A UNIFIED FRAMEWORK FOR AUTOMATIC METADATA EXTRACTION FROM ELECTRONIC DOCUMENT

Asanee Kawtrakul and Chaiyakorn Yingsaeree

Department of Computer Engineering, Faculty of Engineering,
Kasetsart University, Bangkok, Thailand
Email: {ak,g4765410@ku.ac.th}

ABSTRACT

Metadata is one of the most important components of modern information system since it helps people to distinguish relevant from non-relevant documents during an information retrieval operation. However, annotating metadata manually is time-consuming, labour-extensive, and expensive. This paper describes a framework for automatic metadata extraction from electronic documents that can be both text documents and images of paper documents to ease metadata creation process. The system consists of three main components: a text conversion module for converting electronic document into standard text file format, a task-oriented parser module for automatically extracting metadata from converted text using pre-defined grammar, and data verification module for identifying and correcting the errors in extracted metadata. The experimental results show that using the proposed framework greatly could reduce the labor work of metadata creation process.

INTRODUCTION

Metadata is, most generally, data that describes other data to enhance their usefulness in content explanation. A typical example of metadata is the traditional library card catalogue; each card gives information about a resource, using elements which are physically included in the resource itself (e.g., author, title, publisher), plus other details which are added by librarians (e.g., subject, classification code, call number). Recent researches [1][2] have been shown that using metadata can significantly improve resources discovery by helping search engines and people to distinguish relevant from non-relevant documents during an information retrieval operation. Although the important of metadata is evident, means for efficient and effective implementation are not. Metadata implementation is complex, due to tremendous growth in digital resource repositories and the development of many different metadata standards.

Addressing this challenge is a growing body of research on automatic metadata generation which can be categorized into two subcategories: metadata harvesting and metadata extraction [3]. Metadata harvesting occurs when metadata is automatically collected from previously defined metadata. The harvesting process relies on the metadata produced by humans or semi-automatic processes supported by software. For example, Web editing software (e.g. Microsoft Publishing) automatically produces metadata at the time a resource is created or updated for ‘format’, ‘date of creation’, and ‘revision date’, without human intervention. The harvesting process usually performs by creating a parser to analyze source metadata using predefined grammar and transform paring results into an appropriated format using mapping rules. The examples of its applications are facilitating interoperability between metadata of different systems and platforms [4], and retrieving metadata from non-cooperating digital libraries [5].

Metadata extraction, on the other hand, occurs when an algorithm automatically extracts metadata from a resource’s content. Among many proposed methods, regular expression, rule-based parser, and machine learning are the most popular of these [6]. In general machine learning are robust and adaptable and, theoretically, can be used on any document set. Generating the labeled training data is very time-consuming and costly. Although regular expression and rule-based system do not require any training and are straight forward to implement, their dependence on the application domain and the need for an expert to set the rules or regular expression causes these method to have limited use.

This paper, then, focused on the problems of automatic metadata generation from electronic documents in an organization, such as Graduate School, in order to provide a simple solution to ease metadata creation process. In this case, adaptability is not a major concern since the documents in an organization usually have a well-defined structure. Thus, it is better to use a rule-based system due to simplicity and cost. Moreover, using rule-based system, one can easily adapt the system to perform both metadata harvesting and extraction functions by changing the rules of the system.

The remainder of this paper is organized as follows: Next section reviews problems in automatic metadata extraction from electronic documents. System architecture is described in Section 3. Section 4-6 describes each of system components. Section 7 reports current development status, and the last section gives the conclusion.

