Automatic Discovery of Technology Trends from Patent Text

Youngho Kim, Yingshi Tian, Yoonjae Jeong, Ryu Jihee, Sung-Hyon Myaeng
School of Engineering
Information and Communications University
119, Moonji-ro, Yuseong-gu, Daejeon, 305-732, South Korea
Tel. +82-428666210
{yhkim, yingshi424, hybris, zzihee5, myaeng}@icu.ac.kr

ABSTRACT

Patent text is a rich source to discover technological progresses, useful to understand the trend and forecast upcoming advances. For the importance in mind, several researchers have attempted textual-data mining from patent documents. However, previous mining methods are limited in terms of readability, domain-expertise, and adaptability. In this paper, we first formulate the task of technological trend discovery and propose a method for discovering such a trend. We complement a probabilistic approach by adopting linguistic clues and propose an unsupervised procedure to discover technological trends. Based on the experiment, our method is promising not only in its accuracy, 77% in R-precision, but also in its functionality and novelty of discovering meaningful technological trends.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering*

H.4.m [**Information Systems**]: Information Systems Application – *Miscellaneous*

General Terms: Algorithms.

Keywords: Patent, text mining, information extraction, trend.

1. INTRODUCTION

In many application domains, we often encounter a stream of text that meaningfully flows along the time line [2]. In a collection of research papers, for example, we can explore how technological approaches have been changed for a particular topic based on the publication dates of articles (e.g., CiteSeer). Also, as in Topic Detection and Tracking [1], we can monitor the stories for detected events following the time stamp of each event. Thus, based on such text stream, we can discover a meaningful trend in the past by catching the key themes of each domain and predict future trends. Numerous attempts have been made for such discovery of trends from a text stream (i.e., textual-data mining, text-mining) mostly in news articles [1,2].

Similarly, in patent domain, several researchers tried to recognize the progresses of technologies [3,4,9] as patent text is an ample resource to discover technological progresses. They employed a text-mining approach to generate a patent network (i.e., shows how a patent is associated with the other patents) as an analytical tool to recognize emerging technologies [5,6,7,8]. However, those approaches have some drawbacks. First, the results from the previous research are not always helpful for patent readers, depending on how they are organized and presented. A query for a patent retrieval system typically returns too-many related patent documents, which cause information overload. Though Yoon [6] and Kim [7] derived a network-based patent analysis to alleviate this problem, their network is unclear sometimes since the association between key concepts (which they extracted from each patent) is ambiguous and the graph is pell-mell, making it difficult to recognize important concepts. For example, the network in [6] is too complex, not easy to recognize salient concepts. Moreover, in [7], “universal PnP GPS” is connected to “remote control system”, but we cannot see what a specific relation exists between the two concepts. If we can discover the information that “universal PnP GPS” is a solution for implementing “remote control system”, it would be very useful.

Second, the method to develop such a network needs domain-specific knowledge. In [7], for example, forming the network relied on domain expertise. PATExpert [9] also needs expensive ontology for semantic mapping. The technology map of [4] is assisted by patent experts. Thus, the requirements for such knowledge and expertise may hamper wide applications of the proposed methods. At the same time, these knowledge-intensive methods are not so easily generalizable as statistical learning methods. Ahmad[8]’s study is short of a learning algorithm; rather, he proposed a frequency-based analysis of “circuit devices” related documents. Also, in [5] a finite-state machine used to extract key concepts is somewhat rigid, not simple to extend. A more detailed discussion of related work is given in Section 5.

In an attempt to alleviate the limitations mentioned above, we propose a method for (1) semantic key-phrase extraction that plays a key role in discovering technological progresses and (2) technological trend discovery. More specifically, we automatically discover latent technologies from a patent-text stream, and select key technologies for technological trends during a specific time span. We formally define *technology* as a

SAC’09, March 8-12, 2009, Honolulu, Hawaii, U.S.A.
Copyright 2009 ACM 978-1-60558-166-8/09/03…$5.00.

combination of *problem*, such as “recognizing spoken language” and its *solution*, such as “language model” A solution can solve the identified problem in the particular *domain* such as “speech recognition”, which can be given from a user-issued query.

We argue that by extracting a problem and an associated solution in a domain, we can identify a manifestation of technology and develop a Technological Trend Discovery (TTD) system that can help users explore numerous technical documents efficiently. Let’s take patents in the domain of “speech recognition” for example. We can retrieve about 1400 patents from the patent search engine in the USPTO (United States Patent and Trademark Office) web-site. It would be very difficult to go through them one by one to understand what technology has been developed and which technology can be a main trend during a time period.

