PatentMiner: Topic-driven Patent Analysis and Mining

Jie Tang[†], Bo Wang[†], Yang Yang[†], Po Hu[†], Yanting Zhao[†], Xinyu Yan[†], Bo Gao[†],
Minlie Huang[†], Peng Xu[♮], Weichang Li[♮], and Adam K. Usadi[♮]
[†]Department of Computer Science and Technology, Tsinghua University, China
[♮]ExxonMobil Research and Engineering Company, New Jersey, USA
jietang@tsinghua.edu.cn

ABSTRACT

Patenting is one of the most important ways to protect company's core business concepts and proprietary technologies. Analyzing large volume of patent data can uncover the potential competitive or collaborative relations among companies in certain areas, which can provide valuable information to develop strategies for intellectual property (IP), R&D, and marketing. In this paper, we present a novel topic-driven patent analysis and mining system. Instead of merely searching over patent content, we focus on studying the heterogeneous patent network derived from the patent database, which is represented by several types of objects (companies, inventors, and technical content) jointly evolving over time. We design and implement a general topic-driven framework for analyzing and mining the heterogeneous patent network. Specifically, we propose a dynamic probabilistic model to characterize the topical evolution of these objects within the patent network. Based on this modeling framework, we derive several patent analytics tools that can be directly used for IP and R&D strategy planning, including a heterogeneous network co-ranking method, a topic-level competitor evolution analysis algorithm, and a method to summarize the search results. We evaluate the proposed methods on a real-world patent database. The experimental results show that the proposed techniques clearly outperform the corresponding baseline methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining; H.2.8 [**Database Management**]: Database Applications

General Terms

Algorithms, Experimentation

Keywords

Patent analysis, Competitor analysis, Company ranking, Social network

KDD'12, August 12–16, 2012, Beijing, China.

1. INTRODUCTION

Patenting becomes one of the most important ways to protect company's core business concepts and proprietary technologies. Before a company entering the market of a new field, a comprehensive patent landscape overview is always necessary, lest be faced with a flurry of third-party licensing opportunities or even lawsuits. Nowadays, the collection and retrieval of patent publications is a critical component of a company's intellectual property strategy. Within many companies (or organizations), patent analysts have the responsibility to determine how patent information is best made available and how it is best used within their organizations in a strategic manner. From the technical perspective, patent, as one of the few real indicators of future product releases, carries out the technical details of research of different companies long before the product reaches the marketplace. Keeping being aware of novel technology development and competitors' technological advancement also becomes more and more important for a company to make the decision on marketing and R&D strategies.

However, patent analysts in the 21st century now face many challenges. They must search over the huge volume of patents to find relevant patents, to recognize potential competitors (or collaborators), and to identify inventors with significant impact. Despite a great deal of theoretical development in information retrieval and data mining techniques, advanced search tools for patent professionals are still in their infancy. Most existing patent analysis systems such as Google Patent[1], WikiPatent[2], FreePatentsOnline[3] only focus on the search function. A few other systems such as Patents[4], PatentLens[5], and PriorArtSearch[6] provide more advanced analysis and mining capabilities. For example, the Patents system uses iBoogie to cluster the retrieved patents. It also provides a forum for people to discuss different patenting related problems. PatentLens provides a function called Patent Landscapes, which lists a number of "White Papers" discussing technologies relevant to life scientists. However, the Technology Landscapes require extensive searching and analytical work by people skilled in both science and intellectual property, thus infeasible to scale up to various topics.

[1] http://www.google.com/patents
[2] http://www.wikipatents.com/
[3] http://www.freepatentsonline.com/
[4] http://www.patents.com/
[5] http://www.patentlens.net/
[6] http://www.priorartsearch.com/

This paper reports the development of PatentMiner[7], a novel topic-driven patent analysis and mining system. PatentMiner is designed for an in-depth analysis of patent activity at the topic-level. The main unique characteristics of the PatentMiner system that distinguish it from traditional patent search systems are as follows: (1) topic-driven modeling; (2) heterogeneous network co-ranking; (3) intelligent competitive analysis; and (4) patent summarization.

1. Topic-driven modeling. The fundamental problem in most existing systems is that all patents are simply modeled based on keywords. In this work, we present a probabilistic model to simultaneously model the topical aspects of different objects in the heterogeneous patent network.

2. Heterogeneous network co-ranking. When a company plans to enter a new market or brainstorm novel ideas, several typical questions are: what are the most active companies in this area? what are the most relevant patents? and who are the most prolific inventors? We propose a heterogeneous co-ranking algorithm to address these questions.

3. Competitive analysis. It would be very helpful for a company to make right business strategies by identifying who are its competitors and what is the trend of a competitor's technology development. We define four measures to identify competitors, topic-level competitors, and competitors' evolutionary pattern based on the topic modeling results.

4. Patent summarization. It is always expensive to digest the large number of patents returned by a search engine. We present a maximum coverage method to automatically generate a summary for the results of patent search.

We conducted empirical evaluations of the proposed methods. Experimental results show that our methods clearly outperform the baseline methods for addressing the above issues. Our technical contributions in this paper include: (1) a proposal of a probabilistic topic modeling approach, (2) a proposal of a heterogeneous co-ranking method for search over the patent network, (3) a proposal of a topic-level competitive evolution analysis approach, and (4) a proposal of a maximum coverage model for patent summarization.

2. OVERVIEW

Figure 1 shows the architecture of the system. The system consists of five major components:

1. *Patent Network Extraction*: The patent data contains huge amount of information. From the patent data, we derive a heterogeneous information network consisting of different types of objects such as companies, inventors, and patents.

2. *Patent Network Storage*: It provides storage and indexing for the extracted patent networking data. Specifically, for storage it employs MySQL; for indexing, it employs the inverted file indexing method [20].

