Managing Offline Educational Web Contents with Search Engine Tools

Choochart Haruechaiyasak[1], Chatchawal Sangkeettrakarn[1], and Wittawat Jitkrittum[2]

[1] Human Language Technology Laboratory (HLT),
National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
{choochart.haruechaiyasak,chatchawal.sangkeettrakarn}@nectec.or.th
[2] School of Information and Computer Technology (ICT),
Sirindhorn International Institute of Technology (SIIT),
Thammasat University, Bangkadi Campus, Pathumthani 12000, Thailand

Abstract. In this paper, we describe our ongoing project to help alleviate the digital divide problem among high schools in rural areas of Thailand. The idea is to select, organize, index and distribute useful educational Web contents to schools where the Internet connection is not available. These Web contents can be used by teachers and students to enhance the teaching and learning for many class subjects. We have collaborated with a group of teachers from different high schools in order to gather the requirements for designing our software tools. One of the challenging issues is the variation in computer hardwares and network configuration found in different schools. Some shools have PCs connected to the school's server via the Local Area Network (LAN). While some other schools have low-performance PCs without any network connection. To support both cases, we provide two solutions via two different search engine tools. These tools support content administrators, e.g., teachers, with the features to organize and index the contents. The tools also provide general users with the features to browse and search for needed information. Since the contents and index are locally stored on hard disk or some removable media such as CD-ROM, the Internet connection is not needed.

Keywords: Web Content Management, Search Engine Tools, Educational Web Contents.

1 Introduction

With its exponential growth rate, the World Wide Web is now well recognized as the world's largest online information resource. Today there are many Web sites which provide excellent educational contents. Some of the interesting examples include BBC's Learning [5], MIT's OpenCourseWare [11], Wikipedia [12] and Google Book Search's Library Project [7]. BBC's Learning is a Web portal that provides links to various learning subjects for different audience ranging

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 444–453, 2007.

from kids to adults. MIT's OpenCourseWare Web site provides a free and open educational resource (OER) for educators, students, and self-learners around the world. The contents are course materials such as presentation slides and publications used for teaching many different classes at MIT. Wikipedia is a free well-known online encyclopedia. The outstanding feature of Wikipedia is the use of collaborative concept in which each user is allowed to edit and share the definitions and contents of the posted articles. Currently there are over *1,600,000* articles available in English and many in other languages. Google's Library Project has collaborated with many universities in order to scan and export contents of many books into digitized format. Users are then able to search and retrieve the book contents which match their interests. Besides these examples, there are many other Web sites which provide great information and knowledge and are publicly available.

These information and knowledge resources are however only accessible by users who could connnect to the Internet. Today, the problem known as *digital divide* still exists among people who live in rural or remote areas, especially of developing countries. The digital divide problem can generally be described as the lack of computer equipments, network infrastructure, and knowledge in IT. Therefore, our main goal is to bridge this gap for the users who have difficulties in accessing the Internet. In this paper, we focus on high schools in rural areas of Thailand. Most of these schools do not have the Internet connection. Some schools do have the Internet connection via satellite communication. However the available bandwidth is very limited, thus making the Internet usage impractical. To use the bandwidth efficiently, the teachers often save and share useful Web contents by using a Web browser or some downloading software. However, these software tools do not provide enough functions to help users organize, index and search the contents.

To design some suitable solutions, we have discussed with a group of teachers from many high schools in different regions of Thailand. One of the challenging issues that we discovered is the variation in computer hardwares and network configuration found in different schools. Some schools have low-performance PCs with limited hard disk capacity. These computers are often donated by some universities and organizations. Some fortunate shools have PCs connected to the server via the Local Area Network (LAN) configuration. While some other schools have individual PCs without any form of network connection. To support these cases, we provide two solutions via two different search engine tools: *Sansarn Look!* and *Sansarn Offline*. Sansarn Look! was designed as a Web-based application and is therefore suitable for the client-server model. Sansarn Look! allows Web content and its index to be stored on the server. Users can use a Web browser to browse and search for the content via LAN. On the other hand, Sansarn Offline was designed as a stand-alone application and is therefore suitable for stand-alone PCs without any network connection. Web content and its index are stored on removable media such as CD-ROM. Users can retrieve the content through the interface of the program.