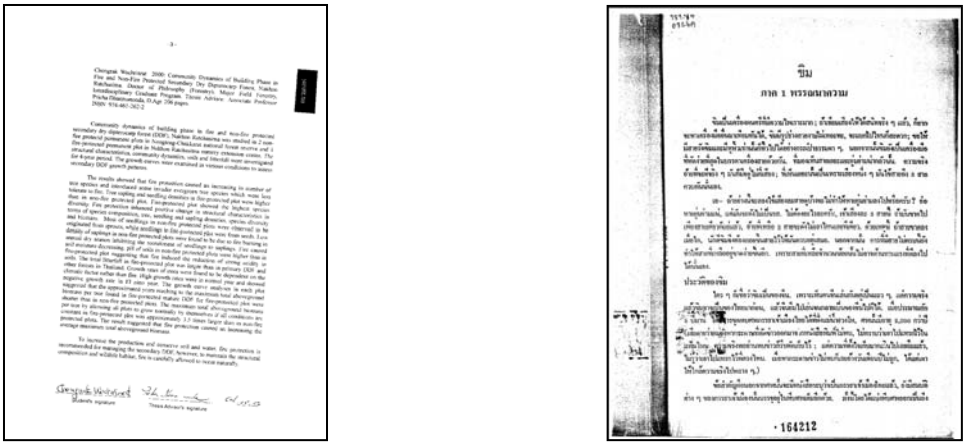
PROBLEMS IN AUTOMATIC METADATA EXTRACTION FROM E-DOCUMENT

Automatically extracting metadata from electronic documents has many problems. Two problems that are worthy to mention are variety of electronic document formats and quality of extracted metadata.

Variety of electronic document formats

Electronic documents in an organization can be stored in a variety of formats. For example, text document can be stored both in standard text file format (.txt) and Microsoft Word file format (.doc). Content of some formats, such as PDF, PS, and Microsoft Word, cannot be accessed directly. To access the content, it is necessary to know the mechanisms to convert the content to an accessible format (i.e. standard text file format).

Recently, in an attempt to move toward a paper-less office, large quantities of printed documents are digitized and stored as images in database. Thus, electronic document is not only text document but also images of paper documents. Working with document images is, then, a lot harder than text documents since the content of the image is not directly available. Future processing, namely Character Recognition, must be applied in order to extract the content from the image. However, the images obtained by scanning paper documents usually contaminated by various type of noises. For example, an improper page placement and feeder malfunction may cause an image to contain skew angle as shown in Figure 1a. Marginal noise, as shown in Figure 1b, may occur when scanning thick documents. Contaminated image are not only unpleasant to view on display device but also challenges to character recognition system whose accuracy is well-known to depend critically on image quality [7]. Hence, these noises should be removed before sending the images to process by character recognition systems.



(a) (b)

Fig.1 The examples of contaminated image.

Quality of extracted metadata

The metadata produced by automatic metadata extraction may contain errors both from original documents and text conversion process, such as Character Recognition. To obtain a high-quality metadata, the extracted metadata should be reviewed carefully. However, manually reviewing all extracted metadata could be time consuming and costly. It will be very helpful if the system can help users to interact only when necessary.

SYSTEM ARCHITECTURE AND DESIGN

As shown in Fig.1, the proposed system can be divided into three main components: a text conversion module for converting electronic document into standard text file format, a task-oriented parser module for automatically extracting metadata from converted text, and data verification module for verifying extracted metadata.