[7]http://pminer.org

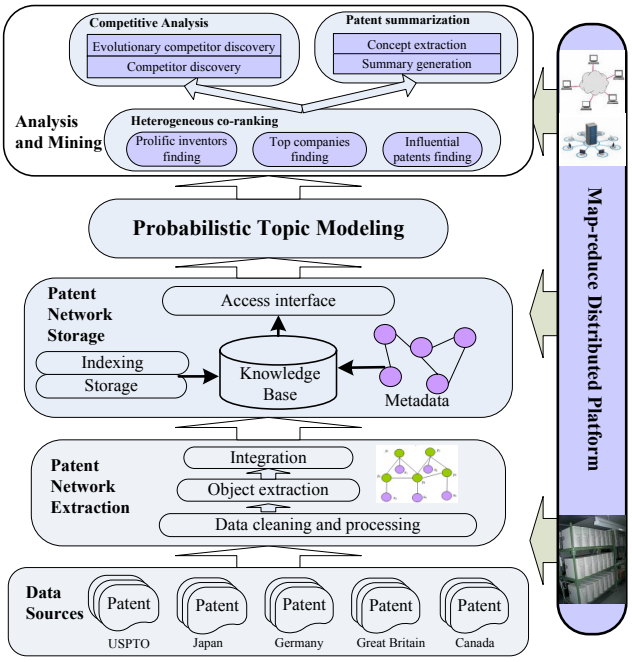

Figure 1: Architecture of PatentMiner.

3. *Probabilistic Topic Modeling*: It utilizes a generative probabilistic model to simultaneously model the different types of objects. After modeling, each object is associated with a topic distribution. The topic modeling is the basis for the analysis and mining component.
4. *Analysis and Mining*: This is the most important component in the PatentMiner system, and our major technical contribution also lies in this component. In summary, it provides three functions: heterogeneous co-ranking, competitive evolution analysis, and patent summarization.
5. *Distributed Platform*: The back-end system is built on the Map-reduce distributed platform [6], a programming platform for distributed processing of large data sets. For some analysis and mining tasks (such as topic model and competitive analysis), we implement the distributed version of the proposed algorithms.

At present, we maintain an English patent database extracted from USPTO.gov, which consists of nearly 4,000,000 patents, 2,000,000 inventors, and 400,000 companies. The system is very flexible and can be easily extended to multiple different sources.

Preliminaries Assume that a patent d contains a vector $\mathbf{w}_d$ of N_d words, in which each word w_{di} is chosen from a vocabulary of size V, and the patent d is developed by a group of inventors $\mathbf{a}_d$ and is owned by company c_d, then a collection of M patents can be represented as $\mathbf{D} = \{(\mathbf{a}_1, c_1, \mathbf{w}_1), \ldots, (\mathbf{a}_M, c_M, \mathbf{w}_M)\}$. Given a collection of patents $\mathbf{D}$, we extract the inventor and company information from each patent, and derive a heterogeneous patent network.

DEFINITION 1. **Heterogenous Patent Network.** *The heterogeneous patent network can be represented as a graph centered by the patent* $G = (V_d \cup V_a \cup V_c, E_{da} \cup E_{dc} \cup E_{dd'} \cup$

Table 1: Notations.

Symbol	Description
M	Number of patents
W	Number of unique words in patents
K	Number of topics
A	Number of inventors
C	Number of companies
N_d	Number of words in patent d
d, c, a	A patent, a company, and an inventor respectively
w_{di}	The i-th word in patent d
x_{di}	The chosen inventor to be responsible for word w_{di}
z_{di}	The topic assigned to the i-th word in patent d
θ_a	Multinomial on topics specific to inventor a
ϕ_z	Multinomial on words specific to topic z
ψ_c	Multinomial on topics specific to company c
α, β, μ	Dirichlet priors to the multinomials θ, ϕ, and ψ

E_{ac}), where V_d includes all patents, V_a includes all inventors, V_c includes all companies, and the edge $(v_d, v_a) \in E_{da}$ (or briefly e_{da}) suggests that there is a relationship between patent v_d and inventor v_a. Similarly we can define the other relationships e_{dc}, $e_{dd'}$, and e_{ac}.

The heterogeneous patent network is comprised of three different types of objects: inventors, companies, and patents. Each object may be associated with different topics. For example, a patent may talk about "web search" and "data mining". The goal of topic modeling over the patent network is to discover the latent topics associated with each object. Each topic z is defined as a mixture of words and their probabilities belonging to the topic, i.e., $\{(w_1, P(w_1|z)), \cdots, (w_{N1}, P(w_{N1}|z))\}$. The definition can be extended to other information sources. For example, we can extend the topic definition by companies, i.e., $\{(c_1, P(c_1|z)), \cdots, (c_{N1}, P(c_{N1}|z))\}$. After topic modeling, each object would be associated with a topic distribution, e.g., an inventor a is associated with $\{P(z|a)\}_z$. By further considering the time information, a patent network can be segmented into a network sequence $GS = (G^1, G^2, \cdots, G^T)$ according to the time-stamp associated with each object. Each sub network G^t in the sequence is comprised of objects in the time window t, e.g., patents published at time t. In this work, the time-stamp is defined as the published year of each patent. Table 1 lists the major notations.

3. MODELING PATENT NETWORK

Several topic models have been proposed and successfully applied to text mining tasks [1, 4, 9, 13, 18]. However, these models can only model patent contents and are not able to incorporate the company and inventor information, which contains valuable information for topic modeling. Moreover, they cannot model the time information. In this section, we will first present an Inventor-Company-Topic model which leverages interdependencies between different objects to learn the topic model, and then describe an extension of the model by combining time information.