However, such difficulty can be alleviated if the information in the patent documents is succinctly summarized as in Figure 1. The user can easily recognize the technological progress in “speech recognition” as follows: (1) “dynamic programming” was developed to solve the problem of “speech recognition” in 1980s, and it is changed to “hidden markov model” in 1990s; (2) The problem of “speaker verification” is suggested from 1990s, and “dynamic time warping” was used initially; (3) From 2000s, the “language model” technique has been prevalent for both “speech recognition” and “speaker verification”. Such analysis is derived from the *problem-solution* relation between technological concepts, which facilitates a discovery of technological trends in a domain like “speech recognition”

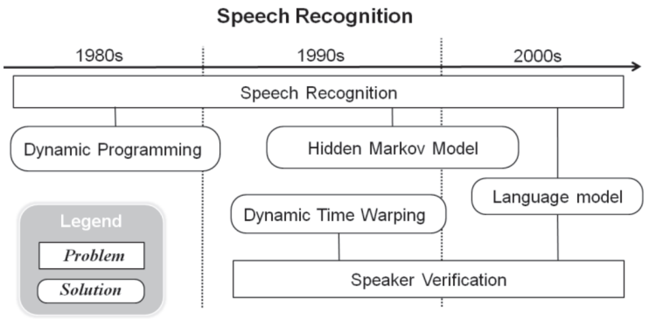

Figure 1. Technological Trend of Speech Recognition

From the previous scenario, we clearly understand the usefulness of our TTD method as an assistant for patent analysis. To implement a TTD system, there should be two key modules corresponding to two tasks: technology identification through semantic key-phrase extraction (Task 1) and technological trend discovery (Task 2). For Task 1, we utilize linguistic clues in combination with a probabilistic approach. The probabilistic method introduced in [2] is useful because it deals with temporally related topics as we do for a time span (e.g., salient technology in 1990s). However, the method is limited to only unigrams whereas our task needs to handle multi-word concepts, i.e., n-grams as can be seen in Fig. 1. In addition, our predefined semantic categories, *problem* and *solution*, are different from the theme defined in [2]. Thus, we extend the probabilistic framework with linguistic clues applicable to our task and increase the adaptability of linguistic clues by adopting a statistical learner assisted by a pattern weighting scheme. For Task 2, we attempt to discover technological trends by selecting important technologies during a time span and linking them according to their semantic relatedness we defined in Section 2. As a result, we can discover technological advances in the past.

We evaluated our system based on USPTO patent data related to the topic of “speech recognition” during 1976~2003. The experimental results show that our method can accurately discover underlying technologies and discover meaningful technological trends in the specified domain of “speech recognition”

This paper is organized as follows. In Section 2, we formulate the task of technological trend discovery. We present our method and system in Section 3. Based on this system, we show the evaluation including the experimental result and related discussions in Section 4. Finally, we present related work and conclusion in Section 5 and 6, respectively.

2. PROBLEM FORMULATION

In this section, we begin with some definitions to formally define the tasks of TTD as follows, which, to our knowledge, have not been described elsewhere.

Definition 1 (Domain): A *domain* is a field of technology given by a user query. A user enters a technological topic as a query if he has information need for an analysis of the related field. Given a domain *D*, we can generate a collection of related documents, i.e., $C_D = \{d_1, d_2, ...d_N\}$ and gather all the key-phrases from every document d_j, i.e., all key-phrases in the collection, $K_D = \{k_1, k_2, ...k_l\}$ where $k_i \in d_j$.

Definition 2 (Problem): A *problem* is a target that a patent or a method attempts to solve and manifested as a key-phrase at a varying level of abstraction in C_D, such as “recognizing signal patterns” Formally, a set of problems is a sub-set of K_D , $P_D = \{p_1, p_2, ..., p_m\} \subset K_D$. A domain sometimes covers a general problem (e.g., “speech recognition”), and therefore a problem can be identical to the given domain.

Definition 3 (Solution): A *solution* is a method, a model or an approach that is associated with a particular problem and manifested as a key-phrase such as “Hidden Markov Model”. We formulate a set of solution phrases as $S_D = \{s_1, s_2, ..., s_n\} \subset K_D$. Generally, the sum of problem and solution key-phrases is bounded by the size of all the key-phrases, $m + n \le l$, since not all key-phrases belong to both categories.

Definition 4 (Technology): A *technology* is defined as a combination of a problem, a solution and the given domain (e.g., “recognizing signal patterns” using “hidden markov model” in “speech recognition”). Basically, the relation between a problem and a solution is inherited if both are extracted from the same document. Since each patent contains its time stamp (i.e., the time when the patent was published) a technology can implicitly inherit the time stamp τ . A formal definition of technology in domain *D* is $t_D =< p, s, \tau >$ where $p \in P_D$, $s \in S_D$

Definition 5 (Time Span): Let the set of all time stamps in the collection *C* be $\mathrm{T}_c = \{\tau_1, \tau_2, ..., \tau_N\}$. A *time span*, *l* is a time period bounded within $\tau_i \le l \le \tau_j$ where $\tau_i, \tau_j \in \mathrm{T}_C$. Determining the time span for a technology is critical, since salient technology is confined within a specific time span. Also, for users it is facile to see a few meaningful technologies; rather than too-many atomic technologies. More details are discussed in Section 3.3.

Definition 6 (Technological Trend): A *technological trend* is a main stream of technologies during a time span *l*. In other words, several salient technologies semantically related to each other can be technological trends consisting of multiple atomic technologies. To recognize such trends, our system should select the most important technologies during *l*. The specific formula to select the representative technologies is introduced in Section 3.3. Among selected technologies, we can make an association between different technologies if they share the same problem (Association 1) or the same solution (Association 2).

