Adopting the Information Retrieval Approach for Storing and Retrieving Thai-text Structured Data

Marut Buranarach and Choochart Haruechaiyasak
Human Language Technology Laboratory (HLT)
National Electronics and Computer Technology Center (NECTEC), Thailand Science Park,
112 Phahon Yothin Rd., Klong Luang, Pathumthani 12120 Thailand
Tel. 02-564-6900 Fax. 02-564-6873
E-mail: {marut.buranarach, choochart.haruechaiyasak}@nectec.or.th

ABSTRACT – This paper describes an approach of using full-text search engine in storing and retrieving structured data in Thai language. It discusses some limitations of database management system (DBMS) in querying Thai full-text based content. These limitations can result in degrading of retrieval performance both in terms of result accuracy and system response time. Information Retrieval (IR) system or search engine typically focuses on storage and retrieval techniques for unstructured data, i.e. documents. In this paper, we advocate a less common use of search engine to store and retrieve Thai-text structured data to overcome the limitations found in DBMS. The technique is elaborated in three major steps. We present a case study on the development of a Thai patent search service, in collaboration with the Department of Intellectual Property, Ministry of Commerce. The search prototype was built using Sansarn Look!, a system platform for developing IR systems for Thai language that has been developed at NECTEC. Based on the initial testing on the system, the retrieval results in terms of accuracy and response time are improved over the DBMS approach.

KEY WORDS -- Search Engine, Information Retrieval, Structured Data, Patent Search, Thai-Language Processing

1. Introduction
Structured data is a form of data that is normally represented in terms of records. A record may consist of multiple field-value pairs. For example, a record of “Student” may contain a field of “Firstname” with a value of “สมชาย” along with other field-value pairs (Figure 1). Structured data is often stored in Relational Database Management System (RDBMS), which utilizes SQL as the query language.

Retrieval of text data in DBMS is most effective when text string stored in each field is short. When a field contains short text strings, DBMS allows the data in the field to be indexed based on some efficient data structure, e.g. B+-Tree [1]. However, when the stored text strings are longer than a size limit, the indexing
technique can not be applied and must be disabled. The size of text string that can be indexed using B+-Tree is generally limited by disk block size. Thus, searching fields that stores long text strings in DBMS requires combination of sequential search and string pattern matching technique which results in longer processing time.

<table>
<thead>
<tr>
<th>Student Record 1</th>
<th>Student Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Firstname:</strong></td>
<td><strong>Firstname:</strong></td>
</tr>
<tr>
<td>สมชาย</td>
<td>สมศักดิ์</td>
</tr>
<tr>
<td><strong>Lastname:</strong></td>
<td><strong>Lastname:</strong></td>
</tr>
<tr>
<td>ใจดี</td>
<td>แสนสุข</td>
</tr>
<tr>
<td><strong>Birthyear:</strong></td>
<td><strong>Birthyear:</strong></td>
</tr>
<tr>
<td>2530</td>
<td>2529</td>
</tr>
</tbody>
</table>

**Figure 1 Example of structured data**

Using DBMS in storing and retrieving text in Thai language has a limitation in terms of retrieval accuracy. Thai language is considered as a non-segmented language in which words are continuously written without the use of explicit word-ending marker. The string pattern matching technique used in DBMS when applied to Thai text can sometimes result in matches of non-word tokens. For example, a query “รถ” (“Car” in English) can result in the retrieval of the values containing the word “สามารถ” (“To be able to” in English). The two words have no semantic relations but share some common string of characters. The same can be said for “นก” (“Bird”) and “กนก” (a person name). The problem was caused by the retrieval technique used in DBMS which processes a text value as one long string of characters and does not distinguish between word and non-word tokens. This can lead to lower accuracy of the retrieval results.

2. Information Retrieval Approach to Storing and Retrieving Thai-text Structured Data

2.1 Information Retrieval

Information Retrieval (IR) [2] or full-text search is another approach of searching the information used by IR system or search engine. It mainly focuses on the storage and retrieval techniques for unstructured data, i.e. documents. Web search engines, e.g. Google.com, are examples of IR system that helps users locate information from Web documents.

The major technique used in IR system is full-text indexing. Using this technique, an IR system automatically indexes documents by terms they contain. The index terms are associated with list of documents containing them. This information is typically stored in an inverted file. Searching is the process of matching user query with index terms in the inverted file. The list of documents associated with the matched terms are ranked and returned to the user.

The indexing technique used for Thai texts requires an additional step. When indexing Thai texts, word segmentation technique must be used in preprocessing a text string into multiple terms. Figure 2 shows an example of constructing inverted file for Thai-text documents [3].