3.1 Inventor-Company-Topic (ICT) Model

We present an Inventor-Company-Topic (ICT) model, which incorporates companies, inventors, and patents into a unified probabilistic model. The basic idea is to describe patent writing in a generative process. The generative process can be described as follows: when preparing a patent d, each inventor $x \in \mathbf{a}_d$ would suggest what topics to be included in patent according to his expertise (the associated topic distribution $P(z|\theta_x)$ or θ_{xz}); then the word w_{di} is sampled from a suggested topic z_{di} by the inventor according to $P(w_{di}|z_{di})$ or $\phi_{z_{di}w_{di}}$. In the generative process, all the suggested topics specific to the patent d are relevant to the own company c_d. Aggregating topics of all patents owned by company c_d could constitute a topic distribution of the company $P(z|\psi_c)$ or ψ_{cz}. Formally, we use a uniform distribution to associate each patent with its inventors, use a multinomial distribution (with a prior α) to associate each inventor with the topics, and use a similar multinomial distribution (with a prior β) to associate each topic with words. The company-topic distribution is represented as a mixture of topic distribution extracted from each patent (again can be considered as a multinomial distribution with a different prior μ). Finally, given a collection of patents $\mathbf{D}$, we could write its generative log-likelihood as:

$$\mathcal{L}(\mathbf{D}) = P(\mathbf{x}, \mathbf{z}, \mathbf{w}, \mathbf{c}|\Theta, \Phi, \Psi, \mathbf{a}) = \prod_{d=1}^{M} \prod_{i=1}^{N_d} \frac{1}{A_d} \times \prod_{z=1}^{K} \left(\prod_{x=1}^{A} \theta_{xz}^{m_{xz}} \prod_{j=1}^{W} \phi_{zw_j}^{n_{zw_j}} \prod_{c=1}^{C} \psi_{zc}^{n_{zc}} \right) \quad (1)$$

where m_{xz} is the number of times that topic z was associated with inventor x, n_{zw_j} is the number of times that word w_j is generated by topic z, n_{zc} is the number of times that company c is generated by topic z, N_d is the number of words in patent d, and A_d is the number of inventors for patent d.

Learning the ICT model is to estimate the unknown parameters in the ICT model. There are two sets of unknown parameters: (1) the distribution θ of A inventor-topics, the distribution ϕ of K topic-words, and the distribution ψ of K topic-companies; and (2) the corresponding topic z_{di} and inventor x_{di} for each word w_{di}. It is usually intractable to do exact inference in such a probabilistic model. A variety of algorithms have been proposed to conduct approximate inference, for example variational EM methods [1] and Gibbs sampling [7]. We chose Gibbs sampling for its ease of implementation. Specifically, we calculate the posterior distribution on z and then sample the topic for each word. Based on the sampling results, we could infer the distribution θ, ϕ, and ψ. For the hyperparameters α, β, and μ, for simplicity, we take fixed values (i.e., $\alpha = 50/K$, $\beta = 0.01$, and $\mu = 0.01$.

3.2 Dynamic ICT Model

Though the ICT model incorporates companies, inventors, and patents together, it still cannot capture the temporal information. We therefore propose a dynamic extension of the ICT model. The general idea is that topic distribution of an object (e.g., company) in adjacent time-stamps (e.g., two continuous years) should be similar. At each time-stamp, an ICT model is built, i.e., each object is associated with a topic distribution which will be used as a prior for the object in the next time-stamp. The prior has a smoothing effect to make the discovered topic models between adjacent time-stamps similar to each other. The new model is referred to as Dynamic Inventor-Company-Topic (DICT). To summarize, there are three smoothing requirements for the dynamic modeling:

- Inventor-topic smoothing. The topic distribution of

an inventor should be smooth over time, aka $\Omega_1 = \sum_z(\theta^t_{az} - \theta^{t-1}_{az})^2$ should be small, where θ^t_{az} indicates the probability of topic z given inventor a at time t.

- Company-topic smoothing. The topic distribution of a company should be smooth over time, aka $\Omega_2 = \sum_z(\psi^t_{cz} - \psi^{t-1}_{cz})^2$ should be small.

- Topic smoothing. The topic distribution itself should be smooth over time, aka $\Omega_3 = \sum_z(P(z)^t - P(z)^{t-1})^2$ should be small.

Now, the problem is how to combine the three smoothing hypothesis into the ICT model. One strategy is to use a regularization framework to describe the three smoothing factors as three regularization terms, and plug into the original log-likelihood function, thus we obtain a new objective function:

$$\mathcal{O}(\mathbf{D}) = -\mathcal{L}(\mathbf{D}) + \gamma_1\Omega_1 + \gamma_2\Omega_2 + \gamma_3\Omega_3 \quad (2)$$

where γ_1, γ_2, and γ_3 are three parameters to balance the importances of different smoothing factors.

It is intractable to solve the new objective function. In existing literatures, there are a few attempts to deal with the constrained regularization framework using approximation algorithms such as [24] and [11]. However, these methods cannot guarantee a convergence. In this paper, we consider an alternative method to solve the problem. Instead of plugging the regularization terms into the log-likelihood function, we use the learned topic model of previous time-stamp as the prior for the topic model of the current time-stamp. Specifically, we use the Gaussian distribution as the prior distribution, e.g. $\alpha_t|\alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I)$. In principle, we can also consider other prior distributions, such as Dirichlet distribution or Gamma distribution. We tested a few distributions and found that the Gaussian distribution has the best performance. With a prior distribution, we can incorporate the smoothing effect directly into the probabilistic generative process (as summarized in Algorithm 1).

To learn the Dynamic ICT model, we can still use the Gibbs sampling algorithm for parameter estimation of the DICT model. In an analogous way, we first estimate the posterior probability of sampling the topic z^t_{di} for each word w^t_{di} with time-stamp t, and then use the sampling results to infer θ^t, ϕ^t, and ψ^t. Specifically, with the learned topic model at time t, we can estimate the probability of a topic given an inventor θ^t_{xz}, the probability of a word given a topic ϕ^t_{zv}, and the probability of a company given a topic ψ^t_{zc} respectively by (Derivation is omitted for brevity.):

$$\theta^t_{xz} = \frac{m^t_{xz} + \alpha^t_z + \tau(m^{t-1}_{xz} + \alpha^{t-1}_z)}{\sum_{z'}(m^t_{x,z'} + \alpha^t_{z'}) + \tau\sum_{z'}(m^{t-1}_{xz'} + \alpha^{t-1}_{z'})} \quad (3)$$

$$\phi^t_{zw} = \frac{n^t_{zw} + \beta^t_w + \tau(n^{t-1}_{zw} + \beta^{t-1}_w)}{\sum_{w'}(n^t_{zw'} + \beta^t_{w'}) + \tau\sum_{w'}(n^{t-1}_{zw'} + \beta^{t-1}_{w'})} \quad (4)$$

$$\psi^t_{zc} = \frac{n^t_{zc} + \mu^t_c + \tau(n^{t-1}_{zc} + \mu^{t-1}_c)}{\sum_{c'}(n^t_{zc'} + \mu^t_{c'}) + \tau\sum_{c'}(n^{t-1}_{zc'} + \mu^{t-1}_{c'})} \quad (5)$$

where τ is a parameter to control the influence of the topic model of the previous time on the topic model of to the current time.

