Table Detection and Extraction with XML Format from PDF Documents

Pan Nu Wai
University of Computer Studies(Banmaw)
pannuwaicu@gmail.com

Cho Cho Lwin
University of Computer Studies(Banmaw)
chocholwin216@gmail.com

Nwey Zin Moe
University of Computer Studies(Banmaw)
nweyzinmoe@gmail.com

Abstract

Tables are ubiquitous in digital libraries. Tables are simple representations of information that show relationships between concepts. Most publications use tables to present, list, summarize, structure the important data in document and concrete findings of research report. Tables allow the authors to present information in a structured manner and to communicate and summarize key results and main facts. In information retrieval, understanding the structure of the table and automatically extracting the content of the table are very important. In this paper, the system proposes to automatically detect and extracted the contents of tables from PDF document.This system is devised on a novel page box-cutting method. By using this method, the system can provide all the necessary facts of a table for the table searching and can improve the performance of table detection. This paper describes techniques for table metadata extraction with XML format and tests on PDF documents.Finally, for table extraction,the comparison of heuristic methods and page box-cutting method are mentioned.

Keywords: *Metadata extraction, XML, PDF*

1. Introduction

Scientific digital documents allow us to present and share information different areas of knowledge. Automatic table extraction and table searching are challenging problems for several reasons. Tables use to represent results and associations. Tables are ubiquitous in scientific publications, web pages, financial reports, newspapers, magazine articles, etc. Tables present structural data and relational information in a two-dimensional format and in a condensed fashion. Researchers often use tables to concisely display their latest experimental results or statistical data. Tables become an important information source for information retrieval and the demand for searching table is increasing [7].

Automatic table extraction is certain to the success of table search. However, current search engines do not support it. Automatic table extraction is critical because there exists no table markup language that scientists and other users have adopted for representing table information in documents. A system uses automatically extracted table metadata and searches tables in digital libraries.Tables are previously used in many media, such as HTML, PDF, images, etc.This system focuses on PDF documents. These methods serve the purpose of extracting information from any PDF document containing tables and compare to other table extractors.

This system is to extract table metadata automatically. Because of two reasons, this system is focus on PDF documents. First, PDF gains popularity in digital libraries due to the compatibility of output on a variety of devices. Second, PDF documents are overlooked in table extraction field.

In this paper,an automatic table-metadata extraction algorithm is designed and tested on PDF. This paper describes techniques for extracting table metadata automatically and then gives these metadata as XML format. After that, we compare other table extractor with heuristics methods and page box-cutting method for table extraction.

Theheuristic method abstracts cell data and associates them with their metadata. It was learned from this work that accurate tagging of table is important in building the representation used for information retrieval. However, since this method did not support a fine distinction between types of header rows, it tended to draw in extraneous metadata. Study of the retrieval errors shows that the extraneous metadata is a leading cause of failure.

The table extraction technique based on machine learning uses page box-cutting methods to label individual box of document content file. With the *box-cutting* method, *Table Seer* easily exclude more than 93.6% of the document content in the beginning, which impressively increases the efficiency and accuracy of the table detection. As a result, they often perform significantly better than the heuristic method for table extraction. Table detection results are presented in Section 6.

2. Related Work

Tables are a common structuring element in many documents, such as PDF files. To reuse such tables, appropriate methods need to be develop, which capture the structure and the content information. Although table-related research has recently received considerable attention, most of the research focuses on the table extraction.

Researchers in the automatic table extraction field largely focus on analyzing the table structure in a specific document media. Some researchers try to

associate the table extraction with question answering (QA) or information retrieval. For example, Pyreddy and Croft [6] design a character alignment graph (CAG) to extract tables for information retrieval.Hu [2] designs a system that extracts table related information, stores them in databases, and generates a man-machine dialog to access the table data via a spoken language interfaceAkira Amano [5]proposes a representation of table form document based on XML. However, none of them covers table structure and layout information, as well as the table-related information and the document background.. None of above provide a real web search engine to tables. To the best of our knowledge, *TableSeer* is the first table search engine, which supports the automatic table metadata extraction and table search.

