A Patent Document Retrieval System Addressing Both Semantic and Syntactic Properties

Liang Chen
Computer Science Department
University of Northern British Columbia
Prince George, BC, Canada V2N 4Z9
chenl@unbc.ca

Naoyuki Tokuda **Hisahiro Adachi**
R & D Center, Sunflare Company
Shinjuku-Hirose Bldg., 4-7 Yotsuya
Sinjuku-ku, Tokyo, Japan 160-0004
{tokuda_n,adachi_h}@sunflare.co.jp

Abstract

Combining the principle of Differential Latent Semantic Index (DLSI) (Chen et al., 2001) and the Template Matching Technique (Tokuda and Chen, 2001), we propose a new user queries-based patent document retrieval system by NLP technology. The DLSI method first narrows down the search space of a sought-after patent document by content search and the template matching technique then pins down the documents by exploiting the words-based template matching scheme by syntactic search. Compared with the synonymous search scheme by thesaurus dictionaries, the new method results in an improved overall retrieval efficiency of patent documents.

1 Introduction

Information (document) retrieval systems resort to two classes of approaches; the first makes use of the form-based or words-based approach addressing the exact syntactic properties of documents, while the second makes use of the content-based approach which exploits the semantic connection between documents and queries. While most of commercial systems adopt the form-based approach exploiting the simple string matching algorithm or the weighted matching algorithm, the approach needs a thesaurus dictionary to resolve the synonym-related problem. Some research works have now been underway from the content-based approach focusing the dimension reduction scheme.

The content-based approach is motivated by semantics-based search schemes. Assuming that the content of a document is closely related to the *tf-idf* of the words used (Zobel and Moffat, 1998), we first represent documents as term vectors. One of the immediate difficulties we encounter in dealing with document vector spaces lies in its too high a dimensionality of the vector spaces which is particularly true in document analysis largely due to a large variety of synonyms and polysemic words used in natural language. In image recognition field (Turk and Pentland, 1991; Chen and Tokuda, 2003b), a so-called PCA (principal component analysis) principle has been used successfully in facial recognition problems as a most effective scheme of dimension reduction. The LSI (latent semantic indexing) technique (Berry et al., 1999; Littman et al., 1998) is a counterpart of the PCA in text document processing.

We have recently extended the LSI to a DLSI (differential latent semantic indexing) method (Chen et al., 2001), where in the DLSI scheme, we improve the robustness of the LSI scheme by introducing and making use of projections of, interior as well as exterior differential document vectors (see Section 2 for detailed discussions). Our present study shows how we can make use of the characteristics in improving the IR performance in patent document search. In patent retrieval application, we are fortunate because all the patent documents are well structured with very precise, human generated abstracts attached so that two interior and exterior documents are automatically provided, facilitating the application of the DLSI method in developing a patent document retrieval system.

Despite the improved superiority of the DLSI technique over the LSI technique (see Section 2 for detailed discussions), the system still has a problem of instability when used as an NLP-oriented query-based commercial product due to content search's inherent poor precision and recall rate. A content-based information retrieval system is still far beyond our research ability to be implemented into a coding system. Some syntactic properties seeking the "form" or "word" similarity must be introduced if the LSI/DLSI based system can be used with robustness. This is so because we have to resolve some conflicting factors here. The content based IR system tries to search the document in accordance with the similarity of "meaning" of a query, which captures the abstraction of the exact words used. For example, we believe that the LSI/DLSI based system should be able to retrieve a similar set of documents to a query "Information Processing Devices" and "Computing Machinery", where probably some of documents obtained might not contain even the phrases "Information Processing Devices" or "Computing Machinery", or even neither of these words at all. Form based systems, on the other hand, have to depend on the exact words used; in other words, unless a "perfect" thesaurus dictionary is used, we may not capture the correct documents. Unfortunately we know of no such complete thesaurus dictionary, and even if there is such a dictionary, the matching or collating method will be still too complex with respect to computing resources.