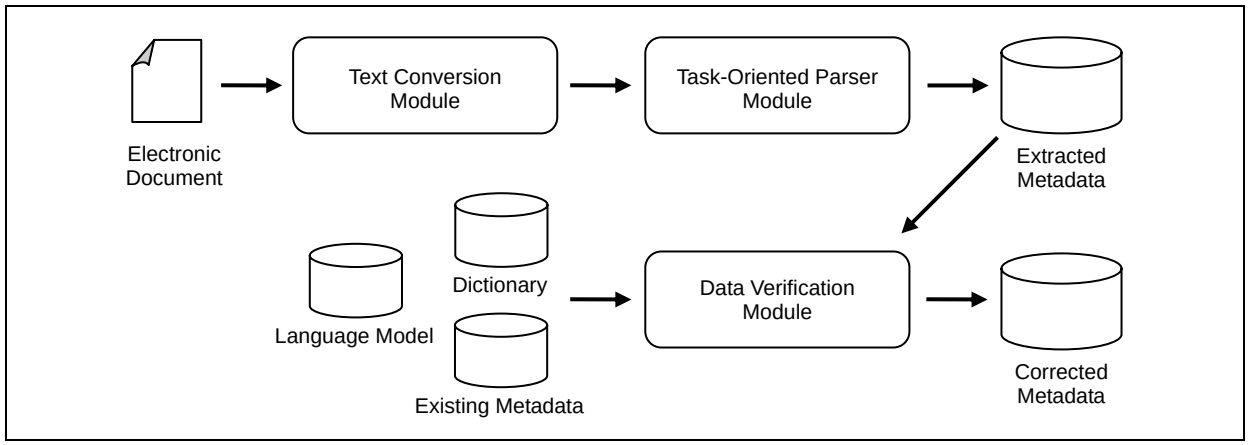


Fig.2 An Architecture of the proposed system

The input to the system is an electronic document which can be in both text format (e.g. PDF, PS, Microsoft Word, HTML), and images of document obtained by scanning paper documents using optical scanner. The Text Conversion Module firstly converted electronic documents into standard text file format. The Task-Oriented Parser Module, then, analyzes the converted text using predefined grammar to automatically extract the metadata. User interaction through the Data Verification Module is required in order to correct parsing errors and the errors from original documents and text conversion module. The following Sections will deeply described the details of each module.

TEXT CONVERSION MODULE

The purpose of the text conversion module is to convert electronic documents into standard text file format by using converter tools. The module is specially designed so that adding new converters can be done easily. Currently, the system supports three types of popular electronic document file formats: PDF, PS, and Microsoft Office documents, plus any accessible text documents (e.g, standard text file, html, and xml) and images of documents. For PDF and PS documents, the system use AFPL GhostScript [8], which is a publicly available software for converting PDF and PS documents into standard text file formant. For Microsoft Document, CATDOC [9], which is opensource converter developed by Vitus Wagner is used as a converter engine.

To handle images of documents, the system firstly removes the noises from the images by using three image-enhancement techniques: document skew correction [10], marginal noise removal [11], and salt-and-pepper noise removal [12]. The examples of images before and after process by each technique are illustrated in Fig.3, Fig.4, and Fig.5. After removing the noises, OCR technique is used to produce text representation of the image. The algorithm used in this system is the one describe in [13]. The recognition system is composed of two major components: document layout analysis for image alignment, locating text blocks and further segmenting into text lines and characters, and character recognizer for classifying each extracted character images. The character recognizer works by using multiple features extraction; three relevant features extracted from a set of training characters are the contour direction of each character, the density of character body and character peripheral information. These sets of features are used as references for classifying unknown input characters. Based on Euclidean space model, the category of the reference vector yielding the minimum distance is assigned to the input character patterns.

The result of this process is, then, sent to Task-oriented Parser Module for analyzing the converted text into to extract relevant metadata from the text.