3.TableDefinition and Characteristics

A table is an arrangement of data in rows and columns, or possibly in a more complex structure. Tables are widely used in communication, research, dataanalysis and information retrieval.Tables appear in print media, handwritten notes, computer software, architectural ornamentation, and many other places. Further, tables differ significantly in variety, structure, flexibility,representation and use. This definition summarizes the characteristics of a table using three different aspects: content, form, and function [3].

4. The Table Metadata Extractor

The table metadata extractor comprises of three key parts: 1) a text information stripper (TIS), 2) a table box detector with the page box-cutting method, and 3) a table metadata extractor from both the document level and the table level.

4.1. Text Information Stripper (TIS)

Initially, for each PDF document, TISstrips out the text information from the original PDF source file word by word through analyzing the text operators and the related information. This systemreconstructs these words into lines with the aid of their position information and saves the lines into a Document Content Filein the *T*XTformat. For each document page, *the system* analyzes the text information and merges them into different physical component levels (lines, paragraphs, boxes, pages) according to their font and position information.

4.2. Table Box Detection with Page Box-Cutting Method

The table metadata extraction includes two tasks: table detection and metadata extraction. To effectively detect tables, the system designs a novel page box-cuttingmethod as following. The input of the box-cuttingmethod is a Document Content File. The system defines a page boxas a rectangle of adjacently connected lines with a uniform font size in the same document page. Whether two lines merge into a same page boxis decided by two factors: the font size and the position.

This method treats a line L_n in a document content file as the seed line L_{seed} of a box b_θ. Initially, n=1, θ=1, $L_s=L_n$, found=0 and Ln is the only line in b_θ.If the next line L_n+1 in the same Document Content Filesatisfies the following three conditions, we combine L_n+1 into $b_\theta$and set L_n+1 as the new L_{seed}. Otherwise, L_n+1 will be the $L_{seed}$of a new *box* $b_\theta+1$.

1. C1 is defined that font of L_n+1 is equal to font of L_{seed};
2. C2 is defined that $L_n+1$is adjacent to L_{seed};
3. C3 is as L_n+1 is equal enough to L_{seed};

For all the segmented boxes in a document, this systemclassifies them into three categories: small-font boxes B^{SF}, large-font boxes B^{LF}, and regular-font boxes B^{RF}, whose font sizes are smaller than, larger than, or equal to the fontsize of the document bodytext F_b.Based on the observation and statistical study on the proceeding/journal templates, we summarize a set of heuristic rules (see *Table 1)*, which are crucial for mapping boxes to different logical components (*titles*, *authors*, *affiliations*, *abstract*, *references*, etc.) and specific physical components (*tables*, fi*gures*, etc.).

This system detects tables in at most three iterations (see algorithm 1). In each iteration, this system use a keyword matching method and a white space checking method to examine one group of the boxes. This system create a predefined keyword list K^lthat records all the possible starting keywords of the table captions in scientific documents, such as “Table", “TABLE", “table", “Form", “form", “FORM", “Figure", “FIGURE", etc [1].

Table 1: Heuristic Rules for Table Box Detections

Rules	Contents
1	• Document title, author, affiliation, heading ∈ B^{LF}.
2	• Document title, author, affiliation, abstract ∈Page 1, in a fixed order.
3	• Figure caption, table caption, table body, footnote, Reference usually∈ B^{SF}.
4	• Table caption, table body, footnote, Reference ∉ B^{LF}.
5	• ∀ Figure and table have captions.
6	• ∀ Figure caption are beneath the figure; table captions are above the table.
7	• Table captions start with keywords “Table” or “TABLE” while Figure captions

	start with “Figure” or “FIGURE”.