Initialize $\alpha^0 = 50/K$, $\beta^0 = 0.01$, and $\mu^0 = 0.01$;
foreach *time-stamp* t **do**
  Draw $\alpha^t|\alpha^{t-1} \sim \mathcal{N}(\alpha^{t-1}, \delta^2 I)$;
  Draw $\beta^t|\beta^{t-1} \sim \mathcal{N}(\beta^{t-1}, \sigma^2 I)$;
  Draw $\mu^t|\mu^{t-1} \sim \mathcal{N}(\mu^{t-1}, \epsilon^2 I)$;
  For each topic z^t, draw ϕ^t_z and ψ^t_z respectively from Dirichlet prior β^t and μ^t;
  foreach *word* w_{di} *in patent* d **do**
    Draw an inventor x_{di} from $\mathbf{a}_d$ uniformly;
    Draw a topic z^t_{di} from a multinomial distribution $\theta^t_{x_{di}}$ specific to inventor x_{di}, where θ^t is generated from the Dirichlet prior α^t;
    Draw a word w^t_{di} from multinomial $\phi^t_{z_{di}}$;
    Draw a company stamp c^t_{di} from multinomial $\psi^t_{z_{di}}$;
  end
end

Algorithm 1: Probabilistic generative process in DICT.

4. HETEROGENEOUS CO-RANKING

The unique requirements of searching over the heterogeneous patent network give rises to several challenging issues and make them different from general search engines. First, the information seeking practice [8] is not only about patents, but also about other information sources, such as companies and inventors. In this spirit, a good patent search engine should provide supports not only for patents, but also for these information sources. Second, search over the patent network typically requires much higher retrieval accuracy. Given a query, such as "data mining", a user does not mean to find patents merely containing these two words. Her/his intention is to find patents on the data mining topic. Finally, these two issues are often intertwined.

Formally, given a heterogeneous patent network $G = (V, E)$, and a query $q = \{w_1, \cdots, w_n\}$, our objective is to leverage the power of both *textual content* (patent content) and the *network information* (relationships between different types of objects) in the patent network to obtain accurate ranking results for companies, inventors, and patents.

To deal with the ranking problem over the patent network, a straightforward method is to first represent each object with a bag of words, and then calculate the relevance score of each object with a given query q by using methods such as language model [22] or vector space model [20]. We use language model as the example to explain how to calculate the relevance score. Language model is one of the state-of-the-art approaches for information retrieval. It interprets the relevance between a document and a query word as a generative probability:

$$P_{LM}(w|d) = \frac{N_d}{N_d + \lambda} \cdot \frac{tf(w,d)}{N_d} + (1 - \frac{N_d}{N_d + \lambda}) \cdot \frac{tf(w,\mathbf{D})}{N_\mathbf{D}} \quad (6)$$

where N_d is the number of words in document (patent) d, $tf(w,d)$ is the word frequency (i.e., occurring number) of word w in d, $N_\mathbf{D}$ is the number of words in the entire collection, and $tf(w,\mathbf{D})$ is the word frequency of word w in the collection $\mathbf{D}$. λ is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [22]. Further, the probability of the document model d generating a query q can be defined as $P(q|d) = \Pi_{w\in q}P(w|d)$. For companies and inventors, we first combine all patents associated with each object and create a virtual document, and then use a similar formula

to calculate the relevance score of company $P(q|c)$ and inventor $P(q|a)$.

However, such a method is only based on keyword matching and cannot leverage the topic modeling information. To take advantage of the topic modeling results, we define another relevance score:

$$P_{ICT}(w|d,\theta,\phi) = \sum_{z=1}^{K}\sum_{x=1}^{A_d} P(w|z)P(z|x)P(x|d) \quad (7)$$

where $P(w|z) = \phi_{zw}$, $P(z|x) = \theta_{xz}$, and $P(x|d) = \frac{1}{A_d}$. By combining the two relevance scores, we have:

$$P(w|d) = P_{LM}(w|d) \times P_{ICT}(w|d) \quad (8)$$

In practice, the learned topics by the topic model is usually general to a given query while the language model is specific to keywords in the query. Combining the two relevance scores achieves a balance between generality and specificity, thus could improve the ranking performance. However, this method still does not consider the network information. We therefore propose a heterogeneous co-ranking method to address this issue.

The basic idea of the heterogenous co-ranking method is to propagate the relevance score between the linked objects in the network. The intuition behind the method is as follows: (a) a patent applied by inventors with higher expertise degrees on a topic (query) is more likely to have a higher quality (or impact); (b) a company who owns many high quality patents on a topic is more likely to be ranked higher; (c) an inventor who applies many patents with high impacts should be ranked higher. Similar strategies have been also considered in [23, 14] for academic search. Based on this intuition, we propose a two-stage method. In the first stage, we use Eq. 8 to calculate the relevance score of each object to the given query q and select the top-ranked objects as candidates. In the second stage, we use the candidates to construct a heterogeneous subgraph and perform a score propagation on the subgraph. Finally, we use the new score to re-rank each type of objects. In the propagation, we calculate the new score by (here we use company as the example):

$$r^k[c] = (1-\xi_1-\xi_2)r^{k-1}[c] + \frac{\xi_1}{|V_a^c|}\sum_{d\in V_a^c} r^{k-1}[a] + \frac{\xi_2}{|V_d^c|}\sum_{d\in V_d^c} r^{k-1}[d]$$

where $r^k[c]$ is the ranking score of company c after the k-step propagation; the score is initialized by Eq. 8; V_a^c denotes a set of inventors related to company c and V_d^c denotes a set of patents owned by company c; ξ_1 and ξ_2 are two parameters to control the propagation. The number of propagation steps reflects how we trust the network information. Setting $k = 0$ indicates that we only use the content information, thus the method degrades to Eq. 8; while setting $k = \infty$ indicates that we only trust the network information, thus the algorithm obtains a result similar to that of PageRank [12] on the heterogenous network.