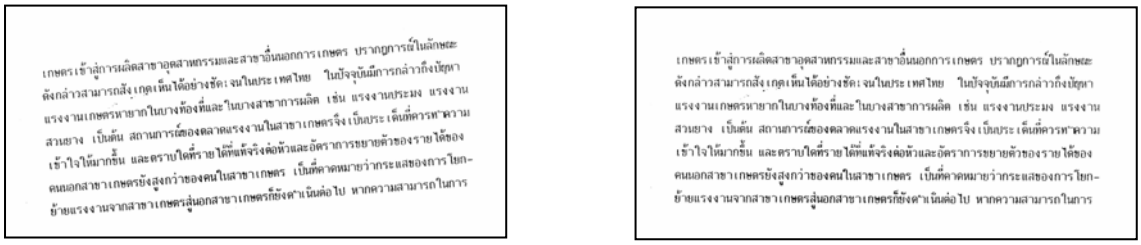


Fig.3 The image before (left) and after (right) performing skew correction

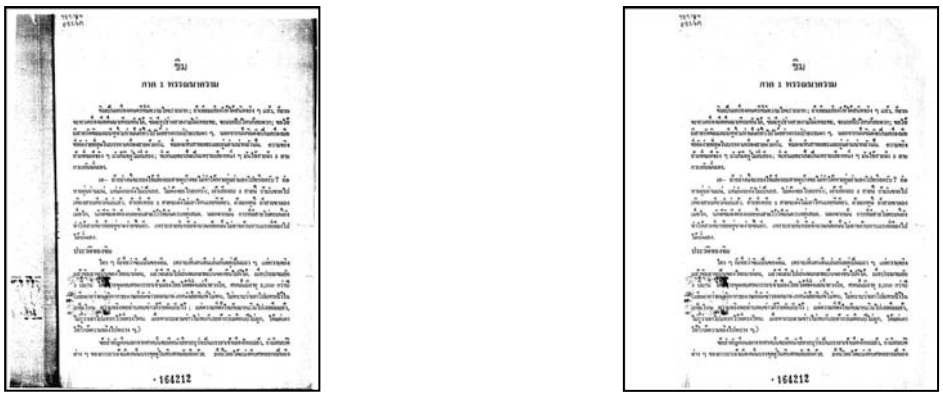


Fig.4 The image before (left) and after (right) performing marginal noise removal

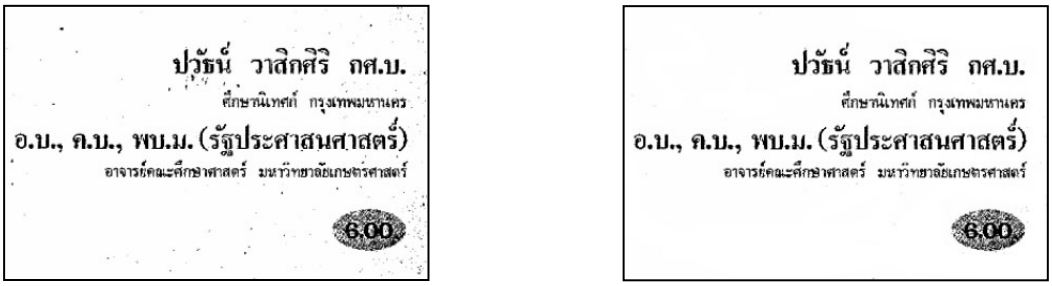


Fig.5 The image before (left) and after (right) performing salt-and-pepper noise removal

TASK-ORIENTED PARSER MODULE

Task-oriented parser module is responsible for automatically extracting the metadata from converted text using predefined grammar. The users must firstly define a specified grammar to analyze the structure of the converted text. Thus, the converted text must have a well-defined structure so that one can create a specified grammar to analyze its structure. In this framework, we use a context-free grammar together with LL(1) parser as an internal engine of the Task-oriented Parser Module.

A context-free grammar (CFG) is a set of rules with specified properties that describes the set of all possible 'sentences' in a specified language. For example, the language of the header of student thesis abstract , as shown in Fig.7, is roughly separated into nine parts which are student's name, graduate year, thesis title, degree name, major name, department name, advisor's name, advisor's degree, and total pages number. After analyzing the header of thesis abstract, one can easily identify the boundary of each part by using special symbols (e.g. ':' and ',') and keyword markers (e.g., "Doctor of", "Major Field", "Thesis Advisor", "pages"). Thus, one can create a grammar to analyze the structure of thesis abstract by using those boundary markers as a part separation point. The example of created grammar is illustrated in Fig.6. To analyze the abstract header using created grammar, we use YAPPS2 (Yet Another Python Parser System) which is a LL(1) parser as a parser engine. After parsing the converted text, the metadata can be directly extracted from parsing results.

