PDF Extraction Based on Lexical Analysis for Thai Texts

Santipong Thaiprayoon[1], Choochart Haruechaiyasak[2] and Alisa Kongthon[2]

[1]Business Computer Department, Faculty of Business Administration
Ratchaphruek University, Thailand
santipongto@gmail.com
[2] National Electronics and Computer Technology Center
National Science and Technology Development Agency, Thailand
{choochart.har, alisa.kon}@nectec.or.th

Abstract

Today, one of the most widely used digital document file format is the Portable Document Format or PDF. The main advantage of PDF is the ability to create and share documents across platforms with different operating systems and hardware environments. Although, many tools for generating PDF files from text documents exist, however, there is no standard tool for converting PDF files into texts with 100% accuracy. The errors are mainly caused by the misplacement of some characters in the resulting texts. For Thai language, the problem is more intensified due to the complex lexeme structure, i.e., character composition, of Thai words. In this paper, we first surveyed PDF extraction tools which is suitable for Thai language. To further improve the quality of the extracted texts, we propose an approach called PDF-PP (Thai PDF Post Processor), which performs text cleansing based on the lexical analysis. The experiment results using a large corpus showed that the proposed Thai PDF-PP could help improve the accuracy of extracted texts up to 99.78%.

Keywords: PDF Extraction, PDF Generator, Lexical Analysis, Thai Texts

1. Introduction

PDF has become the universal exchange format for digital document. Despite its advantages on viewing and printing capabilities, PDF has some drawbacks on search and extraction abilities. Extracting original text from PDF documents can be a very challenge problem. Most of the open-source PDF extracting tools can handle low-level parsing and manipulation of objects in PDF documents. However, these tools do not fully support recovery of original text in reading order and complex structures. As a result, many studies have been developed for high-quality extraction of text and structure from PDFs in English documents [1, 2, 3, 4]. However, for Thai texts, the fundamental problem for PDF extraction is the inability to return correct characters due to the complex lexeme structure, i.e., character composition, of Thai words. In this paper we propose a new approach for automatic extraction of Thai texts from PDF file. With the improved extracted texts, we can develop high-level applications such as indexing, searching and text mining.

2. Proposed Approach

This section first gives a brief review on the characteristics of Thai written language. We then report a survey which compares several PDF generating and extracting tools for Thai language. In this section, we also propose an approach for performing post processing based on the lexical analysis to improve the quality of the extracted texts.

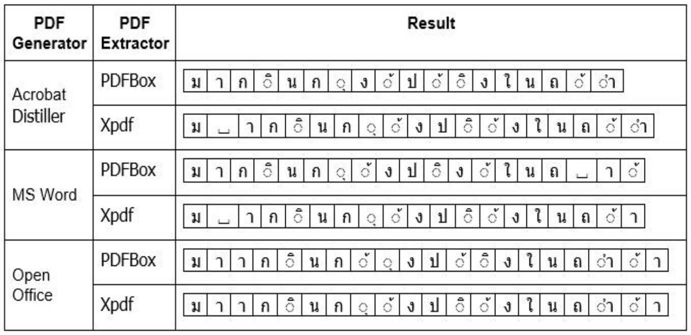
2.2 Thai Language Characteristics

Thai character set consists of 44 consonants, 15 vowel symbols and four tone marks. Figure 1 shows 87 valid characters in the Unicode system. The lexeme structure or the character composition of Thai words is more complex than English. In Thai written texts, consonants are written horizontally from left to right, with vowels placed in four positions: above, below, left and right of the corresponding consonant. Tone marks can be placed in the position above of consonants and some vowels. The complex writing system makes the PDF extraction task more difficult.

2.1 A Survey of PDF Extracting Tools

From our survey, there are currently many PDF extracting tools for converting PDF file into text. In this paper, we compare between two open-source tools, PDFBox and Xpdf, which are widely used among developers.

