work further tries to analyze the roles of linkage analysis (e.g., the co-authorship [14] or citation analysis [7]) in topic detection and evolution. However, these existing methods cannot be simply applied to our problem setting of generating an evolutionary tree of patents. In addition, the characteristics of patent domain (e.g., lengthy and ambiguous description, full of technical terms) render these methods ineffective in generating patent evolution tree.

Given a topic or a classification code related to a specific technical field, we initially retrieve all available patent documents from a patent database. The problem of generating an evolutionary patentline can be defined as follows: Given a collection of granted patents $D = \{d_1, d_2, \ldots, d_n\}$, generate a patentline represented as a tree $P = \{p_1, p_2, \ldots, p_m\}$ in which each node p_i denotes the summary of patents associated with a timestamp t_i.

Inspired by [13], we first construct a multi-view patent graph using the available information, and then identify dominating/influential patents from the graph, and finally generate summarized patentline based on dominating patents. The procedure is described in Figure 1.

2.1 Constructing Multi-View Patent Graph

As introduced in Section 1, the patent data consists of multiple types of information that shape the relations among patent documents. We use a multi-view graph $\mathbb{G}$ to represent these relations, where $\mathbb{G} = (V, \mathbf{w}_v, E_s, \mathbf{w}_s, E_{ct}, \mathbf{w}_{ct})$.

$\mathbb{G}$ contains a set of nodes/vertices (patent documents) V, where each node $v \in V$ is associated with a cost value w_v and a timestamp t. In our problem setting, the cost w_v is calculated as the inverse of the total number of citations of the corresponding patent document. When selecting dominating nodes, we expect the total cost of selected nodes is minimized.

In addition, the vertices are connected by two types of edges: E_s and E_{ct}. Here E_s contains undirected edges, where each edge connects two patent vertices and the edge weight w_s denotes the content proximity of connected vertices. For patent documents, it is often difficult to calculate the similarity/proximity, as there are a lot of domain-specific and ambiguous terms, and different patents may have their own writing styles. To this end, we extract the most significant section of patents, i.e., `claims`, since this section defines the major invention of patents and often has relatively stable writing structures. We employ "bag-of-words" representation and the cosine measure for proximity computation. Two vertices are linked together if and only if the content proximity is greater than a predefined threshold δ. In our proposed framework, E_s is used for dominating patent selection. Another set of edges, E_{ct}, are directed edges, which are used for evolution tree generation. Each edge in E_{ct} represents either the citation linkage between two vertices, or the temporal order of two vertices. Two vertices form a temporal link if and only if they do not have a citation link and their respective timestamp difference falls into a predefined time range $[\tau_1, \tau_2]$. For simplicity, we assign a unit value 1 to the weight of edges E_{ct}, i.e., $w_{ct} = 1$.

2.2 Identifying Dominating/Influential Patents

To obtain patent evolution tree, we first need to detect the patent documents with representative power, or say, dominating/influential patents. To this end, we define the problem on the undirected part, i.e., $(V, \mathbf{w}_v, E_s, \mathbf{w}_s)$, of the multi-view graph introduced in Section 2.1. Specifically, given the graph $\mathbb{G}$, a *dominating* set of $\mathbb{G}$ is a subset S of vertices with the following property: each vertex $v \in V$ is either in the dominating set S, or is adjacent to some vertices in S. Note that in $\mathbb{G}$, each vertex has a cost indicating the relative influence in terms of citation count, i.e., the larger the cost, the less influential the vertex. The problem of finding a set of influential patent documents can be formulated as the minimum-cost dominating set problem [5].

PROBLEM 1. *Given a graph $\mathbb{G} = (V, \mathbf{w}_v, E_s, \mathbf{w}_s)$ and a budget L, the problem of minimum-cost dominating set (MCDS) is to find a dominating set S, with size L, of vertices in $\mathbb{G}$ whose total vertex cost is the minimum.*

The set cover problem, which is known as an NP-hard problem, can be reduced to the MCDS problem [8]. It has been shown that no algorithm can achieve an approximation factor better than $c \log |V|$ for some $c > 0$. However, we can obtain a greedy approximation for MCDS, as shown in Algorithm 1. Starting from an empty set, if the current subset of vertices is not the dominating set, a new vertex with the minimum averaged cost (with respect to its neighbor size) and not adjacent to any vertex in the current set will be added. In other words, the cost of the new vertex can be evenly shared by its neighbors. Such a greedy algorithm provides a factor of $1 + \log |V|$ approximation of MCDS [11].