After extracting the metadata from converted text, the Data Verification Module will help users identifying and correcting the error in extracted metadata in order to obtain a high quality metadata.

```
<header>        :-    <author-name> <year> : <thesis-title> .
                      <degree-name>, <major-name>, <department-name>.
                      <advisor-name>, <advisor-degree>.
                      <page-number> pages.
<author-name>   :-    <first-name> <last-name>
<first-name>    :-    [A-Za-z]+
<last-name>     :-    [A-Za-z]+
<year>          :-    [0-9]+
.....
```

Fig.6 An example of the grammar for analyzing abstracter header structure

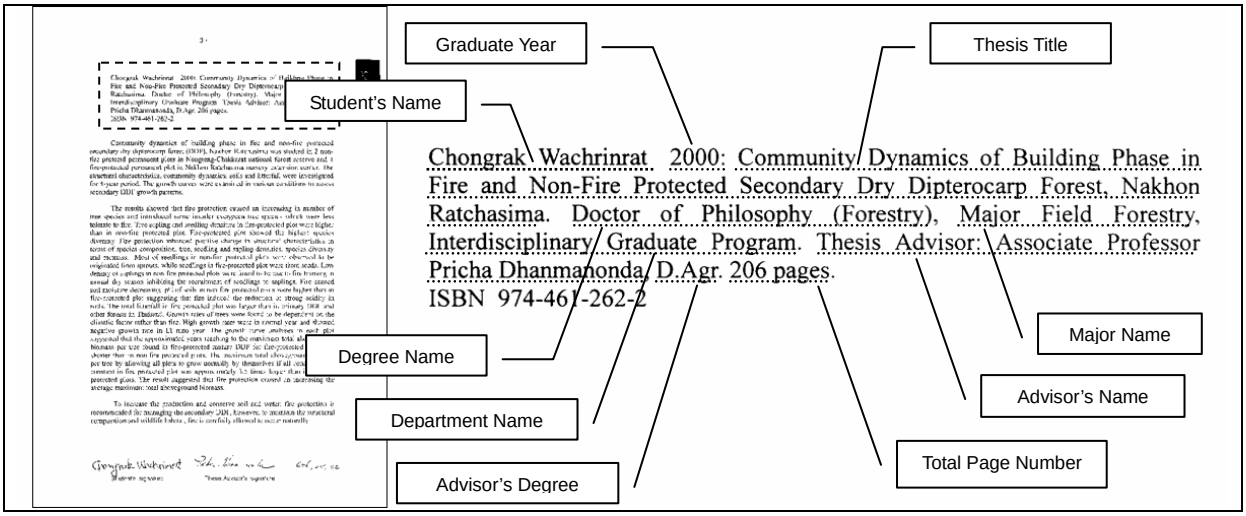


Fig.7 The example of thesis abstract and its structure

METADATA VERIFICATION MODULE

Since the extracted metadata may contain errors both from text conversion module and original documents (e.g, error from mistyping), then, to obtain a high-quality metadata, it is necessary to identify and correct those errors before using the metadata in other applications. The proposed framework integrates four data correction mechanisms in order to help user to correct the errors in different situations.

Error in task-oriented parser

Using a rule-based parser as a core engine, the system may not be able to parse some documents due to incomplete rules or defect in the documents itself. Either creating new rules or fixing the defects is required in order to solve the problem. To ease this process, the system will display error messages, from parser module, that will guide users how to correct those errors. The users then make a decision how to reponse to those errors.

Error in metadata that having controlled vocabulary

Value in some field of extracted metadata can be only a word in controlled vocabulary (e.g., advisor's name, degree name, faculty name, department name). For example, the 'department name' field can be only a name of the department in interested institute. To detect the error in these fields, one can simply compare the extracted metadata with all controlled vocabularies in the dictionary. If the extracted metadata is not matched with any of word in a dictionary, that metadata should contain errors. To correct the errors, the system utilizes the Edit Distance to calculate the distance between extracted metadata and all words in the dictionary. The word that has minimium distance is then used to correct the extracted metadata. If there are more than one minimium, user interaction is required to eliminate the ambiguity.

The Edit Distance was a concept introduced by V.I.Levenshtein in 1965 to measure the distance between two strings [14]. This algorithm is sometimes known as the Levenshtein Distance algorithm. The algorithm measures the difference between two strings by findings the number of edits it takes to change one term to another. An edit is taken as a deletion of a character, substituting a character with another, or an addition of a character. So to change 'Assne' to 'Asanee' would require two edits, a substitution of the character 's' with the character 'a' and an addition of the character 'e'. Therefore, the distance between these two strings is two.