To solve "form" similarity problems encountered in a DLSI/LSI approach, we introduce the template-automaton method which has been originally developed for the language tutoring system (Tokuda and Chen, 2001). The template method sets up a variety of expected patterns of patent document abstracts whereby we want to match a query against a multitude of template paths by pinning down a path having the highest similarity measure to the query from among the documents preselected by the DLSI method. All we have to do here is to maintain the template structure containing the possible candidates of the abstracts of patent documents in natural language, and maintain the template structures in the database. A DP(dynamic programming) based-template matching method is very efficient in finding a best matched path to a query facilitating the final location of the patent document.

The rest of the paper is organized as follows. The scheme of the DLSI method is introduced in Section 2 while the template structure will be explained in Section 3. The Flow of the entire search process and concluding remarks will be given in Sections 4 &5.

2 Differential Latent Semantic Indexing Method

A term is defined as a word or a phrase that appears at least in two documents. We exclude the so-called stop words such as "a", "the" in English which are used most frequently in any topics, but remain irrelevant to our purpose of document search.

Suppose we select and list the terms that appear in the documents as $t_1, t_2, ..., t_m$. For each patent document in collection, we preprocess it and assign it with a document vector as $(a_1, a_2, ..., a_m)$, where $a_i = f_i \cdot g_i$; here f_i denotes the number of times the term t_i appears in an expression of the document, and g_i denotes the global weight over all the documents; the weight denotes a parameter indicating the relative importance of the term in representing the document abstracts. Local weights could be either raw occurrence counts, boolean, or logarithms of occurrence count. Global weights could be no weighting (uniform), domain specific, or entropy weighting. The document vector is normalized as $(b_1, b_2, ..., b_m)$. Since all the patent documents are provided with a formal abstract, we suppose the abstracts be equivalent to their documents in content so that the abstract and the document should both be retrieved as part of the similar documents to the query supplied. We will show below how we can set up the DLSI technique leading to an improved robust scheme below. We have shown how the shortcoming of a global projection-based LSI scheme can be improved by making a best use of differences of two vectors in adapting to the unique characteristics of each document (Chen et al., 2001).

A Differential Document Vector is defined as $I_1 - I_2$ where I_1 and I_2 are normalized document vectors satisfying particular types of documents. An Exterior Differential Document Vector in particular is defined as the Differential Document Vector $I = I_1 - I_2$, if I_1 and I_2 constitute two normalized document vectors of any two different documents. An Interior Differential Document Vector is defined by the Differential Document Vector $I = I_1 - I_2$, where I_1 and I_2 constitute two different normalized document vectors of the same document. The different document vectors of the same documents may be taken from parts of documents including abstracts, or may be produced by different schemes of summaries, or from the querries. The Exterior Differential Term-Document Matrix is defined as a matrix, each column of which is set to an Exterior Differential Document Vector. The Interior Differential Term-Document Matrix is defined as a matrix, each column of which comprises an interior Differential Document Vector.

2.1 Details of a DLSI Model

Any differential term-document matrix, say, of m-by-n matrix D of rank $r \leq q = \min(m, n)$, can be decomposed into a product of three matrices, namely $D = USV^T$, such that U and V are an m-by-q and q-by-n unitary matrices respectively, where the first r columns of U and V are the eigenvectors of $DD^T$and D^TD respectively. $S = \mathrm{diag}(\delta_1, \delta_2, \cdots, \delta_q)$, where δ_i are nonnegative square roots of eigen values of DD^T, $\delta_I > 0$ for $i \leq r$ and $\delta_i = 0$ for $i > r$. By convention, the diagonal elements of S are sorted in decreasing order of magnitude. To obtain a new reduced matrix S_k, we simply keep the k-by-k leftmost-upper corner matrix $(k < r)$ of S , other terms being deleted; we similarly obtain the two new matrices U_k and V_k by keeping the leftmost k columns of U and V respectively. The product of U_k, S_k and V_k^T provides a matrix D_k which is approximately equal to D. Each of differential document vector q could find a projection on the k dimensional differential latent semantic fact space spanned by the k columns of U_k. The projection can easily be obtained by $U_k^T q$. Note that, the mean $\bar{x}$ of the exterior-(interior-)differential document vectors are approximately 0. Thus, $\sum = \frac{1}{n}DD^T$, where $\sum$ is the covariance of the distribution computed from the training set. Assuming that the differential document vectors formed follow a high-dimensional Gaussian distribution, the likelihood of any differential document vector x will be given by