5. COMPETITIVE ANALYSIS

This component aims to quantitatively characterize the competitive relations between companies. Based on the modeling results of the DICT model, we define four measures to quantify the competitive relations.

Global competitor discovery If two companies compete in their major areas, we call these two companies global competitors of each other. For example, ExxonMobil and Shell Oil are two global competitors: they are two energy companies, competing in "oil exploration", "oil refinery", and "chemical". Given a company c, we define the following measures to rank its global competitors:

- Word-based similarity (WBS). It represents each company by a vector of words, and ranks the competitors based on (Cosine) similarity between company c and each candidate.
- Topic-based divergence (TBD). It represents each company using the topic distribution $\{P(z|c)\}_z$ (or ψ_c), and ranks the competitors by the KL-divergence between company c and each candidate c', i.e.,
$$KL(\psi_c||\psi_{c'}) = \sum_{i=1}^{K} \psi_{cz_i} \log \frac{\psi_{cz_i}}{\psi_{c'z_i}}.$$
- Probability-based correlation (PBC). It defines a correlation score to rank the candidate companies. For a company c and a candidate competitor c', the score is defined as:
$$S(c,c') = \sum_{i=1}^{K} p(z_i|c)p(z_i|c') + \eta(\ln(M_c) - \ln(M_{c'}))^2$$
where M_c is the number of patents by company c; η is balance parameter; $P(z_i|c)$ can be obtained using Bayes rule based on ψ_{z_ic}. The second term is used to avoid noise.

Topic-level competitor discovery Two companies may only compete in one or a few specific areas. For example, Apple and Amazon may not be global competitors, but they compete fiercely on "Tablet PC". Given a company c, we aim to find its competitors on a specific topic z. The simplest way is to utilize the topic distribution associated with each company. For two companies c and c', if $|P(c|z) - P(c'|z)| \leq \tau$, then we say that c and c' are competitors on topic z. The method is referred to as distribution-based competitor finding (DBC). However, this method ignores the correlation between topics. For example, companies who compete on "data mining" may also compete on "web search", as the two topics have a strong correlation with each other. To incorporate the topic correlation, we define a hybrid measure (HBC) as follows:

$$S(c,c',z) = (\psi_{zc} - \psi_{z'c})^2 + \eta \sum_{z'\neq z} \rho_{zz'}(\psi_{z'c} - \psi_{z'c'})^2$$

where $\rho_{zz'}$ is the correlation between topic z and topic z', could be calculated as the negative Kullback-Leibler divergence [22]: $\rho_{zz'} = -KL(\phi_z||\phi'_z)$.

Evolutionary competitor discovery Companies may change their IP and marketing strategies, thus the competitive relations between them would change over time as well. Based on the dynamic ICT model, we define the competitive degree between two companies c and c' at time t as:

$$S(c,c',t) = \sum_{t'=1}^{t} \pi_{tt'} \sum_{i=1}^{K} p(c|z_i,t')p(c'|z_i,t') + \eta(\ln(M_c^t) - \ln(M_{c'}^t))^2$$

where $P(z_i|c)$ can be obtained using Bayes rule based on $\psi^t_{z_i c}$. We use the parameter $\pi_{tt'}$ (defined as $e^{-|t-t'|}$) to model the historic information in the above equation, which means if two companies are competitors in a recent past time, it is also likely that they are competitors in the current time.

We can further extend the evolutionary competitor discovery to the topic level. The basic idea is to replace the inner summation in the above equation with the specific topic.

6. PATENT SUMMARIZATION

When a user performs a search in the patent mining system, a large number of objects (patents, companies, and inventors) may be returned. It is always expensive for the user to digest the large volume information. It is desirable that the system can automatically generate a concise and informative summary for the returned objects, so that the user can quickly grasp a global picture before 'click-and-view' each object. A high-quality summary should satisfy the following requirements: (1) cover the most important information in the returned objects, (2) be relevant to the query as well, and (3) minimize the redundancy in the generated summary.

To solve the patent summarization problem, we propose a maximum coverage method. The idea is to choose a set of representative sentences as the summary from the returned objects for a query. The method consists of three major steps. First, when the user issues a query, it retrieves relevant patents for the query. Second, it extracts concepts from each sentence in the patents. All the extracted concepts form the knowledge space for the query and each sentence is also represented by the extracted concepts. Finally, it employs integer linear programming to find a set of sentences whose concepts maximally cover the knowledge space.

Concept extraction In our maximum coverage method, concept is the basic unit to represent the knowledge. A concept can be a word, a phrase, a named entity (e.g., Person or Location), or even a parsed syntax subtree of a sentence. Each concept has an importance score to the query. The schemes of concept scoring vary from simple term frequency to sophisticated machine learning methods. In our method, candidate concepts are selected using bigrams and are weighted using $TF * IDF$. Specifically, all patents are preprocessed by (1) sentence splitting, (2) part-of-speech tagging, (3) stemming and phrase chunking. Then we collect all extracted noun phrases (obtained by phrase chunking) from the patents and further split them into bigrams as candidate concepts. The next step is to score each candidate concept. In particular, we segment each patent into five fields: Title, Abstract, Claim, Background, and Other. Each field has a weight to reflect its importance (we empirically set the field-weights as 2, 1.5, 1.5, 1.0 and 0.5 respectively). Then, the importance score of each candidate concept is defined as the sum of weighted frequencies, i.e., filed-weight×concept-frequency-in-the-field. Finally, we choose the top 50 candidate concepts with the highest scores to form the knowledge space of the given query.