Other important design issue is to provide and distribute the software tools to interested users without any licensing fee. Therefore, our tools are based on the open-source software concept. With the thorough survey of the open-source Information Retrieval (IR) libraries, we found *Lucene* to be a suitable choice for implementing our search engine tools. Lucene is one of the most widely used IR library currently maintained under the Apache project [10]. Lucene provides the core indexing and searching functions which are very efficient and scalable [1]. The key success to the high scalability is the intelligent algorithm in managing the inverted index files between the memory and the secondary storage. The users may set the number of index segments and their sizes according to the system specification in order to optimize the indexing and searching performance. Lucene library, however, only comes with the support of English language. To extend the library to support other languages, developers must add language-specific analysis package into the library. To make Sansarn Look! and Sansarn Offline support both English and Thai texts, we have developed a Thai-language analysis package, *ThaiAnalyzer*, which is integrated into the Lucene package. Integrating and extending Lucene library is very simple and easy, since its creator has designed the framework via the object-oriented programming concept of Java [2]. This object-oriented feature is another reason that makes Lucene a very attractive choice for implementing our tools.

To demonstrate our search engine tools, we have asked a group of volunteer teachers to help select, organize and index some class-related Web contents. The contents and its index will be distributed to high schools in rural areas where the Internet connection is slow or not available. Students can search for useful information from either server's hard disk or some CD-ROMs as if they are connected to the Internet. After the tools and the contents are distributed, we will collect the feedbacks from the teachers and students in order to improve the features for the next version of the tools.

The remainder of this paper is organized as follows. In next section, we explains a case study of designing solutions to support different hardware and network configurations in high schools. Section 3 gives details on Lucene library and the implementation of ThaiAnalyzer package. Section 4 and 5 gives the discussion on the system design and architecture of Sansarn Look! and Sansarn Offline, respectively. Section 6 gives the conclusions and future work.

2 A Case Study of Managing Educational Web Contents for High Schools

One potential application of our search engine tools is to manage educational Web contents and distribute them to schools which have slow or no Internet connection. As mentioned earlier, we have found from the survey and discussion with many teachers that the hardware and network configuration is extremely varied among different schools. Therefore, we have designed our solutions to support all different use cases.