By such associations, we can trace the progresses of technologies, considered as TTD in this paper. An example of TTD is shown in Figure 2, where related technologies are associated (e.g., technology 1 and 2 are linked by Association 1); Association 1 and 2 are external linking whereas *Problem* and *Solution* are internal linking by Definition 4. We can analyze the situation as follows; (1) Technology 1 had advanced to Technology 2, since the Solution A had been changed to Solution B in the next time span for the same problem; (2) Technology 4 and Technology 5 can be seen closely related to each other because of the common Solution C; (3) Solution E is a powerful one since it is shared concurrently by Technology 3 and Technology 6.

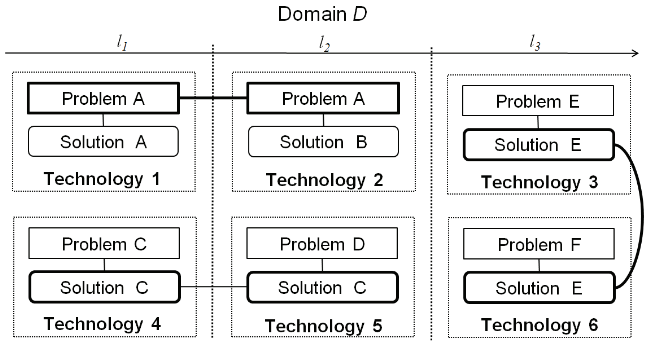

Figure 2. An example of Technological Trend Discovery

3. Technological Trend Discovery System

Our goal in this paper is technological trend discovery. To achieve this goal, we first extract latent technologies as Task 1, and extract salient technologies semantically related each other in Task 2.

3.1 Structure of Patent Documents

From USPTO database[1], we collected US patent data for a given domain. This database is a representative of patent data [8]. As a preliminary work, we analyzed 400 patents related to "speech recognition" by identifying *Problem* and *Solution* in each patent. From this annotation work, the difficulty of acronym (e.g., HMM) handling was identified. To resolve it, we first gathered pairs of an acronym and its full description based on simple parenthetical patterns, e.g., Hidden Markov Model (HMM), from a collection. To collect more of such pairs, a Wikipedia[2] corpus was explored in the same manner. Also, WordNet[3] was used to normalize Noun and Verb; plural and singular for Noun, and conjugations for Verb. A patent document can be divided into two types: structured items (e.g., date, patent class, etc.) and unstructured items (e.g., title, abstract, claim, etc.). Our interest lies in unstructured text, unlike the previous work focusing on structured items [5,7,9], since text streams contain technological trends as discussed in Section 1 and relatively less explored. However, we also utilize the structural information as an aid to text mining.

First, dates give information about when the patent was filed and granted, respectively, and the data in **Filed Date** is considered as a time stamp. The **Reference By** field contains external references to the patents which cite the patent at hand. This bibliographic information is critical for our work, since it is highly probable that patents would cite others if they share the same salient *Problem* or *Solution*. Additional details are discussed in Section 3.2.1. Second, the **Claims** field consisting of a list of claims from the patentee are itemized and initialized by numbers. Many patents contain the basis of the abstract in the first claim. **Claims** and **Description** include a formalized text. For example, the first noun phrase carries a name of a device or a system. Thus as indicated in [9], a simple grammar can be applied for the text-mining purpose, and these linguistic features are also discussed in Section 3.2.2. Finally, the explanation about the others such as assignee, inventors, etc. could be abbreviated, since those do not make much contribution to our task.

Table 1. Structure of US Patent Document

Field	**Value**	**Field**	**Value**
US Patent Number	Number	International Patent Class	Number
Title	Free Text	Reference By	Patent Number
Abstract	Free Text	**Claims**	
Inventors	Proper Noun	**Claim 1**	Formulaic Free Text
Assignee	Proper Noun	**Description**	
Filed Date	Date	**Background**	Formulaic Free Text
Issue Date	Date	**Summary**	Free Text
US Patent Class	Number	**Detailed Description**	Free Text

3.2 Semantic Key-phrase Extraction

The task of semantic key-phrase extraction is to extract *Problem* and *Solution* key-phrases (i.e., Task 1) from a text stream $C_D = \{d_1, d_2, ..., d_N\}$ and consists of the following three steps.

Step-1. All the key-phrases from each document are extracted as candidates for Task 1. By parsing a patent [10], we can recognize a key-phrase as an atomic noun phrase (i.e., the smallest noun phrase, tagged as *NP*), and we can expand the noun phrase to a verb phrase by adding a related verb; dependency between the noun phrase and the verb. Since several problems start with a verb (e.g., recognize signal patterns), the verb expansion is necessary. Thus, we can generate a candidate key-phrase list for each document, since the same phrase can be differently categorized (e.g., "noise reduction" which once was *Problem* can be a *Solution* in another patent, but such ambiguity would not happen in a single patent.)

[1] USPTO search collection http://www.uspto.gov/main/search.html

[2] Free Encyclopedia http://en.wikipedia.org/wiki/Main_Page

[3] Lexical database http://wordnet.princeton.edu/

Step-2. Based on each candidate list, we first identify *Problem* key-phrases by classifying them, since building an effective classifier for *Solution* phrases is more difficult and *Problem* phrases can be used as a lexical indicator for identifying *Solution* key-phrases (discuss in Section 3.2.2). As a result, we have a set of *Problem* key-phrases, $P = \{p_{d_1}, p_{d_2}, ..., p_{d_N}\}$.