Summary generation An ideal summary should cover as many important and diverse concepts as possible. The problem can be formulated as a 0-1 knapsack problem. Formally, it aims to maximize the total importance score of concepts that form the knowledge space. In addition, we need to consider several constraints. The first one is the summary length: a user would not want to read a long summary. Thus, we define a maximal number of words in the extracted summary, i.e., $MaxLength$. Another constraint is to maintain the logical consistency between variables. For example, if the summary contains an important concept, then it at least includes one sentence which contains this concept. By combining the objective and the constraints together, we define the following constrained optimization problem:

$$\begin{aligned} & \sum_i IS_i * C_i \\ \text{s.t.} \quad & \sum_j L_j * S_j \leq MaxLength \\ & S_j * Occ_{i,j} \leq C_i \ \forall i,j; \quad \sum_j S_j * Occ_{i,j} \geq C_i \ \forall i \end{aligned} \tag{9}$$

where C_i is an indicator to represent whether the i-th concept is included in the summary ($C_i = 1$) or not ($C_i = 0$), and IS_i is the importance score of the i-th concept. In the first constraint, L_j is the number of words in the j-th sentence, and S_i is an indicator of whether the j-th sentence is included in the summary. In the second and the third constraints, $Occ_{i,j}$ is an indicator of whether the i-th concept occurs in the j-th sentence.

We solve the above optimization problem using integer linear programming (ILP). Specifically, we employ an open source software, LP-Solve (http://lpsolve.sourceforge.net/). The obtained result is an optimal solution that maximizes the objective function. Finally, we construct the summary by selecting sentences with $S_j = 1$.

7. EXPERIMENTAL RESULTS

We evaluated the proposed methods in the context of the PatentMiner system[7], consisting of 3,880,211 patents, 2,134,211 inventors, and 421,032 companies. We conducted three experiments to evaluate the proposed methods: heterogeneous co-ranking, competitor analysis, and patent summarization.

7.1 Results on Heterogeneous Co-Ranking

Data Sets, Evaluation Measures, and Baselines To qualitatively evaluate the proposed methods and compare with existing methods, we collected a list of 50 popular queries (e.g., "data mining", "web search"). For each query, we independently request five annotators (two undergraduates, two PhD students, and one faculty) to provide human judgement on the top (20) returned objects (companies, patents, and inventors) by the system. The judgement is about relevant (Like) or irrelevant (DisLike). If there are more than two annotators saying that an object is irrelevant, we remove the object from the returned list. As it is really difficult to judge the expertise for inventors even for human, we only perform the evaluation for companies and patents. We conducted the evaluation in terms of P@N (Precision for top N results), mean average precision (MAP), and normalized discounted cumulative gain (NDCG) [2, 5].

We used language model (LM) as the baseline method. For language model, we used Eq. 6 to calculate the relevance between a query term and a patent and similar equations for an inventor/company, where an inventor is represented by

Table 2: Average ranking performance for the patents and companies. N@1 and N@5 indicate NDCG for the top 1 or 5 results. HCR-1 indicates our proposed HCR method with one step propagation.

Object	Method	P@1	P@5	MAP	N@1	N@5
Patent	LM	.7001	.6900	.6991	.7021	.6833
	HCR-1	.7592	.7102	.7359	.7592	.7310
	HCR-2	.7598	.7201	.7361	.7600	.7300
	HCR-5	.7600	.7298	.7400	.7678	.7367
Company	LM	.6931	.6790	.6654	.6888	.6532
	HCR-1	.7167	.6833	.7058	.7167	.6934
	HCR-2	.7189	.6900	.7100	.7200	.7000
	HCR-5	.7201	.6999	.7210	.7201	.7031

his/her published patents and a company is represented by its held patents. Our heterogeneous co-ranking method is referred to as HCR. We tried different settings for the propagation. For example, HCR-2 indicates the heterogeneous co-ranking method with 2-step propagation. We preprocessed each patent by (a) removing stopwords and numbers; (b) removing words that appear less than three times in the corpus; and (c) downcasing the obtained words. Moreover, we performed company name disambiguation (e.g., IBM versus IBM Corp.) based on a company name dictionary.

Results and Analysis Table 2 shows the ranking performances for patents and companies. We see that the proposed HCR method (with all settings) clearly outperforms the baseline method using language model. In terms of MAP, the improvement of HCR-5 over language model is 5.3% for patent ranking and 5.01% for company ranking. Our method benefits from the topic modeling results which consider the companies, patents, and inventors in a unified model while the language model can only use the content information. In addition, the propagation process in our HCR methods further leverages the network information.

We studied how the number of propagation step influences the ranking performance. Figure 2 shows the ranking performance in terms of MAP with varied propagation step. Increasing the propagation step from 1 to 5 results in improved performance. This is because that our methods integrated more network information with more propagation steps. However, continuing to increase the propagation step make the network information an dominate factor for the final ranking results, thus hurts the relevance performance.

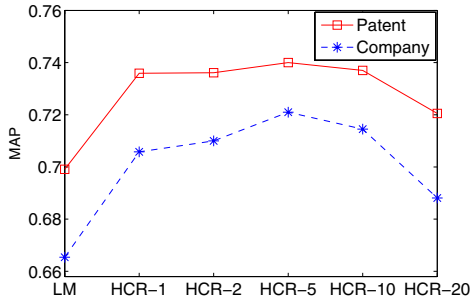

Figure 2: Ranking performance with varied propagation step.

7.2 Results on Competitive Analysis

Data Sets, Evaluation Measures, and Baselines It is difficult to obtain the ground truth for evaluating our method's performance on competitor analysis. For a relatively fair comparison, we have obtained the competitor information from Yahoo! Finance[8], which hosts information of all companies listed on NASDAQ, Dow Jones, S&P, etc. For each company, it gives a list of competitors based on their business performance (such as revenue and employees). It also provides the topic information for the competitors, such as Microsoft and Oracle compete on "Software". In summary, we obtained 543 companies and their global competitor information, and 326 companies and their competitors on 18 different topics.