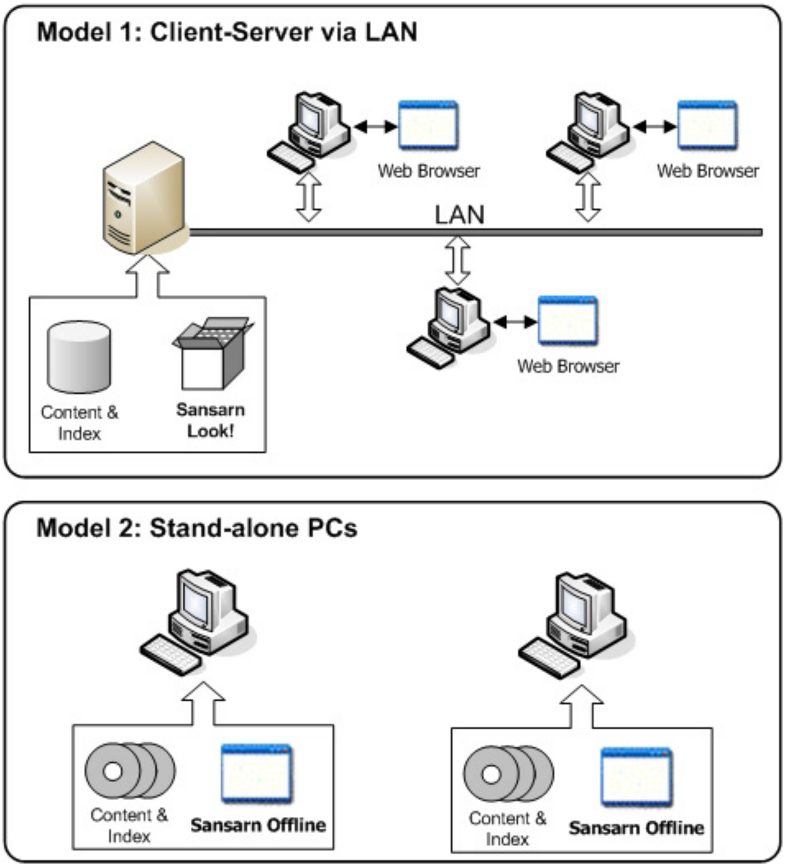

Fig. 1. Hardware and network configuration models

Figure 1 illustrates two solutions according to different hardware and network configuration models. The first configuration model is a typical client-server via LAN. In this model, the administrator installs Sansarn Look! on the server and uses the tool to collect and index the contents. Users on the client PCs can use a Web browser to connect to the server via the *localhost* URL. A user can perform the full-text search by entering query on the Web browser interface. The query is sent to the local server where the search result is compiled and returned to the user. This model offers scalability, sharability and ease of maintenance. Since contents are organized and indexed on the hard disk of the server, they can easily be updated and shared among many users at the same time.

The second configuration model is the stand-alone PCs. Each PC must install Sansarn Offline program. The contents with the index must be recorded and distributed via removable media such as CD-ROM. This model offers great reliability since the query and search results are performed on local PC. Another advantage is that the response time depends only on the I/O activity of the computer which is generally more reliable than the network.

We have asked a group of volunteer teachers to select and organize the Web contents according to the class materials taught in high schools. Typical learning contents are composed of Web pages with some image and other multimedia files such as Macromedia Flash. In Thailand, classes are organized into different subjects such as Thai literatures, English language, physics, biology and chemistry. Web contents often offer better learning approach for students because they are more dynamic than materials available from textbooks. In some contents related to physics, an animation simulating projectile movement can be replayed to help the students understand the concept better. Therefore, it is quite obvious that by offering these offline educational contents to high schools, both teachers and students can benifit more from teaching and learning the class subjects than using only textbooks.

3 Lucene Library with ThaiAnalyzer Package

Lucene was designed and implemented based on the object-oriented programming framework of Java. Figure 2 illustrates the *Lucene* API which is organized into the following four packages.

- *Document*: In Lucene, a text is organized as a document which represents a collection of fields. Before a text document could be indexed, it must be created by constructing a new instance of *Document* class. The seach results are also returned in the form of *Documents*.
- *Index*: Index contains many different classes responsible for indexing process. *IndexWriter* is the main class whose task is to create a new index and add documents into an existing index.
- *Search*: Search contains classes related to search operations. *IndexSearcher* is the main class whose task is to search within the given index directory for matching documents.
- *Analysis*: Analysis contains classes for performing the text processing. Different *Analyzer* classes are provided for various specific tasks such as stopword removal and stemming which are suitable only for English language. To make the analysis applicable to Thai language, we developed *ThaiAnalyzer* package whose basic task is to tokenize Thai texts into a set of words [3].