Error in general text metadata

In general metadata field (e.g, title, summary) using edit distance may not be a good idea since any possible string can be appeared in these fields. Thus, more sophisticated method should be employed in order to help the users detecting and correcting the errors. One good choice is to use a spelling correction technique to detect errors and suggest the correction. However, since the errors can be caused by both original documents and text conversion process. Thus, it is better to have two separated module; one for mistyping and one for OCR error correction. This framework utilizes the technique originally developed for Writing Production Assistant System [15]. In current state, this module is currently under development and requires further analysis.

Using metadata extraction to correct the error in existing repository

Hand-made metadata repository usually contains many errors. Correcting those error manually could be time-consuming and labour extensive. To solve this problem, one can uses automatic metadata extraction technique to automated error correction process. After extracting all required metadata from original documents, each extracted metadata entry is matched against the metadata in existing repository to find the corresponding entry. Ideally, each extracted entry should be matched with only one entry of the existing repository. However, since the text conversion module may introduce errors in the extracted text, the query may result in not-match or giving more than one result. In such case, human interaction is required in order to match them manually. To ease this task, the data verification module will display both extracted information and original documents to provide all information user needed to make a matching decision. After matching all extracted entries, the system will compare the extracted metadata with their corresponding entry in existing repository. If they are similar, that entry should be corrected and no correction is needed. On the other hand, if they are different, that entry should contain errors and human interaction is required to correct these errors. Repeatedly, the Data Verification Module will display the extracted metadata, the metadata from existing repository, and their original content to provide all information users needed to correct that entry.

CURRENT STATUS

Most of the modules in the proposed framework are currently developed as separated applications. To build a system for specified task, each module is communicated by reading and writing files controlled by command line interface. Currently, the proposed framework has been employed in two different applications.

Extracting metadata from student's thesis abstract

In this application, we have developing a system to extract metadata from student thesis abstracts as shown in Fig.7. The text conversion module is designed, especially, to convert only the header of the abstract since it contains all metadata needed which are student's name, graduate year, thesis title, degree name, major name, department name, advisor's name, advisor's degree, and total pages number. The interface of developed system is illustrated in Fig. 8.

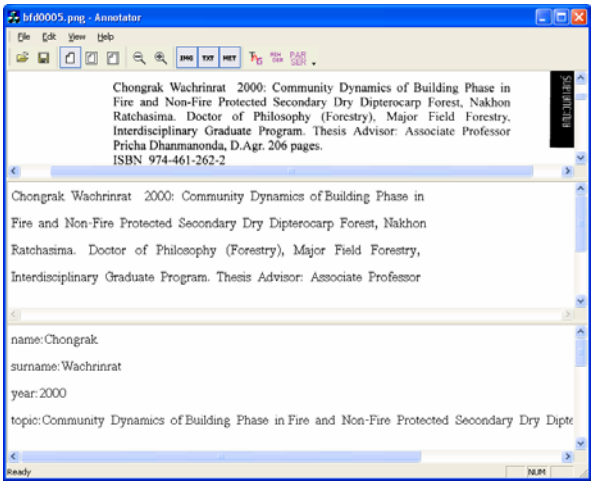


Fig.8 The interface of the system developed for extracting metadata from student's thesis abstract.

To extract metadata, user firstly loads the document into the system. The system will automatically extract metadata from the document. If the system cannot extract the metadata, the system will display error messages that will guide user how to correct those errors. The preliminary results with 3,712 thesis show that using this system greatly reduce the labour work of metadata creation process by correctly extracting metadata 91.41% of the documents.

Extracting plant information from image of Thai plant name dictionary

Thai plant dictionary is a dictionary that contains information of all plants growing in Thailand. The information includes Genus Name, Family Name, Specific epithet, Epithet's author name, Plant habits, Thai name, English pronunciation, and Province that people called that name. As can been seen from Fig.9, this dictionary has a well-defined structure so that one can create a grammar to analyze its structure. The interface of developed system for extracting these information is shown in Fig.10.