Algorithm 1: Pseudo code of table detection using the page box-cutting method algorithm

```
Begin
    n ← 1; θ1; Ls Ln; found ← 0;
C1 = font (Ln+1) is equal to font (Lseed);
 C2 = Ln+1 is adjacent to Lseed;
 C3 = Ln+1 is equal enough to Lseed;
K^L[ ] = { “Table”, “TABLE”, “table”, “Form”,“form”, “FORM”, “Figure”, “FIGURE”, “figure”};
for Ln ∈ L do
{
if Ln+1 does not satisfy C1, C2 and C3 then
{
        compare bθ.fontsize() withFbodytext;
classify bθ into one of B^Smallfont, B^Largefont, B^Regularfont;
θθ+1; ←
}
Ln+1∈bθ , Lseed Ln+1, Ln  Ln+1;
}
while ( found = = 0 ) do
{
for bθ to bmax∈B(Smallfont, Regularfont, Largefont) do
{
if b[Startword] exists in K^L then
{
if there is tabular structure in b or its neighbor boxes then
getFtable;found 1;break;
}
}
for each b with b.fontsize = Ftable do
{
if b[StartWord] exists in K^L and has a tabular structure then
b is a table;
        }
}
End
```

4.3.Table Metadata Extraction

Some characteristics of our table metadata extraction are: 1) the table metadata should have meaningful names. 2) They should be easily stored for detecting, indexing and searching. 3) They can be combined differently according to the users' purposes. In order to be able to characterize tables occurring in very diverse composite documents, the system designed a rich and flexible representation scheme for table metadata that describe tables in digital documents.

5. Extensible Markup Language (XML)

XML is a relatively new programming languages, but one which is becoming more and more widely used in a vast range of applications. XML is a markup language, used to describe the structure of data, so anywhere that data is input/output, stored or transmitted from one place to another is a potential fit for XML’s capabilities. Perhaps the most well known applications are web related (especially with the latest developments in handheld web access for which the technology is XML-based).

An XML document enables you to store data in the same way a database enables you to store data. However, unlike databases, an XML document stores data in the form of plain text, which can be understood by any type of device, whether it is a mainframe computer, a palmtop, or a cell phone. Thus, XML server as a standard interface required for interchanging data between various web applications.

XML is platform and language independent, which means it doesn’t matter that one computer may be using, for example, Visual Basic on a Microsoft operating system, and the other is a Unix machine with Java code. Really, any time one computer program needs to communicate with another program, XML is a potential fit for the exchange format. The benefits of XML document become even more apparent when people use the some format to do common things because this allows us to interchange information much more easily.

There have already been numerous projects to produce industry standard vocabularies to describe various types of data. Of course, you could write your own XML vocabularies to describe this type of information if you so wished, but if use other, more common, formats, there is a better chance that you will be to produce software which is compatible with other software.

XML structure: Every XML document includes both logical structure and a physical structure. The logical structure is like a template that enlists the elements to be included in a document and in the order in which they have to be included.

The physical structure contains the actual data used in a document. XML documents are also known as self-describing documents. That is, each document contains a set of rules to which its data must conform. Since the

same set of rules can be reused in another document, other authors can easily create the class of document, if necessary [ref].

XML is a relatively new programming languages, but one which is becoming more and more widely used in a vast range of applications. XML is a markup language, used to describe the structure of data, so anywhere that data is input/output, stored or transmitted from one place to another is a potential fit for XML's capabilities. Perhaps the most well known applications are web related (especially with the latest developments in handheld web access for which the technology is XML-based).

An XML document enables you to store data in the same way a database enables you to store data. However, unlike databases, an XML document stores data in the form of plain text, which can be understood by any type of device, whether it is a mainframe computer, a palmtop, or a cell phone. Thus, XML server as a standard interface required for interchanging data between various web applications.

XML is platform and language independent, which means it doesn't matter that one computer may be using, for example, Visual Basic on a Microsoft operating system, and the other is a Unix machine with Java code. Really, any time one computer program needs to communicate with another program, XML is a potential fit for the exchange format. The benefits of XML document become even more apparent when people use the some format to do common things because this allows us to interchange information much more easily.

There have already been numerous projects to produce industry standard vocabularies to describe various types of data. Of course, you could write your own XML vocabularies to describe this type of information if you so wished, but if use other, more common, formats, there is a better chance that you will be to produce software which is compatible with other software.

XML structure: Every XML document includes both logical structure and a physical structure. The logical structure is like a template that enlists the elements to be included in a document and in the order in which they have to be included.