$$P(x|D) = \frac{\exp\left[-\frac{1}{2}d(x)\right]}{(2\pi)^{n/2}|\Sigma|^{1/2}},$$

where $d(x) = x^T\Sigma^{-1}x$. Since δ_i^2 are eigenvalues of DD^T, we have $S^2 = U^TDD^TU$, and thus

$$d(x) = nx^T(DD^T)^{-1}x = ny^TS^{-2}y,$$

where $y = U^Tx = (y_1, y_2, \cdots, y_n)^T$.

Because S is a diagonal matrix, $d(x) = n\sum_{i=1}^r y_i^2/\delta_i^2$.

It is convenient to estimate the quantity by

$$\hat{d}(x) = n(\sum_{i=1}^k y_i^2/{\delta_i}^2 + \frac{1}{\rho}\sum_{i=k+1}^r y_i^2).$$

where $\rho = \frac{1}{r-k}\sum_{i=k+1}^r \delta_i^2$.

Because the columns of U are orthonormal vectors, $\sum_{i=k+1}^r y_i^2$ could be estimated by $||x||^2 - \sum_{i=1}^k {y_i}^2$. Thus, the likelihood function $P(x|D)$ could be estimated by
$\hat{P}(x|D) =$

$$\frac{n^{1/2}\exp\left(-\frac{n}{2}\sum_{i=1}^k \frac{y_i^2}{\delta_i^2}\right)\cdot\exp\left(-\frac{n\epsilon^2(x)}{2\rho}\right)}{(2\pi)^{n/2}\prod_{i=1}^k \delta_i \cdot \rho^{(r-k)/2}}, \quad (1)$$

where $y = U_k^Tx$, $\epsilon^2(x) = ||x||^2 - \sum_{i=1}^k y_i^2$, $\rho = \frac{1}{r-k}\sum_{i=k+1}^r \delta_i^2$, r is the rank of matrix D. In practical cases, ρ may be approximated by $\delta_{k+1}^2/2$, and r by n.

2.2 Algorithm

2.2.1 Setting Up Retrieval System

1. Text preprocessing: Identify words and noun phrases as well as stop words.
2. System term construction: Set up the term list as well as the global weights.
3. Set up the document vectors of all the collected documents in normalized form .
4. Construct interior differential term-document matrix $D_I^{m\times n_1}$, such that each of its column is an

interior differential document vector.
5. Construct an exterior differential term-document matrix $D_E^{m \times n_2}$, such that each of its column is an exterior differential document vector.
6. Decompose D_I and D_E by SVD (singular value decomposition) algorithm into USV form. Find proper values of k's to define the likelihood functions $P(x|D_I)$ and $P(x|D_E)$ as Equition (1).
7. $P(D_I|x) =$

$$\frac{P(x|D_I)P(D_I)}{P(x|D_I)P(D_I) + P(x|D_E)P(D_E)},$$

where $P(D_I)$ is set to an average number of recalls divided by the number of documents in the data base and $P(D_E)$ is set to $1 - P(D_I)$.

2.2.2 Patent Document Search

1. A query is treated as a document; a document vector is set up by generating the terms as well as their frequency of occurrence, and thus a normalized document vector is obtained for the query .
Each document in the data base are processed by the procedures in items 2-5 below.
2. Given a query, construct a differential document vector x .
3. Calculate the interior document likelihood function $P(x|D_I)$, and calculate the exterior document likelihood function $P(x|D_E)$ for the document.
4. Calculate the Bayesian posteriori probability function $P(D_I|x)$.
5. Select those documents whose $P(D_I|x)$ exceeds a given threshold (say, 0.5), or choose N documents having the first N largest $P(D_I|x)$.