Up to this point, we can obtain a set of dominating patents related to the specific technical field, with the limit of a predefined dominator number L.

2.3 Generating Tree-Based PatentLine

The dominating patents obtained from dominating set approximation are capable of representing the rest of patents

Algorithm 1: Approximation of MCDS.

Input: $\mathbb{G} = (V, \mathbf{w}_v, E_s, \mathbf{w}_s)$: undirected patent graph
L: predefined threshold of dominating patents
Output: minimum-cost dominating set S

$S \leftarrow \emptyset$; $T \leftarrow \emptyset$
while $|S| < L$ **do**
  for $v \in V - S$ **do**
    $s(v) = |\{v' | (v', v) \in E_s\} \setminus T|$
  $v^* = \arg\min_v \frac{cost(v)}{s(v)}$
  $S = S \cup \{v^*\}$; $T = T \cup \{v' | (v', v^*) \in E_s\}$
return S

in the graph in terms of content proximity and citation influence. However, there might be some technical gaps among these patents, that is, they may not be well connected. In order to provide a fluent structure of patent documents, e.g., a patentline, we have to find ways to link them together. Also, for presentation purpose, the generated structure of patent documents should be as dense and informative as possible, i.e., to include the minimum number of patents or have the maximum influence over other options.

To tackle this problem, we utilize the directed part, i.e., $(V, \mathbf{w}_v, E_{ct}, \mathbf{w}_{ct})$, of the multi-view graph introduced in Section 2.1. We formulate the problem as the minimum-cost Steiner tree problem. Given a graph $\mathbb{G}$ and a subset of vertices S, a Steiner tree of $\mathbb{G}$ is similar to minimum spanning tree, defined as the subtree of $\mathbb{G}$ that contains S with the minimum total cost. In our problem setting, the cost is defined as the total cost of vertices in the Steiner tree.

PROBLEM 2. *Given a graph $\mathbb{G} = (V, \mathbf{w}_v, E_{ct}, \mathbf{w}_{ct})$, a vertex set $S \subset V$ (terminals) and a vertex $v_0 \in S$ from which every vertex of S is reachable in $\mathbb{G}$, the problem of minimum-cost Steiner tree (MCST) is to find the subtree of $\mathbb{G}$ rooted at v_0 that subsumes S with minimum total vertex cost.*

Algorithm 2: $Steiner_i(\mathbb{G}, S, v_0, k)$

Input: $\mathbb{G} = (V, \mathbf{w}_v, E_{ct}, \mathbf{w}_{ct})$: directed patent graph
S: terminal set
$v_0 \in S$: root of the Steiner tree
k: target size of terminals to be covered
Output: **T**: a Steiner tree rooted at r_0 covering at least k terminals

$T \leftarrow \emptyset$
while $k > 0$ **do**
  $T_{opt} \leftarrow \emptyset$; $cost(T_{opt}) \leftarrow \infty$
  for v, $(v_0, v) \in E_{ct}$, *and* k', $1 \le k' \le k$ **do**
    $T' \leftarrow Steiner_{i-1}(\mathbb{G}, S, v, k') \cup \{(v_0, v)\}$
    if $(cost(T_{opt}) > cost(T'))$ **then**
      $T_{opt} \leftarrow T'$
  $T \leftarrow T \cup T_{opt}$; $k \leftarrow k - |S \cap V(T_{opt})|$;
  $S \leftarrow S \setminus V(T_{opt})$
return T

The problem of MCST, a directed version of the Steiner tree problem, is known as an NP-hard problem [9]. As suggested by [4], a reasonable approximation can be achieved by finding the shortest path from the root to each terminal and then combining the paths, with the approximation ratio of $O(\log^2 k)$, where k is the number of terminals. The approximation algorithm is described in Algorithm 2.