Step-3. Among the rest of the candidates, we extract *Solution* key-phrases $S = \{s_{d_1}, s_{d_2}, ..., s_{d_N}\}$, and finally, the set of technologies in the collection is $T = \{t_{d_1}, t_{d_2}, ..., t_{d_N}\}$ where $t_{d_i} = < p_{d_i}, s_{d_i}, \tau_{d_i} >$ (Definition 4). The details for Steps 2&3 are described as follows.

3.2.1 Problem Extraction

Our approach to *Problem* extraction is basically a probabilistic method complemented by linguistic clues useful for identification of problem key-phrases. The probabilistic method adopts the topical language model. That is, we expect that the nature of *Problem* key-phrases is very similar to that of topic keywords (i.e., *Problem* appears frequently in a collection). For example, we expect the frequency of "pattern recognition" would be high in a document and a collection. However, the language model is generally weak where n-gram word distribution (i.e., phrase) is required, due to the data sparseness. Moreover, the smoothing method generally used in Information Retrieval (IR) model is not applicable to our task. Therefore, in a document *d* we estimate the probability of a candidate key-phrase *k* by combining the probability of each component word w_i (i.e., unigram) as follows:

$$p(k|d) = \sum_{i=1}^{n} p(w_i|d) \qquad \text{…equation 1}$$

$$\text{where } k = \{w_1, w_2, ..., w_n\}$$

However, since the occurrence of a word is generally relying on its context (i.e., dependency), we need to consider the dependency between adjacent unigrams in equation 1. Thus, we extend our unigram model to a bigram model:

$$p(k|d) = p(w_1|d) \sum_{i=2}^{n} p(w_i|w_{i-1}, d) \quad \text{…equation 2}$$

However, due to the data sparseness, calculating the dependency over tri-gram is very difficult, and smoothing is not applicable because it is very difficult to find other documents resource which contains similar nature.

The next step is to measure each unigram and bigram probability, and we assume that a word would be derived from the mixture of a language model in the cited documents and a background language model (i.e., a collection). As explained in Section 3.1, each patent document contains its external references, and many *Problem* keywords are shared within cited documents. Such sharing can be investigated, for example, through **Background of Description** field and **Abstract** field in a patent document, since we usually cite the related documents to explain something known (especially, salient *Problem* in our work). Consequently, we assume that problem keywords repeatedly appear within referred documents and frequent in the whole collection. Specifically, let θ_R be the word distribution in references and θ_B be a background language model. We measure the unigram & bigram probability:

$$p(w_i|d) = \lambda p(w_i|\theta_B) + (1-\lambda) p(w_i|\theta_R)$$

$$\begin{aligned} p(w_i|w_{i-1}, d) = {} & \lambda\left\{ p(w_i|w_{i-1}, \theta_B) \cdot p(w_{i-1}|\theta_B) \right\} \\ & + (1-\lambda)\left\{ p(w_i|w_{i-1}, \theta_R) \cdot p(w_{i-1}|\theta_R) \right\} \end{aligned}$$

where λ is a mixing weight, the first is for unigram and the other one is for bigram. Probabilities from θ_B and θ_R are estimated as follows, respectively:

$$p(w|\theta_B) = \frac{\sum_{d_i \in C} cnt(w:d_i)}{\sum_{w' \in V_{Uni}} \sum_{d_i \in C} cnt(w':d_i)}$$

$$p(w|\theta_R) = \frac{\sum_{d_i \in R_d} cnt(w:d_i)}{\sum_{w' \in V_{UniR}} \sum_{d_i \in R_d} cnt(w':d_i)}$$

where *C* is a collection, R_d is references from a target patent *d* (i.e., $w \in d$), V_{Uni} is a unigram vocabulary set from *C* , V_{UniR} is a unigram vocabulary set from R_d, and *cnt*(*w*: d_i) is a word count in a document d_i. Besides, bigram estimation is done as follows:

$$p(w_i|w_{i-1}, \theta_B) = \frac{\sum_{d_i \in C} c(w_{i-1} w_i : d_i)}{\sum_{w'_{i-1} w'_i \in V_{Bi}} \sum_{d_i \in C} c(w'_{i-1} w'_i : d_i)}$$

$$p(w_i|w_{i-1}, \theta_R) = \frac{\sum_{d_i \in R_d} c(w_{i-1} w_i : d_i)}{\sum_{w'_{i-1} w'_i \in V_{BiR}} \sum_{d_i \in R_d} c(w'_{i-1} w'_i : d_i)}$$

where $w_{i-1}w_i$ is a bigram, V_B is a bigram vocabulary set from *C* , V_{BR} is a bigram vocabulary set from R_d, and *cnt*($w_{i-1}w_i$: d_i) is a bigram word count in a document d_i. In addition to this, stop words (e.g., a, the, is) are removed. However, the statistical model alone is not much discriminative at this point. Since θ_B is biased to the topicality, even though θ_R can compensate for it, and *Problem* keywords are not perfectly suitable for the topic keywords, we should seek other discriminative function whose goal is to handle *Problem* keywords only, not topicality.