3 Template Structure for Storing Patent Abstracts

Each patent document is usually provided with an abstract. The abstract can be used for content-based information retrieval by using DLSI method as described above. As we have mentioned before, the content-based information retrieval system by LSI analysis is not robust enough to be directly applicable to a real system. We will use the DLSI method only to narrow down the search space at a first stage of filtering in information retrieval. We will resort to a form based searching strategy to pin down the patent document.

Now that the content-based DLSI search scheme has narrowed down the search space in content, the form based search strategy we now employ need not to pay attention to the synonymous expressions of the searching terms or sentences.

This first stage of filtering is now implemented without going through the tedious process of dealing with the synonymous expressions by synonym dictionaries which are hard to develop and to use. Even if we succeeded in treating the synomyms, we also have to realize that the polynonym of a natural language will reduce the advantage of using synonym dictionary further, because two words are synonymous in one situation but might not be so in other situations, depending on context.

In view of lengthy sentences used in patent documents including their abstracts, we want to emphasize that automaton-based template structure is an extremely efficient way of expressing lengthy sentences with their synonymous expressions.

We will demonstrate this point by way of examples below. For a sentence, “There are beautiful parks in Japan across the nation”, we can use a template as of figure 1 where a variety of synonymous expressions are explicitly represented.

The problem here is, how we could get the template for an abstract of patent document? Firstly, we regard the original abstract of patent itself as a simplest template. Then, we register queries into the matched template structures by combining each pair of matched terms into one node. This is illustrated by an example procedure in figures 1-3. The original template of an abstract is indicated by figure 1, but when a query of figure 2, namely, “There are lovely parks across Japan”, is matched to the template of figure 1, the template could be modified to a new structure of figure3.

Suppose that the query sentence is, “There are ugly streets in Japan”. Now although we could locate a matching pattern similar to that of figure 2, we will have to rule it out so that we will not come up with a template which include the above sentence as a path, or part of a path . This mechanism should be established from users’ response. We will explain it in Section 4.1.

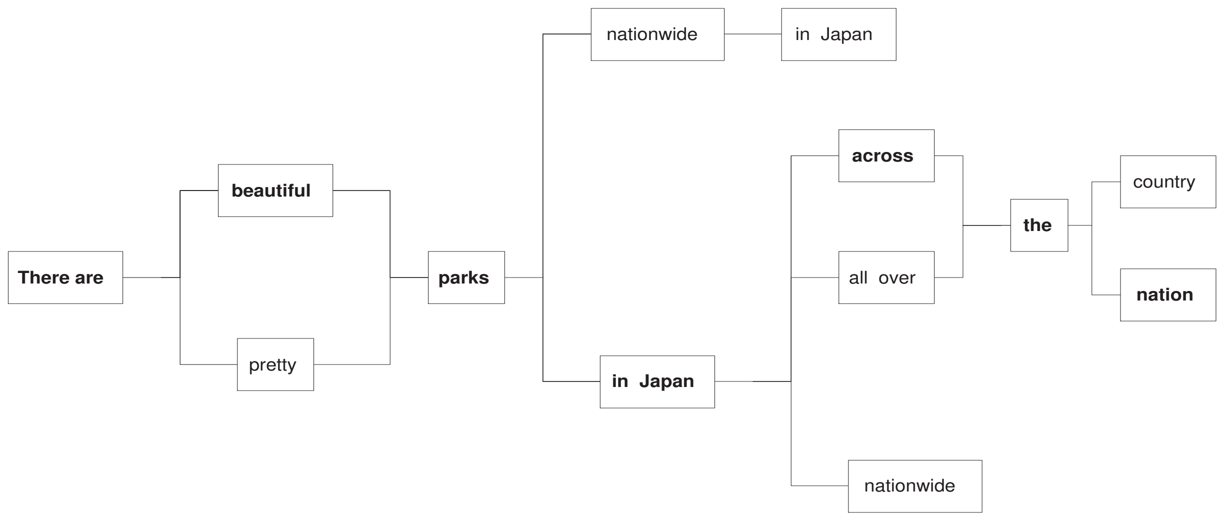

Figure 1: Template Example Indicating a set of Semantically Similar Patent Abstracts