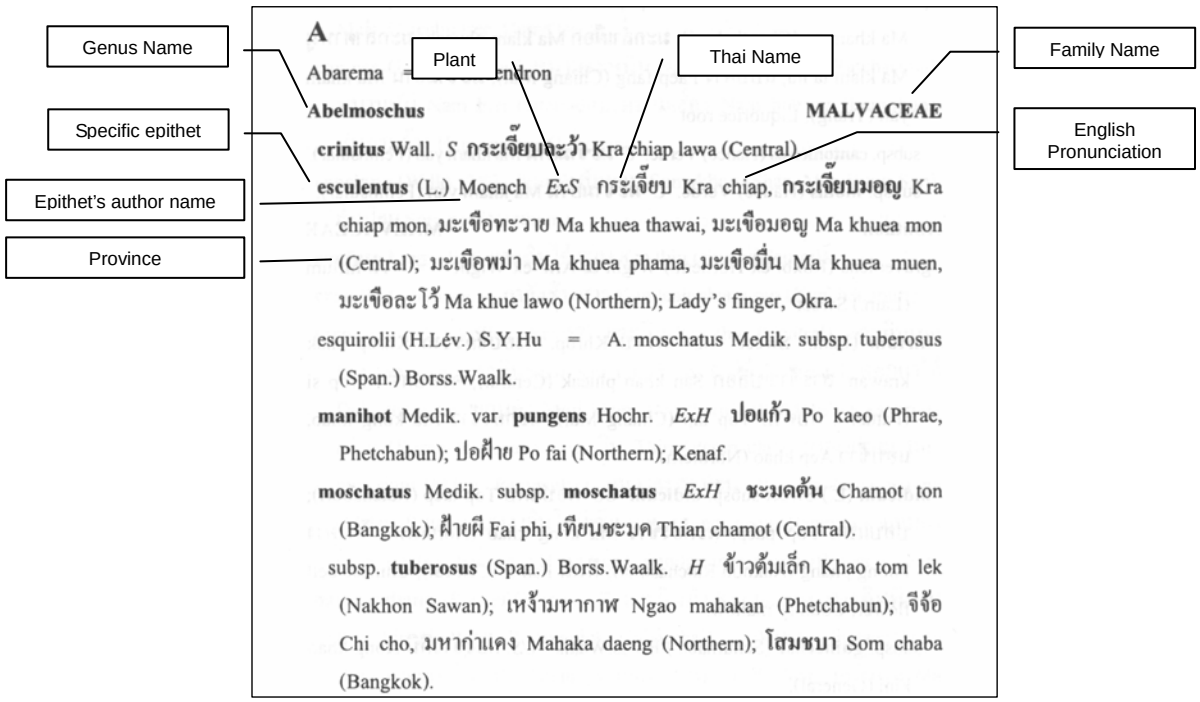


Fig.9 An image of Thai plant dictionary.

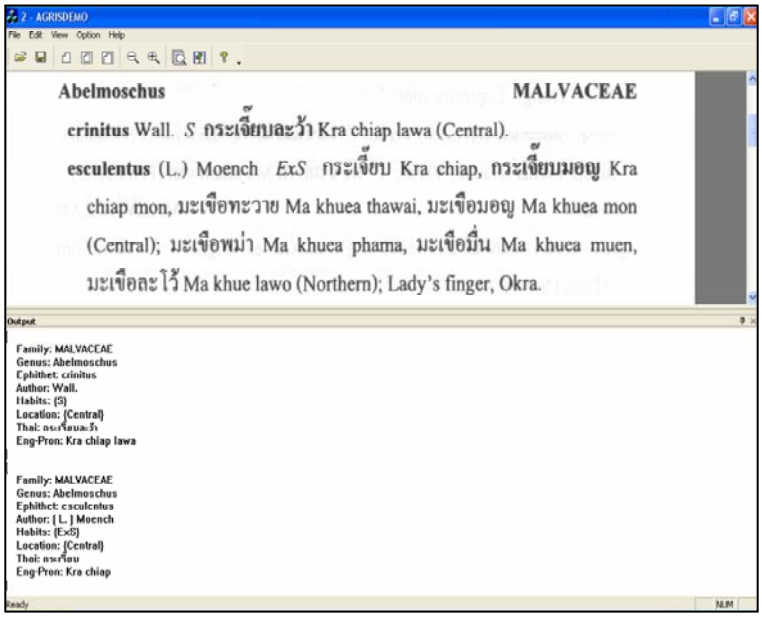


Fig.10 The interface of the system developed for extracting plant information from Thai plant dictionary.