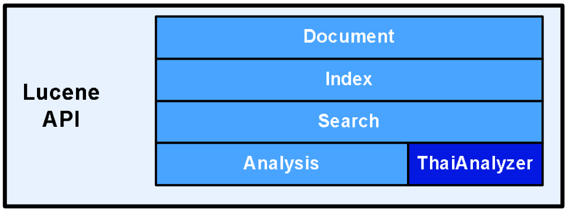

Fig. 2. Lucene API with ThaiAnalyzer package

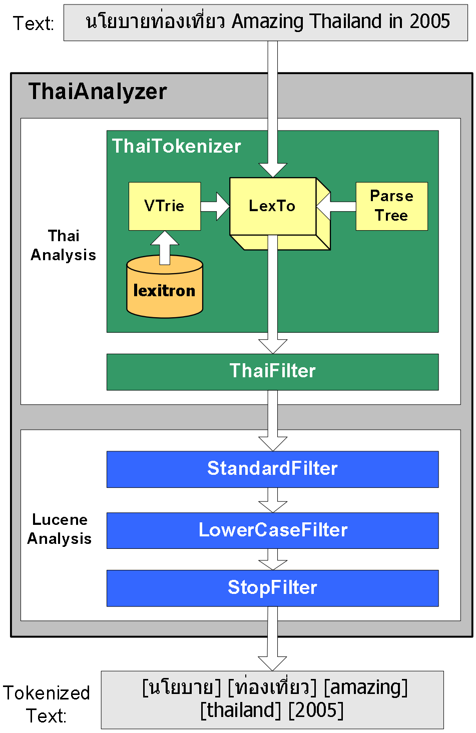

Fig. 3. ThaiAnalyzer package

Figure 3 illustrates the work flow within the ThaiAnalyzer package. ThaiAnalyzer consists of two main analysis packages: Lucene and Thai. Lucene analysis package is originally provided with the library. Thai analysis package is implemented for tokenizing Thai texts. *ThaiTokenizer* is the main class in Thai analysis package which performs the Thai-text tokenization. Our tokenizing algorithm is based on the use of a dictionary, i.e., *LEXiTRON*[9]. To improve the speed of dictionary look-up, words from dictionary are stored by using the *trie* data structure. During the tokenizing process, a text is segmented by looking up the dictionary set stored in the trie. Given a text, *ThaiTokenizer* will perform tokenizing process on Thai texts. English segments and other special characters will be filtered via *ThaiFilter* class. Those English texts and special characters will be further handled by *StandardFilter*, *LowerCaseFilter* and *StopFilter*. LowerCaseFilter will normalize the English token to lower-case characters while StopFilter will remove English stopwords to improve the space efficiency.

4 Sansarn Look! for Client-Server Via LAN Configuration

Sansarn Look! was designed as a Web-based application and is therefore suitable for the client-server model. Sansarn Look! allows Web content and its index to

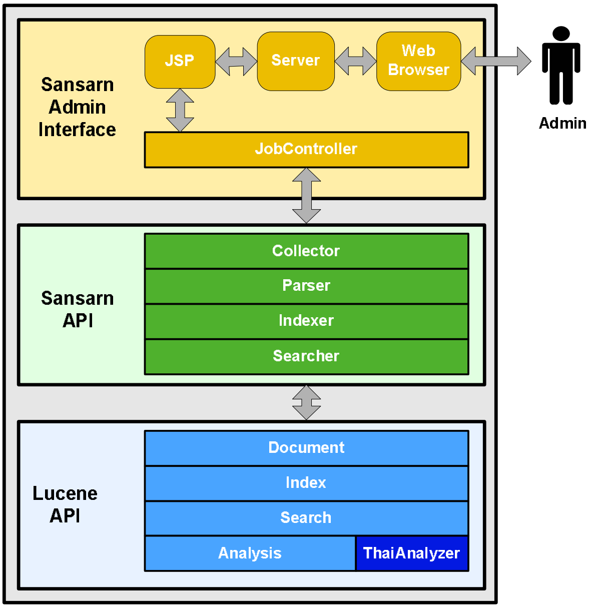

Fig. 4. Sansarn Look!: system architecture

be stored on the server. Users can use a Web browser to browse and search for the content via LAN. Figure 4 illustrates the system architecture of *Sansarn Look!*. The system is composed of three layers: Lucene API (LAPI), Sansarn API (SAPI) and Sansarn Admin Interface (SADI). The source codes in this project is written in Java which provides the platform-independent feature that we need. LAPI is the layer which provides two fundamental IR functions: indexing and searhing. The details of LAPI were given in the previous section.