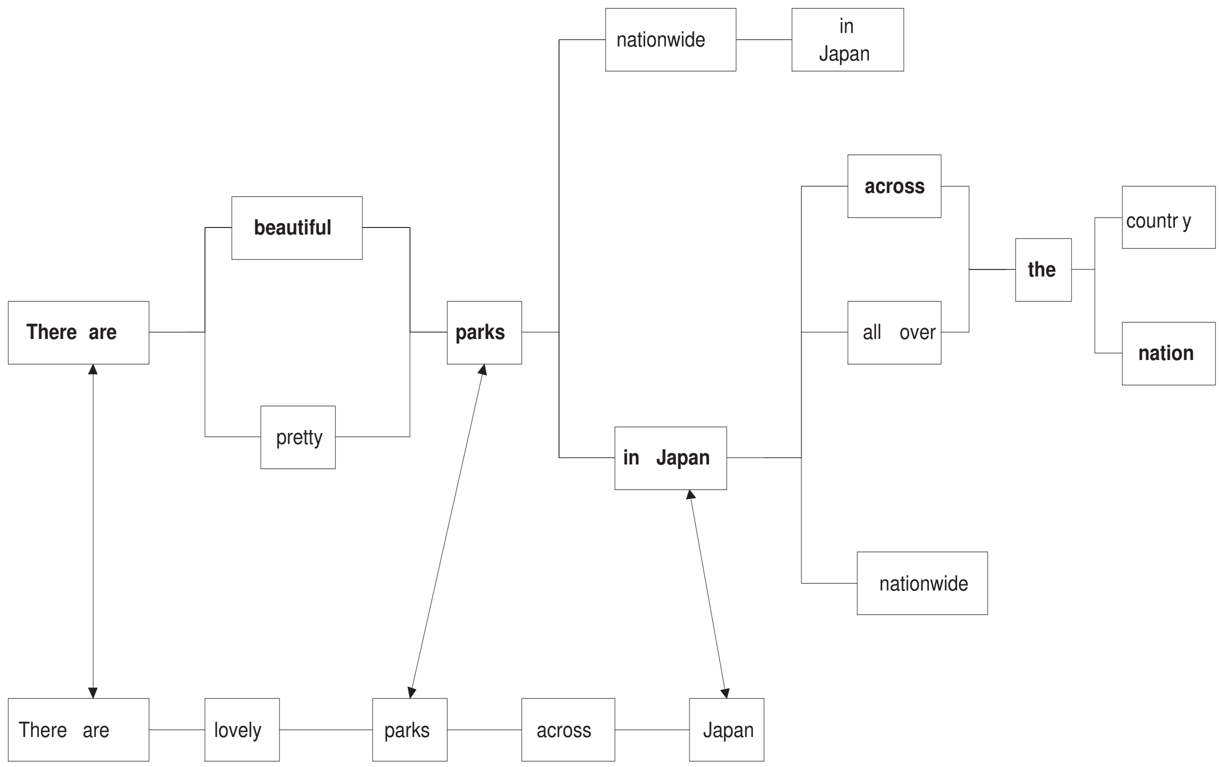

Figure 2: Query Template to be matched with Abstract Template

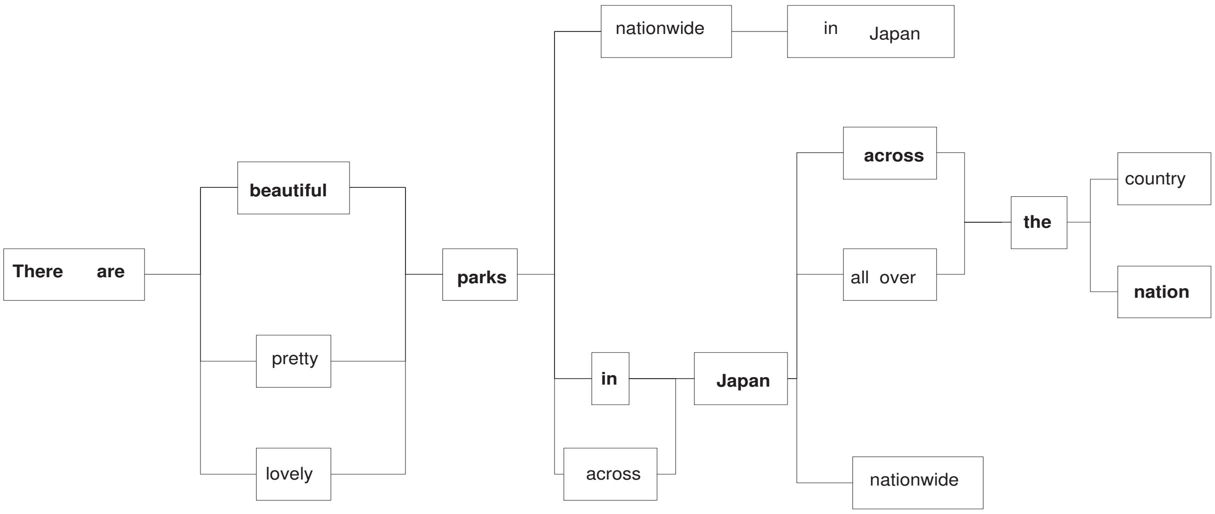

Figure 3: Modified Template

4 The Flow of the Search Process

4.1 The Entire Flow of the Complete Search Process

Before starting the search process, we should set up the DLSI for all the patent documents.

1. Locate the query in the DLSI space.
2. Find and select those patent documents whose abstracts' vector space lie in a neighborhood of the query vector space having semantic similarity to sentences of figure 1 by the DLSI matching algorithm.
3. For each of the abstracts obtained by step 4.1, use the template matching algorithm of (Chen and Tokuda, 2003a) to calculate the similarity of the summary and the query, select the documents of which the abstracts have a highest similarity to the query.
4. Show the result to the user.
5. Modify the abstracts in the database by users' responses.

5 Concluding Remarks

We have proposed a new IR method for patent documents addressing both semantic and syntactic properties by combining a mixed model of content and form based methods; the first stage of DLSI method narrows down the search space by content and the second template method pins down the document by syntactic search on words. We are able to do so, mainly because the DLSI matching in the