CONCLUSIONS

This paper presented a framework for automatic metadata extraction from electronic documents that can be both text documents and images of paper documents. The system consists of three main components: a text conversion module for converting electronic document into standard text file format, a task-oriented parser module for automatically extracting metadata from converted text using predefined grammar, and data verification module for identifying and correcting the errors in extracted metadata. The experimental results has been shown that using the proposed framework greatly reduce the labour work of metadata creation process.

ACKNOWLEDGEMENTS

This work was supported by Kasetsart University Research and Development Institute, and Graduate School of Kasetsart University, Thailand.

REFERENCES

[1] Shreve, Gregory M., and Zeng, M.L. "Integrating Resource Metadata and Domain Markup in an NSDL Collection". Institute for Applied Linguistics, School of Library & Information Science; Kent State University, 2003.

[2] English, J., Hearst, M., Sinha, R., Swearingen, K., and Yee, K.P. "Flexible Search and Navigation using Faceted Metadata", Jannuary, 2002.

[3] Greenberg, J. "Metadata extraction and harvesting: A comparison of two automatic metadata generation applications". Journal of Internet Cataloging, 6(4), p. 59–82, 2004.

[4] Martines, F., and Morale, F. "Investigation of Metadata Applications at Palermo Astronomical Observatory". Library and Information Services in Astronomy IV, July 2-5, 2002.

[5] Shi, R., Maly, F., and Zubair, M. "Automatic Metadata Discovery from Non-cooperative Digital libraries". In Proceedings of IADIS International Conference on e-Society 2003, Lisbon, Portugal, June, 2003.

[6] Han, H., Giles, C.L., Manavoglu, E., Zha, H., and Zhang, Z. "Edward A. Fox: Automatic Document Metadata Extraction Using Support Vector Machines". JCDL 2003, p. 37-48, 2003.

[7] Rice, S.V., Kanai, J., and Nartker, T. A. "An evaluation of OCR accuracy. In Information Science Research Institute", 1993 Annual Research Report, Unitversity of Nevada, Las Vegas, p. 9-20, 1993.

[8] Lang, R. "AFPL Ghostscript". Available at http://www.cs.wisc.edu/~ghost/

[9] Wagner, V. "CatDoc". Available at http://www.45.free.net/~vitus/ice/catdoc/

[10] Nakano, Y., Shima, H., Fujisawa, J., Higashino, and Fujinawa, M. "An algorithm for the skew normalization of document image". In Proceedings of International Conference on Pattern Recognition, volume II, p. 8-13, 1990.

[11] Peerawit, W., and Kawtrakul, A. "Marginal Noise Removal from Document Images Using Edge Density", In Proceeding of Information and Computer Engineering Workshop, January, 2003

[12] Yingsaeree, W., and Kawtrakul, A. "Quadtree Structure based Multiresolution Salt-and-Pepper Noise Removal", In Proceeding of Information and Computer Engineering Workshop, January, 2003.

[13] Waewsawangwong, P., and Kawtrakul, A. "Multi-Feature Extraction for Printed Thai Character Recognition", The Fourth Symposium on Natural Language Processing, 1998.

[14] Levenshtein, V.I. "Binary codes capable of correcting deletions, insertions and reversals". Doklady Akademii Nauk SSSR, 163(4), p. 845-848, 1965.

[15] Kawtrakul, A., et.al., "A Computational Model for Writing Production Assistant System", Proceeding NLPRS'95 Natural Language Processing Pacific Rim Symposium, December 4-7, 1995.