In the second part, we complement the above probabilistic approach by adopting linguistic clues aiming at only *Problem* keywords. The rationale behind developing clues from the annotated set is as follows. First, most of *Problem* phrases appear in **Title** and **Description** fields. From the annotation, about 57% documents contain *Problem* in the two fields. In addition to this position information, we observed lexical patterns that indicate *Problem*. As noticed in Section 3.1, some parts of a patent are formulaic, and we can observe some patterns to lead *Problem*. For example, "system" and "apparatus" frequently precede *Problem.* We gathered all distinct patterns from the annotation, and generalize them by unification with the common syntactic labels. First, we collect all surface patterns which contains *Problem* phrase. Second, we parse the collected text fragments, and then combine the patterns up to their common syntactic labels (e.g., method/*NN*+in/*PP* and system/*NN*+in/*PP* are unified as (method | system)/*NN*+in/*PP*).

Table 2 shows some samples of generalized patterns. As mentioned in Step-1, the form of the candidate is either of *NP* or *VP*. In the patterns 1 and 2, we can cover "system for verifying speakers" and "device recognizes signals" whereas Pattern 3 is for phrases like "system which removes noises". Patterns 4 and 5 cover “of verifying utterances with” and “for noise reduction using”. Pattern 6 deal with an infinitive like “to summarize speech without decoding”, and Pattern 7 covers “pattern matching system”

Table 2. Samples of *Problem* patterns

No.	Lexico-Syntactic Patterns
1	{method | apparatus | system | device}+{for | of}+ [*Problem* : ***NP*** | (***VBG+NP***)]
2	{method | apparatus | system | device}+[*Problem* : ***VP***]
3	{method | apparatus | system | device}+*WHNP*+ [*Problem* : ***VP***]
4	{to | of}+[*Problem* : ***NP*** | (***VBG+NP***)] + with
5	{for | of}+[*Problem* : ***NP*** | (***VBG+NP***)] + using
6	*TO* + [*Problem* : ***VP***]
7	[*Problem* : ***NP***]+{method | apparatus | system | device}

In order to combine the linguistic clues and the language models, we employ a statistical machine learner to classify the candidates into *Problem* category (i.e., binary classification). Additionally, we develop the bias formula for each pattern, since each generalized pattern would have its confidence and knowing such confidence would improve the classification ability as a positive feature. The feature weight ω is designed as follows:

$$\omega(ptn_j) = \frac{\sum_{\hat{k}\in training \cap problem} \delta(\hat{k}, ptn_j)}{\sum_{k\in training} \delta(k, ptn_j)} \ldots \textit{equation 3}$$

where $\delta(k, ptn_j)$ is 1 when a key-phrase *k* is accepted by pattern ptn_j or otherwise is 0, and the weight indicates how many correct *Problem* phrases (i.e., $\hat{k}$) are discovered by ptn_j in training data.

Overall, the feature space for the machine learner consists of (1) unigram language model probability (*equation 1*), (2) bigram language model probability (*equation 2*), (3) the presence of position information (**bold-faced** fields in Table 1), (4) the presence of 342 generalized patterns (samples in Table 2), and (5) their weights from *equation 3*. Features (1) and (2) are probabilistic features, and others are linguistic features. Also, the feature vector includes the length of each candidate and each candidate’s probability (i.e., summation). We expect a machine learner to optimize the larger feature space, effectively and thus include as many patterns as possible. Since too many patterns may be harmful, however, *equation 3* is used to compensate for it.

3.2.2 Solution Extraction

We now turn to the extraction of *Solution* key-phrases. In this step, we also adopt a statistical machine learner that discovers the solution phrases from the remainder of each candidate list. At this point, probabilistic features (i.e., language model probability) would not be useful as much as in *Problem* extraction because we observed that solution phrases are rarely shared within cited documents. Thus, we mainly utilize lexical clues. From the analysis, we observed that solution phrases frequently come with problem phrases within lexical patterns. Especially in **Title** section, as an example, solution phrases follow problem phrases with “using”, i.e., “speech recognition using language model” From this, we started modeling key features by considering lexico-syntactic patterns with co-occurrence.

Table 3. Samples of *Solution* patterns

No.	Lexico-Syntactic Patterns
1	[*Problem*] + {using | utilizing | employing} + [*Solution* : ***NP*** | (***VBG***+***NP***)]
2	[*Problem*] + {by | with} + [*Solution* : ***NP*** | (***VBG***+***NP***)]
3	[*Solution* : ***NP*** | (***VBG***+***NP***)] + for | in + [*Problem*]
4	[*Solution* : ***NP*** | (***VBG***+***NP***)] **+** *TO* + [*Problem* : *VP*]
5	*TO* + [*Solution* : ***VP***]

For generalization of surface patterns gathered from the annotation, we combine them by parsing and unifying through the shared syntactic labels, as in *Problem* extraction. Table 3 presents samples from the generalization. Pattern 1 can cover “speech recognition using dynamic programming” whereas Pattern 2 is for “speaker verification by dynamic time warp”. Pattern 3 and 4 holds “linear discriminant analysis for speaker verification” and “language model to recognize spoken language”, respectively. Pattern 5 handles an infinitive such as “to assemble two acoustic samples” Moreover we formulate a feature weight for above patterns as follows:

$$\omega(ptn_j) = \frac{\sum_{\hat{k}\in training \cap solution} \delta(\hat{k}, ptn_j)}{\sum_{k\in training} \delta(k, ptn_j)} \ldots \textit{equation 4}$$

where $\delta(k, ptn_j)$ is 1 when a key-phrase *k* is accepted by pattern ptn_j or otherwise is 0, and the weight indicates how many correct *Solution* phrases (i.e., $\hat{k}$) are discovered by ptn_j in training data.