We used P@N, MAP, and NDCG to evaluate our method (referred to as TopCom), and to compare with two baseline methods. The first one (referred to as WBS) is to represent each company as a bag of words (by the vector space model [20]), and for a given company, its competitors are ranked according to the Cosine similarity of the company with each candidate. This method can identify the global but not topic-level competitors. The second baseline method is for topic-level competitors analysis. We used Latent Dirichlet Allocation (LDA) [1] to generate the topic-word distribution and then combine the language model (LM) for competitor discovery. Specifically, we used all the patent titles to learn the LDA model, and used language model to find the relevant patents for a given query. Finally we obtained the competitive companies by summing all the scores of their corresponding patents. Hereafter we will use "LM+LDA" to denote this method. In addition, we compared our methods with different scoring measures (Cf. Section 5):

- **TopCom+TBD**: It ranks the competitors by KL-divergence of topic distribution between each candidate and the given company.
- **TopCom+PBC**: It ranks the competitors by the probability-based correlation between each candidate and the given company.
- **TopCom+DBC**: It ranks the competitors by the difference of its distribution in a specific topic between each candidate and the given company.
- **TopCom+HBC**: It ranks the competitors by the hybrid-based score between each candidate and the given company.

Results and Analysis Table 3 shows the performance of different methods on global and topic-level competitor analysis. We see that our method (TopCom) outperforms the baseline methods. For the global competitor analysis, our method with the probability-based correlation (PBC) scoring measure achieves the best performance in terms of P@1, P@5, N@1, and N@5. For the topic-level competitor analysis, the best performance is obtained by our method with the hybrid-based (HBC) scoring measure, which indicates that a scoring measure leveraging both topic and patent content information can achieve a better performance than using only one of the information. The advantage of our method lies in that in the proposed DICT model, we simultaneously model the topic distribution of companies, inventors, and patents;

[8] http://finance.yahoo.com/

Table 3: Performance of competitor analysis. N@1 and N@5 indicate NDCG for the top 1 or 5 returned competitors.

	Methods	P@1	P@5	MAP	N@1	N@5
Global	WBS	.2009	.1087	.2904	.2009	.2841
	TopCom+TBD	.1731	.0846	**.3078**	.1731	.2871
	TopCom+PBC	**.2098**	**.1161**	.2920	**.2098**	**.3085**
Topic	LM+LDA	.1536	.1221	.2643	.1536	.2524
	TopCom+DBC	.1369	.1270	.2388	.1469	.2446
	TopCom+HBC	**.1620**	**.1366**	**.2781**	**.1620**	**.2874**

Table 4: Examples for topic-level competitor evolution.

Cisco (Network Device)		**AT&T Corp.** (Communication)		
1996-2000	2006-2010	1996-2000	2001-2005	2006-2010
IBM	3Com	Lucent	Lucent	Lucent
Microsoft	Juniper	IBM	NEC	NEC
Lucent	Broadcom	NEC	Motorola	IBM
AT&T Corp.	Nortel	Verizon	IBM	Bell
Intel	Intel	Microsoft	Broadcom	Fujitsu
Sun	Canon	Samsung	Intel	Samsung
3Com	IBM	Motorola	Microsoft	Motorola
DEC	Fujitsu	Ericsson	Cisco	Verizon
HP	Sony	Alcatel	Samsung	AOL

while the two baseline methods only model the content information (WBS) or only model the topic distribution of patents (LM+LDA).

Case Study Table 4 shows example results of the topic-level competitor evolution analysis by our TopCom+PBC method. From the example, we have observed some interesting patterns. On topic "Network Devices", Cisco's early (1996-2000) competitors include IBM, Microsoft, Lucence, etc., but now it turns out to be 3Com, Juniper, and Broadcom, etc. Juniper seems to be a rising star in the "Network Device" field; while a few other companies (such as AT&T and Lucent) have passed their bloom on "Network Device" since 2001. Instead, AT&T and Lucent have become more focused in the "Communication" field.

7.3 Results on Patent Summarization

Data Sets, Evaluation Metrics, and Baselines The patent summarization method was first tested on benchmark data sets TAC 2008 and 2009 [9] before being applied to patent data. TAC 2008 and 2009 datasets respectively contain 48 and 44 topics (queries). The document collection for each topic is given. The task is, similar to our problem, to generate a 100-word summary from the document collection for each topic (query). Four human-written summaries are used as gold standard for each topic.

We compared our approach with two baseline methods that reported the state-of-the-art performance on this task. The baselines are: Maximal Marginal Relevance (MMR) [3] and Diversity Penalty (DP) [21]. We use ROUGE-1 and ROUGE-2, two commonly used evaluation metrics in document summarization, as the evaluation measures.

Results and Analysis Table 5 lists the results of different summarization methods on the TAC datasets. Our approach clearly outperforms the baseline systems on both datasets. The relative improvements over the baselines are about 6% and 10% in terms of ROUGE-1 and ROUGE-2. We also

[9] http://www.nist.gov/tac/

Table 5: Summarization performance on the TAC datasets. ILP is our approach.