**Figure 2 Example of constructing an inverted file**

2.2 Sansarn Look!

Sansarn Look! [4] is a system platform for developing IR systems for Thai language that has been developed at NECTEC [5]. The development of Sansarn-Look! was based on Lucene [6], an open source Java-based IR library. It consists of four major components: collector, parser, indexer and searcher. The indexer contains a Thai analyzer software module which enhances the processing of Thai-text data. Sansarn Look! provides a Web-based admin interface that helps to simplify the administration and management tasks of a search engine. It also provides a wide-ranged support for multiple file formats and collector configurations. Many Thai-related features, e.g. word suggestion and word correction, are included in the system. The layered architecture of Sansarn Look! is shown in Figure 3.

**Figure 3 Sansarn Look! layered architecture**
2.3 Storage and Retrieval of Thai-text Structured Data using Sansarn Look!

The process of storing and retrieving Thai-text structured data using IR system can be divided into three major steps. The first step involves creating denormalized table. The second step involves choosing indexing methods for each data field. The final step involves applying various querying methods in retrieving the data. The steps described below assume that original data is stored in DBMS.

2.3.1 Creating Denormalized Table

Storing records using Sansarn Look! is similar to but is simpler than in RDBMS. Sansarn Look! provides support for creating index for multiple data fields. Thus, data stored in a single database table with multiple fields can be straightforwardly exported and indexed by IR system utilizing Sansarn Look!.

RDBMS normally supports storing data in multiple tables that referenced to each other. When original data is stored in multiple tables, exporting to the IR system requires an additional step. In particular, the tables must be joined into a single table in order to allow them to structurally match with that required by the IR system. This process is called Denormalization. In RDBMS, view can be created to join different fields from multiple tables into single table. This table can be used as the basis for the IR system input. Figure 4 shows an example of creating denormalized table.

![Figure 4 Example of creating denormalized table](image)

2.3.2 Choosing Indexing Methods

The second step involves choosing indexing method for each data field. Each data field can have different types and purposes. Thus, each field may require different indexing technique. For example, a “Province” field usually contains short text strings and frequently searched by exact match. This type of field normally does not need word-segmentation preprocessing. In contrast, an “Abstract” field, which contains summary of articles, usually contains long text strings with multiple words and sentences. This type of field often needs indexing technique which combines word-segmentation preprocessing. Adopting inappropriate indexing technique for a data field can result in a less than optimal retrieval performance.

Table 1 Summary of indexing types

<table>
<thead>
<tr>
<th>Type</th>
<th>Word-segmented</th>
<th>Indexed</th>
<th>Stored</th>
<th>Example Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>Document content, title</td>
</tr>
<tr>
<td>Keyword</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>Person name, Number, Date</td>
</tr>
<tr>
<td>Unstored</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>Large document content</td>
</tr>
<tr>
<td>Unindexed</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>Person title (e.g. Mr., Miss, Mrs.)</td>
</tr>
</tbody>
</table>

Sansarn Look! provides four indexing types in storing data fields: Text, Keyword, Unstored and Unindexed. Each is different in terms of whether and how data will be pre-processed and stored in the IR system. They are summarized in Table 1.

Text and Keyword are generally the most frequently used indexing types. Thai-text data stored using Text type will be pre-processed for word segmentation before it is indexed and stored in the system. The method is most useful for storing long text description containing multiple words/sentences, e.g. document abstract, title. Using Keyword type, data is indexed and stored in the system without data preprocessing. The method is most suitable for short, non-dictionary words, e.g. person names, province names, etc., numbers, URLs, and dates. Unstored type is similar to Text except that only index will be stored but not actual data. The method is most suitable for searching from very large text, e.g. document, book, web page content. Using Unindexed type, data is stored as is, i.e. no indexing. The method is most suitable for small and non-searched fields, e.g. person title, middle name, etc., which are primarily only used for displaying purpose.

2.3.3 Choosing Querying Forms

The final step involves querying the data. Sansarn Look! provides support for various querying forms. These include term, range, Boolean, and wildcard queries.

Term query is the simplest form of query, in which query term is matched with index terms in the searched field. For example, the query “Abstract:นิยม” returns the records containing the word “นิยม” in the Abstract field. Range query matches range of index terms by specifying begin and end terms. For example, the query “Year: [1975 TO 1985]” returns those containing the values from 1975 to 1985 in the Year field. Boolean query allows use of logical connectors, i.e. AND, OR, NOT. For example, the query “Abstract:นิยม AND Year: [1975 TO 1985]” returns those satisfy both search conditions. Wildcard query matches query term with some parts of index terms.
The symbol “?” can be used to match 0 or 1 character while the symbol “*” can be used to match 0 or more characters. For example, the query “Title:?ลา*” returns the records containing the words “ลา”, “ปลา”, “หลา”, or “ปลายข้าว” in the Title field.