As well as considering lexical patterns, we devised other lexical features. We found that many solutions share the same keyword such as “model” common in “language model” and “hidden markov model”, for example. We define such common keyword as a *head word*. Since head words are duplicated within key phrases, it is difficult to be recognized by lexical patterns; rather we used them as a feature for a statistical learner. Overall 11 head words (i.e., “model, approach, method, methodology, technique, algorithm, analysis, measure, measurement, transform, structure”) were discovered. As a result, the classifier in this step is trained by using (1) 288 generalized *Solution* lexical patterns (samples in Table 3), (2) their weights (*equation 4*), (3) the shortest word distance from a *Problem* (which would cover the co-occurrence), and (4) 11 *head words*.

3.3 Technological Trend Discovery

After semantic extraction, we can discover underlying *technologies* by matching extracted *problems* and *solutions*, as defined in Definition 4. However, numerous distinct technologies would give a cognitive burden to users. Thus, in this Section, we describe how to select several salient technologies and associate semantic relations to them, according to Definition 6.

In Definition 5, we can define a time span from the time stamp information. Our first task is to find an effective time span to discover effective technological trends. We postulate that language models for two time-spans would be different. In other words, if technological advances such as changing of a solution or recognition of a new problem occur, the word distribution after the fact would change. Moreover, we assume that such change would be helpful for TTD. Thus, we utilize KL-divergence to compare two language models from different time spans.

$$D_{KL}(\theta_{l_2} \| \theta_{l_1}) = \sum_{i=1}^{|V_{l_1} \cup V_{l_2}|} p(w_i | \theta_{l_2}) \log \frac{p(w_i | \theta_{l_2})}{p(w_i | \theta_{l_1})}$$

where V_{l_1} is the vocabulary set from all documents within a time span l_1. Since KL-divergence is asymmetric, l_1 should be an earlier time span to observe the difference from l_1 to l_2. After selecting time spans, the rest is to find salient technologies within time spans. To measure the importance of each technology, we count the number of documents that state the given technology. It is assumed that many patents would refer a particular technology if it is important. Such importance is measured as follows:

$$\text{importance}\ (t,l) = \frac{dc(t,l)}{dc(p_t,l) \cdot dc(s_t,l)}$$

where $dc(t,l)$ is a document count for a technology t within a time span l, p_t is a problem and s_t is a solution, which belong to t. We utilize mutual information between a solution and a problem, since a technology consists of two key phrases (i.e., problem and solution). In other words, a *technology* is most salient if its components are salient together. With these measures, we develop the following procedures to discover salient technologies:

Procedure 1. Define an initial time span (e.g., a day, a month, or a year) depending on how dense the collection is.

Procedure 2. Generate all possible combination of time spans, where the combination is valid if two time span is adjacent. Note that time span can be overlapped (e.g., <1998~2000, 1999~2001> from 1998, 1999,2000,2001).

Procedure 3. Calculate KL-divergences of all pairs combined from Procedure 2, and rank them by KL-divergence values.

Procedure 4. Select the most important technology among those in several top-ranked pairs (empirically selected) in Procedure 3.

Defining an initial time span depends on how many documents a collection includes. In Procedures 2 and 3, we exhaustively find meaningful time spans, and from Procedure 4, we can identify most important technologies by calculating the importance of every technology within each given time span. Since the number of meaningful time spans depends on the domain nature (i.e., one domain contains many important technologies whereas another includes a few), picking time spans is sensitive to given domain.

Identifying salient technologies, we now can associate related technologies by Definition 6. Using Association 1 (i.e., shared problems) and Association 2 (i.e., common solutions), we can define a semantic association between two important technologies. Such links can show how technologies are semantically related, i.e., technology A would be advanced to technology B, if they share the same problem and prosper in different ages.

4. EXPERIMENTAL RESULT

In evaluation, we designed a scenario with a user trying to find technological information for "speech recognition". Since a patent search system returns thousands of documents, the user is expected to be effectively assisted by our TTD system.

4.1 Experimental Set-up

Given "speech recognition" as the domain, our collection consists of 1,420 US patent documents from 1976 to 2003. For an evaluation set, we annotated them with *Problem* and *Solution*. We employed three Computer Science graduate students to identify semantic key-phrases in a sample of 400 documents (which are uniformly selected over the span of 30 years, i.e., evenly for each year), and generated the gold-standard with majority votes. For each patent, a problem or solution tag was attached only when two or more annotators agreed. We obtained agreements for 78% of the samples, i.e., about 300 samples. More precisely, 334 problems and 311 solutions were recognized, used as the standard for Task 1. The evaluation of Task 2 was not as rigorous as Task 1, however, since Task 2 includes finding salient technologies. There are too many time spans and documents belonging to them (from 1976 to 2003), and maintaining the same level of rigorousness in establishing a standard is an insurmountable task for the scale of the current project. This is because the number of time spans is enormous even though we sampled only 400 documents.

4.2 Evaluation

This section describes our evaluation results for the two tasks. For Task 1, we use precision and recall to measure how effectively *Problem* and *Solution* key-phrases are extracted (forming *Technology* from this is trivial). For Task 2, we qualitatively analyzed the discovered technological trends.