first stage captures those documents based on content while the template method can now pin down the patent documents having a highest similarity in form with the query. An experimental verification of the present approach is now underway.

References

M. W. Berry, Z. Drmac, and E. R. Jessup. 1999. Matrices, vector spaces, and information retrieval. *SIAM Rev.*, 41(2):335–362.

L. Chen and N. Tokuda. 2003a. Bug diagnosis by string matching: Application to ILTS for translation. *CALICO Journal*, 20(2):227–244.

L. Chen and N. Tokuda. 2003b. Robustness of regional matching scheme over global matching scheme. *Artificial Intelligence*, 144(1-2):213–232.

L. Chen, N. Tokuda, and A. Nagai. 2001. Probabilistic Information Retrieval Method Based on Differential Latent Semantic Index Space. *IEICE Trans. on Information and Systems*, E84-D(7):910–914.

M. L. Littman, Fan Jiang, and Greg A. Keim. 1998. Learning a language-independent representation for terms from a partially aligned corpus. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 314–322.

N. Tokuda and L. Chen. 2001. An online tutoring system for language translation. *IEEE Multimedia*, 8(3):46–55.

M. Turk and A. Pentland. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Justin Zobel and Alistair Moffat. 1998. Exploring the similarity space. *ACM SIGIR FORUM*, 32(1):18–34.