Other query forms supported by Sansarn Look! include prefix, phrase, and fuzzy queries. Further information about these query forms can be found in [7].

3. Case Study
We have developed a prototype for Thai patent search service using Sansarn Look!. The prototype was resulted from the collaboration between NECTEC and the Department of Intellectual Property (DIP) [8], Ministry of Commerce. The system was designed to provide the users with fast and comprehensive access to the Thai patent information. The patent database contains the information about the issued Thai patents such as title, date of issue, inventor name, International Patent Classification (IPC) number, and abstract.

Sansarn Look! was chosen over DBMS in supporting the storing and retrieving of the Thai patent data. The decision was made due to several reasons. First, most of the Thai patent data are in Thai language with several fields contain long text strings, e.g., abstract and copyright statements. Second, the issued patent data are mostly static and not updated. Based on these characteristics, using DBMS is less suitable compared to IR system. Using DBMS, searching over long text fields can be slow. In addition, the string matching technique used in DBMS can result in lower accuracy when querying Thai-text content.

The development of Thai patent search service was done in three major steps. First, data was exported from the original patent database. In order to prepare the input data for Sansarn Look!, the data was exported into a single denormalized table which contains all the needed fields. Second, we applied the indexing types appropriate for each field. For example, fields applying Text indexing type include title, abstract and copyright statement fields. Fields applying Keyword indexing type include inventor name, issued date, IPC number, etc. Finally, a Web-based interface was developed to provide flexible search options in retrieving the information from the patent corpus.

The Thai patent search prototype is shown in Figure 5. Searching was allowed in two modes: quick search and detail search. Using detail search, multiple search conditions can be combined in a Boolean query using AND logical connector. Searching by date period utilizes range query. Searching by IPC numbers and person names utilizes wildcard query. Search results can be ranked based on relevance score or date of issue.

4. Evaluation
The prototype IR system was evaluated in terms of response time and result accuracy. The performance of the IR system was compared with a DBMS, i.e. PostgreSQL 8.2 [9]. The evaluation was conducted in a system with Pentium M 1.8GHz CPU, 512 MB RAM, Windows XP platform.

4.1 Response Time
In evaluating response time, the patent data were reproduced in 5 sets with progressively increasing sizes (23K, 47K, 70K, 94K, 117K records). The data were then imported into the IR system and the DBMS respectively. Two test queries were formed with the search condition on the “Abstract” field, which should result in longer response time comparing to other fields. One query term is a non-word term with the length of five characters that returns not found result. Another is the term “พลาสติก”, which returns 436 results. The system response time for each query term was measured three times. The average response time of the two queries measured for each test set was plotted as shown in Figure 6.
The result showed that average response time of the IR system was improved over the DBMS. In particular, the response times of the IR system for the test sets were between 0.05-0.18 seconds. The response times of the DBMS were between 7.7-18.6 seconds with higher growing rate.

4.2 Result Accuracy

In evaluating retrieval accuracy, five query terms were used, i.e. ‘ไหม’ (silk), ‘จอ’ (monitor), ‘ราว’ (handle), ‘ปลา’ (fish), and ‘ข้าว’ (rice). These terms were chosen because they can potentially appear as parts of other terms. The terms such as ‘รถ’ (car) and ‘นก’ (bird) were not used because they returned too many results. These five query terms were used in the search condition on the “Title” field against the patent data that are stored in both types of system. The retrieval results from both systems were judged for their relevance to the queries. Result accuracy was measured and plotted as shown in Figure 7.

![Figure 7 Result Accuracy (IR System vs. DBMS)](image)

The result accuracy of the IR system was improved over the DBMS. In particular, the average result accuracy of the IR system over the test queries was 97%. The average result accuracy of the DBMS was 58%.

5. Conclusion

Based on the initial testing, using the IR approach in storing and retrieving Thai-text structured data has the advantages of having better response time and result accuracy over the DBMS. The improvement in response time is particularly critical when querying fields storing long text strings. The improvement in result accuracy is particularly critical when using short query terms which can appear in text string of some other terms.

There are some disadvantages of using the IR approach. First, retrieval performance of the IR technique described in this paper partly depends on dictionary, which affects performance of word segmentation. More comprehensive dictionary will result in better performance of word segmentation, which can subsequently improve retrieval performance. Second, the IR approach stores data in denormalized table. This makes the technique more suitable for data that is static and not updated.

Some future works include more evaluation of retrieval performance under different test cases. Integration with some advanced Thai features such as Soundex search will further improve retrieval effectiveness of the system.

References