For the mixture language model as discussed in Section 3.2, we set the bias for background language model $\lambda = 0.28$ empirically and used LIBSVM[4] as a machine learner (since Support Vector Machine is believed to be most powerful). We used 5-fold cross validation, i.e., SVM was trained by five different 240-document sets and tested on five different 60-document sets. In learning, using 240 documents is sufficient, since our task at this point is finding *Problem* or *Solution* phrases from candidates, i.e., the training set contains 6,580 candidates on average for each training. The measurement involves precision and recall as used in IR tasks, but the recall is slightly different since each patent contains generally a single problem and solution. Thus, we measure *R-precision* of each extraction, i.e., we rank our candidates and average precisions at every recall point from each patent.

Table 4. *Problem* Extraction Result

Feature		R-precision
Language Model	Bigram	0.38 (-34%)
	Unigram	0.58 (0%)
LM + Linguistic	not using *equation 3*	0.71 (+22%)
	using *equation 3* (pattern weight)	**0.76 (+31%)**

We ran the experiment on *Problem* extraction to test two hypotheses; (1) Linguistic clues discussed in Section 3.2.1 would be useful; (2) The pattern weight (i.e., *equation 2*) would help SVM. In Table 4, the language model is tested in two ways: unigram and bigram. As discussed in Section 3.2.1, sparseness is

[4] Library for SVM http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 7. Samples results of Technological Trend Discovery

Time Span	1980-1983	1980-1982	1986-1987	1988-1989	1992-1996
Technology	**speech recognition**	**speech recognition**	**speaker verification**	**pattern matching**	**speech recognition**
	dynamic programming	*dynamic time warping*	*dynamic time warping*	*dynamic programming*	*dynamic time warping*
Time Span	1994-1996	1997-1998	1998-2001	1999-2000	2000
Technology	**speech recognition**	**speech recognition**	**speech recognition**	**speaker verification**	**language recognition**
	hidden markov model	*language model*	*hidden markov model*	*language model*	*language model*

harmful to the bigram result. However, linguistic clues are very effective. Since text in patents is somewhat formalized as discussed in Section 3.1, lexico-syntactic patterns are useful. Also, position information (Table 1) included in linguistic clues contributes on the enhancement, since many problems appear in **Title** and **Description**. Moreover, the weight for each pattern is useful, which slightly improves SVM.

Table 5. *Solution* Extraction Result

Feature Space	**R-precision**
Linguistic Patterns	0.62 (0%)
+ *equation 4* (pattern weight)	0.66 (7%)
+ 11 Head Words	0.74 (19%)
+ Word Distances from *Problems*	**0.75 (21%)**

The experiment on *Solution* extraction validates the effectiveness of each feature in Section 3.2.2. Using linguistic patterns alone was tested first, and then we overlapped other features. As Table 5 shows, *head words* turned out to be most helpful as a single source, since most *Solutions* can share such head words. Also, the bias for each pattern, identically used in *Problem* extraction is slightly effective. While co-occurrence information with problem key-phrases is useful, the amount of improvement is minuscule.

From the result of Task 1, we were able to discover many meaningful problems and solutions as in Table 6. Our *Problem* extraction model can extract not only a broader problem such as "voice recognition", but also a specific problem such as "reduce the storage space for the speech recognition dictionary" Such long phrase is not much meaningful in the view of TTD, since it usually contains too-specific problem which cannot express technology trend effectively. Also, the result is mostly biased to "speech recognition", since it is most frequent in the collection.

From Table 6, we can identify a synonymy issue. Although two phrases contain different words at the surface level, they may be semantically identical as in "speaker recognition" and "voice identification". This problem is difficult to handle even if we utilize synonyms from WordNet. In the result of *Solution* extraction, on the other hand, extracted solutions are quite diverse although some salient phrases such as "dynamic programming", "hidden markov model", and "language model" were obtained.

In failure analysis, we recognized errors caused by a narrative type (e.g., "transform the consistent message into electrical signal representation and generate a likelihood score of recognition"), which was not recognized as a solution (in extracting problems, such long phrases rarely occurred). However, such a long phrase with overly specific words would not be critical in our task of technological trend analysis. The current result of semantic extraction contains relatively short phrases that are most useful in expressing technological trends effectively.

For Task 2, our system can discover technological trends by the four step procedure in Section 3.3. We set the initial time span as a year and ran our system using the original 1,420 patents, since the procedure is fully unsupervised. As a result, we present some fragments of a TTD result in Table 7.

Overall, we can identify that salient technologies are associated with the problem of "speech recognition", and from 1999 new problems such as "speaker verification" began to prosper. We can trace that the solution for "speech recognition" has changed from "dynamic programming" in early 1980 to "hidden markov model" and "language model" in 1990s. Also, in 1990s, "dynamic time warping" was highlighted again from 1980s (we can guess that patents in 1992 might develop a new feature or some breakthrough for "dynamic time warping", but unfortunately such analysis cannot be recognized from our system). Based on this analysis, the user can easily recognize the technological mainstream in "speech recognition"

Table 6. Sample Results of Semantic Extraction

Problem
speech recognition, pattern recognition, noise reduction, reduce storage space for speech recognition, , voice identification, speaker recognition, recognition error reduction
Solution
dynamic programming, vector quantization method, shared speech model, lattice-ladder filters, user-cued speech recognition, acoustic model, neural network, dynamic programming algorithm

5. RELATED WORK

Previously, researches on patents can be classified into two categories: patent search & classification and patent analysis.