On top of LAPI is the SAPI which provides the integrated packages by making use of LAPI layer. SAPI is composed of the following four packages.

– *Collector*: To provide flexibility, the *Collector* package contains various classes for collecting documents from either remote servers or local file system. For remote documents, i.e., Web pages, the collector functions exactly as a Web crawler.
– *Parser*: This package contains classes for removing HTML tags or pre-specified XML tags.
– *Indexer*: This package contains classes which perform indexing functions. Indexer package contain higher-level classes which make use of the *Index* package from LAPI.
– *Seacher*: This package contains classes which perform searching functions. *Searcher* contain higher-level classes which make use of the *Search* package from LAPI.

The topmost layer of the platform is the SADI which is composed of necessary components and interfacing functions between users and the system. *JobController* provides classes for managing all related processes of the system. Our system is run on a server called *JBoss* which is an open-source, well-designed and well-implemented technology [8]. To provide an easy-to-use interface, we adopt the *Java Server Page* (*JSP*) technology which could be integrated seamlessly with any Web browser. Therefore, once installed, users may use a Web browser to control all processes of the system. Sansarn Look! is available for free download from our Web site at *http://sansarn.com/look*.

5 Sansarn Offline for Stand-Alone PCs Configuration

Sansarn Offline was designed as a stand-alone application and is therefore suitable for stand-alone PCs without any network connection [4]. The process of using Sansarn Offline consists of two main tasks: content administration and content retrieval. For the first task, content administrator, i.e., teachers, will use the tool to organize and index Web contents into some predefined categories. The contents and its index are then recorded into some removable media such as CD-ROM. The second task involves user entering search query through the interface. The tool then looks up from the index and returning search results to the user. Since the contents and their indexes are stored on the removable media, the retrieval process can be done without the Internet connection. Another advantage is that the tool could yield search results which are more focused than the ones obtained by using regular search engines. This is because the content administrator would select, filter and collect only high quality Web contents which match the users' information need.

Figure 5 illustrates the overall architecture of *Sansarn Offline*. The system consists of two functions: *content administration* and *content retrieval*. For the first function, there are four modules which support the content administration function: Collector, Parser, Indexer and Categorizer. The first three modules are the same as used in Sansarn Look!. The categorizer module contains classes which perform automatic categorization of collected contents into predefined subjects or categories. The outputs from the first phase are data contents with the index which are then recorded into some removable media.

The second *content retrieval* function consists of two modules which support the retrieval of contents for the users. The searcher module is the same as used in Sansarn Look!. Search Interface module provides a GUI-based interface which connects to the searcher module in order to pass the query from the users and to return the search results back to the users.

Sansarn Offline is implemented in Java by using Eclipse tool [6]. Using Java offers several advantages including platform-independent and open-source development. Sansarn Offline allows users to browse according to the organized subjects and to perform the full-text search on the contents. The returned search results show the snippets with the highlighted terms similar to typical search engines. User can click on the link to view the full page content. Our tool also