Data	Metrics	Methods			Gold Standard
		DP	MMR	ILP	
TAC2008	ROUGE-1	0.349	0.348	**0.371**	0.414
	ROUGE-2	0.097	0.096	**0.103**	0.116
TAC2009	ROUGE-1	0.334	0.343	**0.372**	0.444
	ROUGE-2	0.091	0.096	**0.105**	0.126

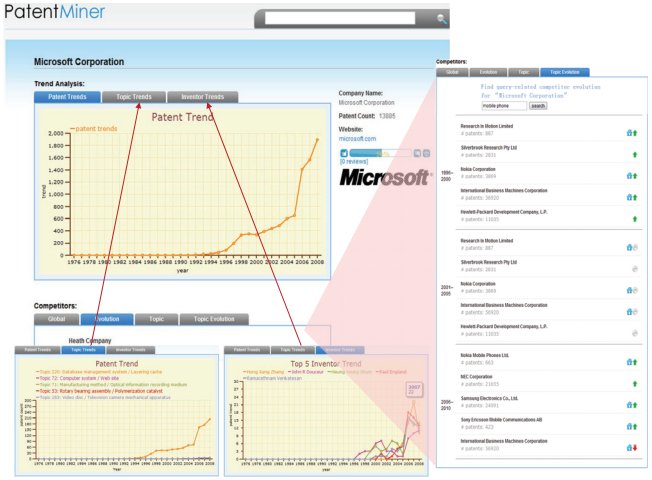

Figure 3: Demonstration system of PatentMiner.

compared our generated results with the human-written results. The last column in Table 5 is the average results of the four human-written summaries for all topics. Therefore, these scores in the last column can be considered as the upper bound of the summarization task. As shown in Table 5, the average score of our approach (0.371/0.372 by ROUGE-1) reaches 86.63% of that of gold standard (0.414/0.444), 86.01% in terms of ROUGE-2, which further confirms the effectiveness of the proposed approach.

7.4 Online System

We have developed a patent analysis and mining system: PatentMiner, and implemented the proposed methods in the system. Figure 3 shows the screenshot of the system. The top-left of is the profile page for "Microsoft Corporation". There are three plots respectively showing the patent application trend, topic trend, and major inventor trend. The right side is the results of competitor evolution analysis on topic "mobile phone".

8. RELATED WORK

There are a few systems for patent search and analysis such as Google Patent, WikiPatent, FreePatentsOnline, Patents, PatentLens, and PriorArtSearch. However, most of these systems focus on search and provide limited macro-level analytic functions. Few systems provide the user with insights into the micro-level analysis of the patent network. No existing system studies the problem of dynamic topic modeling and the topic-level competitor analysis. Tseng et al. [19] introduce a series of text mining techniques for patent analysis, including text segmentation and summary extraction. However, they merely employ existing text mining techniques for patent analysis.

Arnetminer [18] is a "sister" system of PatentMiner. It uses a combination approach to build profiles for academic researchers [16] and provides topic-level expertise search over academic social networks [17]. Compared with these prior works, the PatentMiner system distinguishes itself in the following aspects: dynamic topic-driven modeling, heterogeneous network co-ranking, competitive analysis, and patent summarization, where we propose new approaches to overcome the drawbacks that exist in the traditional methods.

Considerable work has been conducted for extracting topics from text. For example, Hofmann [9] proposes the probabilistic latent semantic indexing (pLSI) and applies it to information retrieval. Blei et al. [1] propose Latent Dirichlet Allocation (LDA) by introducing a conjugate Dirichlet prior for all documents. Several extensions of the LDA model have been proposed, for example, the Author model [10], the Author-Topic model [13], and the Author-Conference-Topic model [18]. The major difference of our ICT and DICT models from existing models is that we simultaneously model dynamic topics of different objects in the patent network.

9. CONCLUSION

We introduce the architecture, algorithms, and main features of the PatentMiner system. We design and implement a general topic-driven framework for analyzing and mining the heterogeneous patent network. Specifically, we first propose a dynamic probabilistic model to model the topical evolution of different objects in the heterogeneous network. We then present a heterogeneous co-ranking approach to rank the multiple objects. We further propose an approach for topic-level competitor analysis. To help users digest the search result, we introduce a method to summarize the patent search results. We evaluate the proposed methods on a real-world patent database and the experimental results validate the effectiveness of our methods.

There are many directions of this work. It would be interesting to further study influence between companies [15]: how a company's technology innovation influences another company's marketing/R&D strategy? Another challenge is how to integrate domain knowledge into the mining process.

Acknolwedgements The work is supported by a research fund from ExxonMobil Corporation. Jie Tang has been supported in part by the Natural Science Foundation of China (No. 61073073, No. 61170061), Chinese National Key Foundation Research (No. 60933013, No. 61035004).

10. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[2] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR'2004*, pages 25–32, 2004.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98*, pages 335–336, 1998.

[4] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML'00*, pages 167–174, 2000.

[5] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec-2005 enterprise track. In *TREC 2005 Conference Notebook*, pages 199–205, 2005.

[6] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI'04*, pages 10–10, 2004.

[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *PNAS'04*, pages 5228–5235, 2004.

[8] M. Hertzum and A. M. Pejtersen. The information-seeking practices of engineers: Searching for documents as well as for people. *Information Processing & Management*, 36(5):761–778, 2000.

[9] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57, 1999.

[10] A. McCallum. Multi-label text classification with a mixture model trained by em. In *Proceedings of AAAI'99 Workshop on Text Learning*, 1999.

[11] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW'07*, pages 171–180, 2007.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford University, 1999.

[13] M. Steyvers, P. Smyth, and T. Griffiths. Probabilistic author-topic models for information discovery. In *KDD'04*, pages 306–315, 2004.

[14] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *ICDM'08*, pages 1055–1060, 2008.

[15] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD'09*, pages 807–816, 2009.

[16] J. Tang, L. Yao, D. Zhang, and J. Zhang. A combination approach to web user profiling. *ACM TKDD*, 5(1):1–44, 2010.

[17] J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su. Topic level expertise search over heterogeneous networks. *Machine Learning Journal*, 82(2):211–237, 2011.

[18] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[19] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin. Text mining techniques for patent analysis. *Inf. Process. Manage.*, 43:1216–1247, September 2007.

[20] C. van Rijsbergen. *Information Retrieval*. But-terworths, London, 1979.

[21] X. Wan, J. Yang, and J. Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *ACL'07*, pages 552–559, 2007.

[22] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR'01*, pages 334–342, 2001.

[23] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *DASFAA'07*, pages 1066–1069, 2007.

[24] X. Zhu and J. Lafferty. Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML'05*, pages 1052–1059, 2005.