Studies for patent search purpose an effective navigator to find desired patent. Takaki [12] analyzed claim structures to improve the effectiveness of the search task. Itoh [13] improved the effectiveness of the technology survey task by using the different term distributions. Koster [14] investigated the effectiveness of the bag-of-words approach in classifying patents. Lai [15] used citation-based analysis to perform a patent classification.

In patent analysis, studies derive new meaningful information for effective analysis and provide an intelligent application for readability. Lent [3] attempted to visualize trends, where a trend is a specific subsequence of the history of a phrase using Shape

Query Language. Pottenger [5] captured emerging concepts recognized by hierarchical clustering system. Yoon [6] generated a patent network as an analytical tool for revealing technical progresses. Kim [7] discovered emerging technologies from drawing a patent map. Ahmad [8]'s study tracked the evolution of technology. Wanner [9] proposed a patent processing system which facilitates user access by showing a technical development in semantic representation. Shinmori [11] aimed to improve the readability of patent claims, and proposed a method for analyzing the rhetorical structure. Chakrabarti [16] analyzed the diffusion of technical information in different organizations.

6. CONCLUSION

Text-mining from patent document is highly desirable. Since text streams from patents bear technological progresses, discovering such trends can not only reveal latent technologies, but also assist an exploration by alleviating information overload caused by patent search results. In this paper, we formally defined the basic tasks of extracting problem and solution key-phrases constituting a technology and discovering technological trends and proposed a TTD system that can automatically capture technological mainstream from thousands of related documents. Also, semantic relations between technologies are very useful for a detailed analysis about technological topics. To the best of our knowledge, there is no published study to identify such semantic relations in patent analysis. We rigorously evaluate our system in the task of semantic extraction and present meaningful technological trends in the domain of "speech recognition" Our system accurately extract technologies and can enhance readability. Discriminative features we proposed are generally applicable to other technological domains, and the method to discover technological trends (i.e., task 2) is fully unsupervised.

As a future study, we intend to manage the synonymy issue as in Section 4.2. Also, a standardized evaluation for Task 2 will be investigated, though establishing such standard is now unfeasible.

7. ACKNOWLEDGMENTS

This work was supported by 3rd Phase of Brain Korea 21 project sponsored by Ministry of Education and Human Resources Development, South Korea.

8. REFERENCES

[1] R. Nallpati. Semantic language models for topic detection and tracking. In *Proceedings of the conference of the North American chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL'03)*, 2003, pages 1-6.

[2] Q. Mei and C. Zhai. A mixture model for contextual text mining. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge Discovery and Data mining (KDD'06)*, 2006, pages 649-655.

[3] B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of the 3rd international conference on Knowledge Discovery and Data mining (KDD'97)*, 1997, pages 227–230.

[4] A. Porter and D. Jhu. Technological mapping for management of technology. In *Proceedings of International Symposium on Technology*, 2001.

[5] W. Pottenger and T. Yang. Detecting emerging concepts in textual data mining. *Computational Information Retrieval*, 2001, pages 1-17.

[6] B. Yoon, and Y. Park. A text mining-based patent network: analytical tool for high-technology trend. *Journal of High Technology Management Research,* Vol. 15 (1), 2004, pages 37–50.

[7] Y. Kim, J. Suh, and S. Park. Visualization of patent analysis for emerging technology. *Expert Systems with Applications*, Vol. 34 (3), 2007, pages 1804-1812.

[8] K. Ahmad and A. Al-Thubaity. Can text analysis tell us something about technology progress? In *Proceedings of the ACL-03 workshop on patent corpus processing*, 2003, pages 41-45.

[9] L. Wanner, et al. Towards content-oriented patent document processing. *World Patent Information*, Vol. 30 (1), 2007, pages 21-33.

[10] D. Klein and C. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, 2003, pages 423-430.

[11] A. Shinmori, M. Okumura, Y. Marukawa, and M. Iwayama. Patent claim processing for readability: structure analysis and term explanation. In *Proceedings of the ACL-03 workshop on patent corpus processing*, 2003, pages 56–65.

[12] T. Takaki, A. Fujii, and T. Ishikawa. Associative document retrieval by query subtopic analysis and its application to invalidity patent search. In *Proceedings of the 13th ACM International conference on Information and Knowledge Management (CIKM '04)*, 2004, pages 399-406.

[13] H. Itoh, H. Mano, and Y. Ogawa, Term distillation in patent retrieval. In *Proceedings of the ACL-03 workshop on patent corpus processing*, 2003, pages 41-45.

[14] C. Koster, M. Seutter and J. Beney. Multi-Classification of Patent Applications with winnow. In *Proceedings PSI 2003*, 2003, pages 545–554.

[15] K. Lai, and S. Wu. Using the patent co-citation approach to establish a new patent classification system. *Information Processing and Management*, Vol. 41, 2005, pages 313–330.

[16] A. Chakrabarti, I. Dror, and N. Eakabuse. Interorganizational transfer of knowledge: An analysis of patent citations of a defense firm. *IEEE Transactions on Engineering Management,* Vol. 40 (1), 1993, pages 91–94.