The algorithm employs a recursive way to generate the Steiner tree T. It takes a level parameter $i \ge 1$. When $i = 1$, $Steiner_1$ is simple to describe, i.e., to find the k terminals which are the closest to the root v_0 and connect them to v_0 using shortest paths. As $i > 1$, $Steiner_i$ repeatedly finds a vertex v adjacent to the input root of the i-th function and a number k' such that the cost of the updated tree is the least among all the trees of this form. After obtaining the expected path, we update the corresponding Steiner tree, the target size k and the terminal set S.

The generated Steiner tree of the patent graph gives us an elegant representation of patent evolution, which describes the transitions from the root patent to all the other dominating patents. Once the Steiner tree is generated, we can easily obtain a concise summary for each patent in the tree by applying document summarization techniques.

3. EMPIRICAL EVALUATION

3.1 Patent Data

The patent dataset we have collected includes 2,378 patent documents granted after Jan 1st, 2006 from United States Patent & Trademark Office (USPTO)[1]. The major international classification code of the collected patents is "G06Q 10/00", representing the topic of "data processing systems or processes for administration and management of an organization, enterprise or employees". This code includes 5 subcodes, and their descriptions are shown in Table 1.

Table 1: The description of patent classification.

Code	Description
G06Q 10/02	Reservations, e.g., meetings
G06Q 10/04	Forecasting or optimization
G06Q 10/06	Workflow management
G06Q 10/08	Inventory management
G06Q 10/10	Office automation

3.2 A Case Study

Evaluating technology evolution is a subjective process, as it is difficult to obtain annotated ground truth. Hence, to illustrate the efficacy of our proposed framework, we present a case study on the collected patent data. As an initial step, we extract the title, claims, and citations of patents, and perform natural language processing on claims, including removing stopwords, tokenizing, stemming, etc. We then calculate the content proximity of patents using "bag-of-words" model. To construct the multi-view patent graph, we empirically set the content proximity threshold δ as 0.2, and the time range as 3 months.

We run MCDS (limiting the number of dominators to be 10) and MCST on the generated multi-view patent graph, and the resulted Steiner tree is demonstrated in Figure 2, organized by the temporal order of patents. For representation purpose, we only list the keywords that are contained in the title of patents. The **bold** rectangles denote the dominators identified by MCDS. As observed in Figure 2, "Management" in "G06Q 10/00" starts from manipulating data, as described

[1]http://www.uspto.gov.

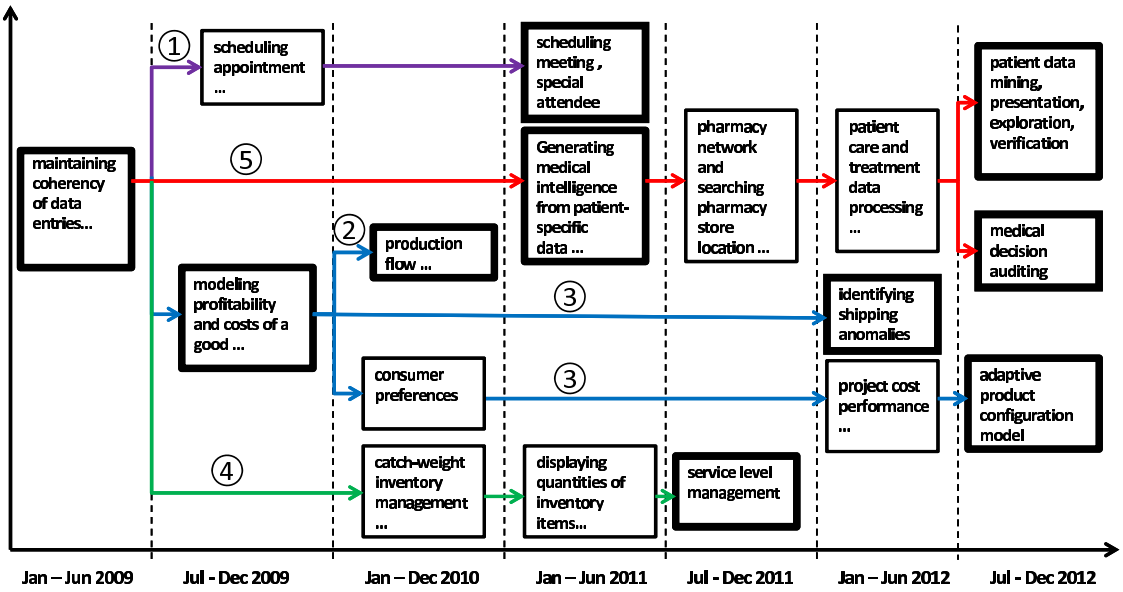

Figure 2: A case study of PatentLine.