To compare the performance between PDFBox and Xpdf, we apply three most popular PDF generating tools: Microsoft Word, Adobe Acrobat Distiller and Open Office. Each generator tool has different approach in encoding the input text into the PDF format. Fig. 1 shows a comparison of output texts using different PDF generating and extracting tools. Example text: มากินกุ้งปิ้งในถ้ำ (Come and eat grilled shrimp in a cave)



PDF Generator	PDF Extractor	Result
Acrobat Distiller	PDFBox	ม า ก ◌ิ น ก ◌ุ ง ◌้ ป ◌้ ◌ิ ง ใ น ถ ◌้ ◌ำ
	Xpdf	ม ␣ า ก ◌ิ น ก ◌ุ ◌้ ง ป ◌ิ ◌้ ง ใ น ถ ◌้ ◌ำ
MS Word	PDFBox	ม า ก ◌ิ น ก ◌ุ ◌้ ง ป ◌ิ ง ◌้ ใ น ถ ␣ า ◌้
	Xpdf	ม ␣ า ก ◌ิ น ก ◌ุ ◌้ ง ป ◌ิ ◌้ ง ใ น ถ ◌้ า
Open Office	PDFBox	ม า า ก ◌ิ น ก ◌้ ◌ุ ง ป ◌้ ◌ิ ง ใ น ถ ◌ำ ◌้ า
	Xpdf	ม า า ก ◌ิ น ก ◌ุ ◌้ ง ป ◌ิ ◌้ ง ใ น ถ ◌ำ ◌้ า

Fig. 1. Result comparison from different PDF extracting tools

From Figure 2, it can be observed that using different PDF generating tool to create a PDF document file yields different text outputs even though the same PDF extracting tool is used. We can summarize the error types for each tool as follows.

Type I error	The vowel, Sara Am (◌ำ) is incorrectly converted into Sara Aa (า). For example, "สำคัญ" is incorrectly converted as "สาคัญ".
Type II error	There is an inserted space before the vowel, Sara Aa (า.(For example, "กระเป๋า" is incorrectly converted as "กระเป๋ า".
Type III error	There is a randomly inserted space before any consonant. For example, "ป้องกัน" is incorrectly converted as "ป้อ งกัน".
Type IV error	The vowel, Sara Am (◌ำ) is incorrectly separated into two characters, Nikhahit (◌ํ) and Sara Aa (า).
Type V error	Sara Ae (แ) is incorrectly separated into two Sara E (เ).
Type VI error	There is an inserted Sara Aa (า) right after Sara Am (◌ำ).
Type VII error	The tonal mark is misplaced with the vowel. For example, the tonal mark, Mai Tho (◌้) is misplaced with Sara Uu (◌ู).
Type VIII error	The above and below vowels are shifted into the wrong position. For example, the sentence "ถ่านหินกาาซธรรมชาติก้าลังขาดัแคลนจ้วนเจียนจัะหมดัโลก".

MS Word has three error types: I, II and III. Adobe has four error types: II, III, IV and V. Open Office Document has three error types: VI, VII and VIII. From the results, we select the Xpdf as the extracting tool since it yields better outputs than PDFBox.

2.2 Thai Extracted PDF Post Processor (PDF-PP)

From previous subsection, it was observed that the results from extracting tools are not perfect and contain some errors. To further improve the accuracy, we need some post processing to handle and correct the errors. We propose a post processing approach called PDF-PP which performs text cleansing based on the lexical analysis. Lexical analysis is the process of converting a sequence of characters into a sequence of words or tokens. The key idea is to parse and tokenize the output texts to ensure the correct word boundaries. The overall process is illustrated in Fig. 2.

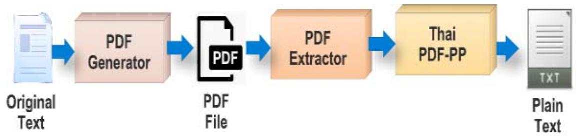


Fig. 2. PDF extraction process with the proposed Thai PDF-PP

We propose solution for each type of error as follows.