The physical structure contains the actual data used in a document. XML documents are also known as self-describing documents. That is, each document contains a set of rules to which its data must conform. Since the same set of rules can be reused in another document, other authors can easily create the class of document, if necessary [5].

XML shall be compatible with SGML. It describes to write programs that process XML documents. The XML design should be prepared quickly. Also, XML documents should be human-legible and reasonably clear. The design of XML has to happen formal and concise. XML documents shall be easy to create. XML shall be simplifyingusable over the Internet. XML shall support a variety of applications [4].

6.Implementation and ExperimentalResults of Table Detection

The system implementation is uses the digital library which composed of PDF document with containing tables. And then, by making text information stripper, it can be transformed into text files. Then the box classifier break these text files intothree categories: small-font boxes B^{SF}, large-font boxes B^{LF}, and regular-font boxes B^{RF}, whose font sizes are smaller than, larger than, or equal to the font size of the document bodytext F_b. By using page box-cutting method and table metadata extraction algorithm, the table extraction metadata is obtained.).The output is represented as Extensible Mark-up language (XML).The step by step procedure of the system is demonstrated in figures 1 to 5.

We study to evaluate the quality of the table detection and the table metadata extraction. Each user randomly checks 20 documents and all the corresponding table metadata files. We using evaluation metrics are precision and recall. The experiment on table detection is conducted on a document set with 200 randomly selected PDF documents. Given the number of true tables extracted A, the number of true positive tables but overlooked B, and the number of true negative tables that is misidentified as table C. the Precision is$\frac{A}{A+C}$ and the Recallis $\frac{A}{A+B}$. We detect 200 ducuments,397 tables exit (see Table 2).

TableSeer recognizes 371 tables and all of them are real tables, which means the number of truenegative tables is 0. Based on these limited experimental results, the precision value is 100% and the recall value is 93.5%. CAG recognizes 360 tables; the number of true negative tables is 5. CAG's results in had an precision value is 98.6% for and recall value is 90.7% for table detection. TableSeer has good performance on table extraction(see Table 2).