in the first dominator, and then can be decomposed into several subtopics. The line labeled as ① mainly describes meeting scheduling, which is related to "G06Q 10/02". The lines of ② and ③ include production workflows and optimizing project, etc., which correspond to "G06Q 10/06" and "G06Q 10/04", respectively. The path labeled as ④ depicts some techniques of inventory and service management, which is relevant to "G06Q 10/08". These three evolution paths give us a general understanding of how technologies evolve with respect to the corresponding categories. These results have been reviewed and assessed by domain experts.

One interesting phenomenon in Figure 2 is the path of ⑤, which describes the technologies of health care management, such as medical intelligence, patient treatment, etc. From Table 1 we cannot find a mapping between this topic and the available codes. We further check the detailed assignments of classification codes to the patents along this line, and find that besides "G06Q 10/00", the patents are all assigned to the code "G06Q 50/00", which includes the classification of health care and patient record management. It somehow indicates that "G06Q 50/00" is more suitable to these patents rather than "G06Q 10/00". The analysts may be able to obtain more insights by using our proposed framework.

4. CONCLUSION

In this paper, we study the problem of exploring technology evolution using granted patent documents. Based on the analysis of domain characteristics of patents, we propose a unified framework, called `PatentLine`, to generate the technology evolution tree in a structural way. We employ graph-based optimization approaches to solve this problem, which is formulated as minimum-cost dominating set and minimum-cost Steiner tree problems. A case study on a patent dataset demonstrates the efficacy of our framework. One interesting extension of our work involves generating a patent evolution path given the earliest and latest patent documents, by which we can have a understanding on how the technologies are evolving from one to another.

ACKNOWLEDGMENT

The work is partially supported by US National Science Foundation under grants DBI-0850203, CCF-0939179, CNS-1126619, and IIS-1213026 and Army Research Office under grant number W911NF-10-1-0366 and W911NF-12-1-0431.

5. REFERENCES

[1] L. Bolelli, Ş. Ertekin, and C. L. Giles. Topic and trend detection in text collections using latent dirichlet allocation. In *Advances in Information Retrieval*. 2009.

[2] L. Bolelli, S. Ertekin, D. Zhou, and C. L. Giles. Finding topic trends in digital libraries. In *Digital libraries*, 2009.

[3] A. F. Breitzman and M. E. Mogee. The many applications of patent analysis. *Information Science*, 2002.

[4] M. Charikar, C. Chekuri, T.-y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed steiner problems. In *SIAM*, 1998.

[5] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 2003.

[6] H. Dou, V. Leveillé, S. D. Manullang, and J. M. Dou Jr. Patent analysis for competitive technical intelligence and innovative thinking. *Data science journal*, 2005.

[7] Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles. Detecting topic evolution in scientific literature: how can citations help? In *CIKM*, 2009.

[8] V. Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, 1992.

[9] R. M. Karp. *Reducibility among combinatorial problems*. 1972.

[10] H. Nanba, T. Kondo, and T. Takezawa. Automatic creation of a technical trend map from research papers and patents. In *Patent information retrieval*. ACM, 2010.

[11] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *ACM Theory of computing*, 1997.

[12] M.-J. Shih, D.-R. Liu, and M.-L. Hsu. Mining changes in patent trends for competitive intelligence. In *Advances in Knowledge Discovery and Data Mining*. 2008.

[13] D. Wang, T. Li, and M. Ogihara. Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In *AAAI*, 2012.

[14] D. Zhou, X. Ji, H. Zha, and C. L. Giles. Topic evolution and social interactions: how authors effect research. In *CIKM*, 2006.

[15] L. Zhang, and T. Li. Data mining applications in Patent Analysis. In *Data Mining Where Theory Meets Practice*. Xiamen University Press, 2013.