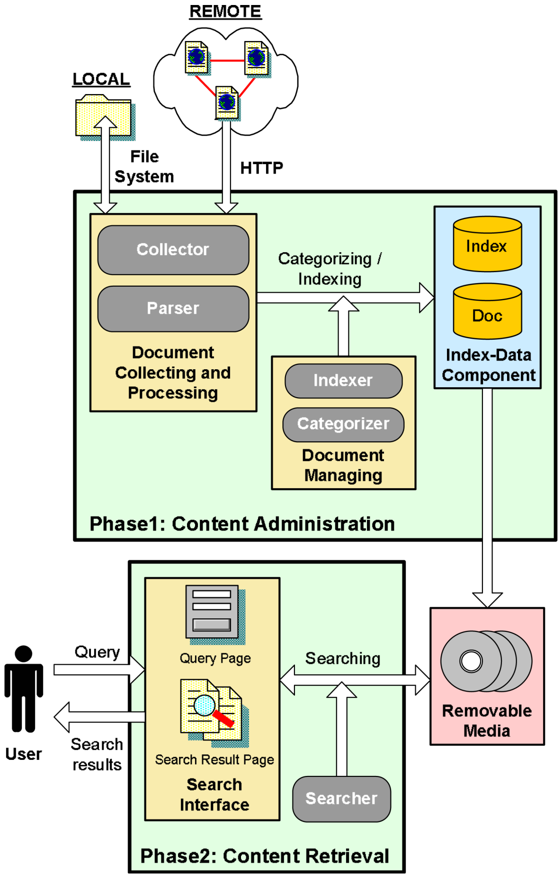

Fig. 5. Sansarn Offline: system architecture

allows the users to set some configuration such as the color used to highlight the terms in the results and number of results shown per page. More details of Sansarn Offline can be found from our Web site at *http://sansarn.com/offline*.

6 Conclusions and Future Work

We proposed two search engine tools: *Sansarn Look!* and *Sansarn Offline* to support managing educational Web contents among high schools with different hardware and network configurations. Both softwares are freely avaiable and open-source which were developed by using *Lucene* IR (Information Retrieval) library. To support the retrieval of Thai texts, we developed a Thai-language analysis package called *ThaiAnalyzer* whose task is to segment Thai written texts into word tokens. Sansarn Look! was designed as a Web-based application and is therefore suitable for the client-server model. Sansarn Offline was designed as a stand-alone application and is therefore suitable for stand-alone PCs without any network connection. Both tools allow searching for contents stored on either server's hard disks or from the removable media, therefore the retrieval process could be done offline or without the Internet connection.

Our future work includes the plan to select, index and distribute more useful copyright-free Web contents. One of the potential contents is the Wikipedia Selection for Schools [13] This Selection is about the size of a *15* volume encyclopaedia with *24,000* pictures, and articles on *4,625* topics. The articles have been cleaned up and checked for suitability for children. However, the current version of the content can only be accessed by browsing on subject index and title word index. By using our tools, users can perform the full-text search function on the contents.

Acknowledgement. This work is part of an ongoing project, Multimedia Search System (VG5001), which is supported by the Knowledge Engineering program, under the National Science and Technology Development Agency (NSTDA), the Ministry of Science and Technology, Thailand.

References

1. Cutting, D., Pedersen, J.: Optimization for dynamic inverted index maintenance. In: Proc. of the 13th Int. ACM SIGIR conf., pp. 405–411 (1989)
2. Cutting, D., Pedersen, J., Halvorsen, P.: An object-oriented architecture for text retrieval. Proc. of Intelligent Text and Image Handling (RIAO 91), 285–298 (1991)
3. Haruechaiyasak, C., et al.: Sansarn Look!: A Platform for Developing Thai-Language Information Retrieval Systems. In: Proc. of the 21st International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2006), pp. 85–88 (2006)
4. Haruechaiyasak, C., Sangkeettrakarn, C.: Sansarn Offline: A Search Engine Tool for Managing, Archiving and Retrieving Offline Web Contents. In: Proc. of the Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 1034–1037 (2007)
5. BBC - Learning, `http://www.bbc.co.uk/learning/`
6. Eclipse - An Open Development Platform, `http://www.eclipse.org`
7. Google's Book Search: Library Project - An enhanced card catalog of the world's books, `http://books.google.com/googleprint/library.html`
8. JBoss Enterprise Middleware Suite, `http://www.jboss.com/ products/index`
9. LEXiTRON Thai-English Dictionary, `http://lexitron.nectec. or.th`
10. Overview - Apache Lucene, `http://lucene.apache.org/java/docs/ index.html`
11. MIT OpenCourseWare, `http://ocw.mit.edu`
12. Wikipedia: The Free Encyclopedia, `http://wikipedia.org`
13. Wikipedia Selection for Schools, `http://schools-wikipedia.org`