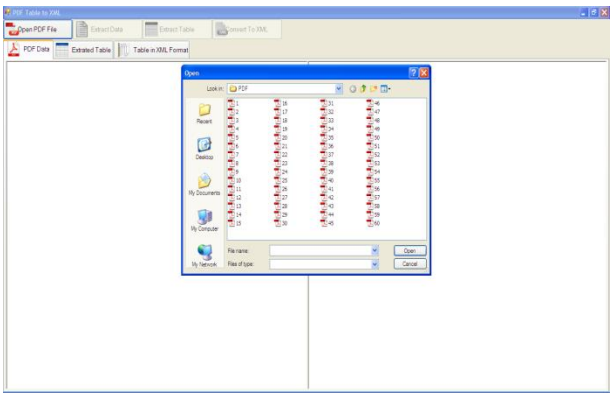

Figure1. Choose PDF file form

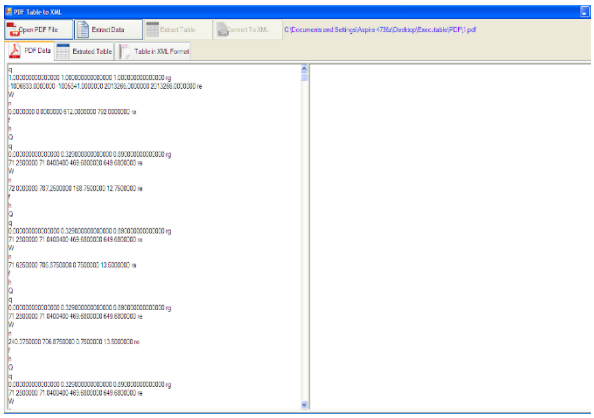

Figure2.Code form PDF file form

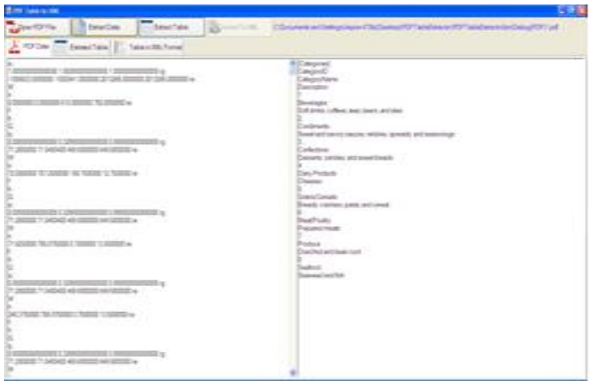

Figure3. Extract data form

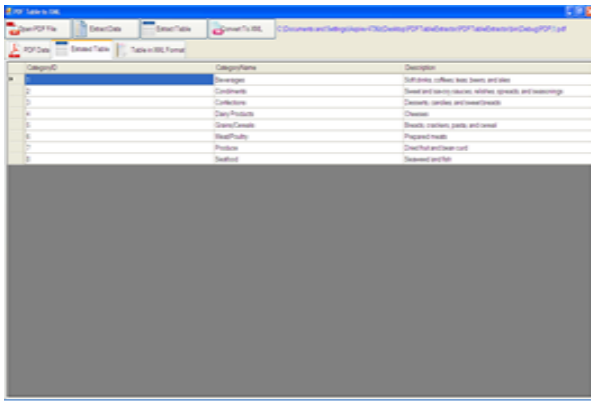

Figure4. Extract table form

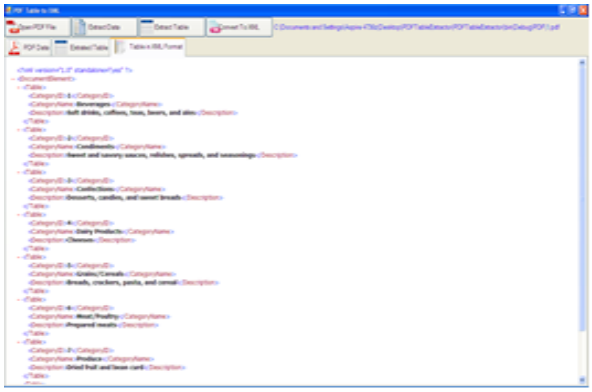

Figure5. Output of XML

Table: 2Experimental Results of Table Detection

Table Detection		
Method	**Precision**	**Recall**
TableSeer	100%	98.6%
Heuristic	93.5%	90.7%

7. Conclusion

The system tested algorithms and randomly selected PDF documents from digital library. Each extracted metadata is evaluated for precision, recall and accuracy. When PDF files are opened, it's related PDF data code will be shown. These data codes are transformed data into text files. And then the extraction stage becomes. After performing the extraction, the table's output will be XML format.By showing extract tables with XML, it can give more easily and clearly understanding advantages. After that,the comparison of heuristics methods and page box-cutting method are mentioned for table extraction.

Acknowledgement

The author would like to express her gratitude to Dr. May phyo Oo, Pro Rector of Computer University (banmaw) and editorial borad of UJCIR, for giving a chance to submit paper. The author is also grateful to the reviewers for their useful comments. Finally, the author thanks her colleagues for their suggestions and cooperation.

References

[1] C. L. Giles, Y. Liu, P. Mitra, K. Bai, "Automatic Extraction of Table Metadata from Digital Documents".

[2] J. Wang and J. Hu, "A machine learning based approach for table detection on the web", In Proceedings of the 11th Int'lConf. on World Wide Web (WWW'02), pages 242{250, Nov2002}.

[3] K. Hadijar, M. Rigamontic, D. Lalanne and Rolf Ingold, "Xed: a new tool for eXtracting hidden structures from Electronic Documents".

[4] Michael J. Casey- Mark A. Austin**, "Sematic Web Methodologies for Spatial Decision Support".

[5] N. A. A. Amano, "Graph grammar based analysis system of complex table form document", In International Conference on Document Analysis and Recognition (ICDAR), pages 916{920,2003}.

[6]P. Pyreddy and W. Croft. Tintin, "A system for retrieval in texttables", In Proceedings of the Second InternationalConference on Digital Libraries, pages 193-200, 1997.

[7] X.Wang, "Tabular Extraction, Editing, and Formatting".