Type I error solution	Automatically detect the words with missing Nikhahit (◌ํ) from Sara Am (◌ำ), then try to insert Nikhahit (◌ํ) and parse text. If the process yields a valid word, then the Nikhahit (◌ํ) is correctly inserted.
Type II error solution	Automatically detect Sara Aa (า) and remove the space in front of it. For example, the word "กระเป๋ า" is converted into "กระเป๋า".
Type III error solution	Automatically detect space characters, then try to remove and parse text. If the process yields a valid word, then the space is correctly removed.
Type IV error solution	Automatically detect Nikhahit (◌ํ) followed by Sara Aa (า), then merge them into Sara Am (◌ำ).
Type V error solution	Automatically detect two Sara E (เ), then merge them into Sara Ae (แ).
Type VI error solution	Automatically detect Sara Aa (า) right after Sara Am (◌ำ), then delete Sara Aa (า).
Type VII error solution	Automatically detect the order of the tonal mark and the vowel. If it is an invalid order then switch the order.
Type VIII error solution	No solution for Type VIII error yet.

3. Experiments and Discussion

To evaluate our approach, we perform experiments on eleven documents using the BEST corpus [5]. The corpus contains approximately 500,000 words. We convert each original document to PDF file format using three popular PDF generators: Adobe Acrobat Distiller 11.0, Microsoft Word 2013 and Open Office 3.2. We applied the Xpdf for extracting a PDF file to plain text. We defined the default setting of Xpdf to raw mode and encoding UTF-8. To measure the correctness of extracted content, we find common word sequence between the original text document in the BEST corpus and the extracted content. We then compare the performance of baseline (i.e., extracted content from Xpdf) and our proposed approach using accuracy measurement. Accuracy is computed as the ratio of words converted correctly and the total number of words in the reference. The experimental results are summarized in Table 1.

Table 1. Experimental results on extracted texts

PDF Generator	Accuracy (%)	
	Baseline	Thai PDF-PP
Acrobat Distiller	96.93	99.78
MS Word	96.99	99.64
Open Office	60.79	65.54

We found that, for Acrobat Distiller and Microsoft Word PDF generators, our approach yielded the best extracted texts with the accuracy equal to 99.78% and 99.64%, respectively. However, our approach is not able to achieve 100% accuracy due to problem with word ambiguity. For Open office PDF generator, the accuracy is only equal to 65.54%. One of the problems we encounter is that both tone marks and vowels are totally shifted into the wrong position.

4. Conclusion

In this paper, we propose an approach called Thai PDF-PP (Thai PDF Post Processor) which performs text cleansing based on the lexical analysis. The proposed approach helps increase the accuracy for all PDF generators by approximately 3 -5%. From our text error analysis, the major problem is the misplacement of some characters. For future work, we plan to improve the accuracy of our approach by applying n-gram model to help reduce word ambiguity.

5. References

1. Lovegrove, W.S., Brailsford, D.F.: Document analysis of PDF files: methods, results and implications. Electronic Publishing, Vol. 8 (2 & 3), pp. 207-220 (1995)
2. Hadjar, K., Rigamonti, M., Lalanne, D., Ingold, R.: Xed a new tool for extracting hidden structures from electronic documents. In: DIAL ’04, First Int’l Conference on Document Image Analysis for Libraries, pp. 212–224 (2004)
3. Berg, Ø., Oepen, S., Read, J.: Towards high-quality text stream extraction from PDF: technical background to the ACL 2012 contributed task. In: Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 98-103 (2012)
4. Tiedemann, J.: Improved Text Extraction from PDF Documents for Large-Scale Natural Language Processing Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science Volume 8403, 2014, pp. 102-112 (2014)
5. Kosawat, K., Boriboon, M., Chootrakool, P., Chotimongkol, A., Klaithin, S., Kongyoung, S., Kriengket, K., Phaholphinyo, S., Purodakananda,S., Thanakulwarapas, T., Wutiwiwatchai, C.: BEST 2009: Thai Word Segmentation Software Contest. In: Proc of SNLP’2009, Bangkok, Thailand, October (2009)