2012 10th IAPR International Workshop on Document Analysis Systems

Dataset, ground-truth and performance metrics for table detection evaluation

Jing Fang, Xin Tao, Zhi Tang*
Institute of Computer Science & Technology,
Peking University, Beijing, China
{fangjing, jolly.tao, tangzhi}@pku.edu.cn

Ruiheng Qiu
Institute of Computer Science & technology, Peking University;
State Key Laboratory of Digital Publishing Technology;
Founder Group Substation, Postdoctoral Workstation of the Zhongguancun Haidian Science Park, Beijing,China
qiuruiheng@gmail.com

Ying Liu
Department of Knowledge Service Engineering
KAIST, Daejeon, Republic of Korea
yingliu@kaist.edu

***Abstract*—Table detection is an important task in the field of document analysis. It has been extensively studied since a couple of decades. Various kinds of document mediums are involved, from scanned images to web pages, from plain texts to PDF files. Numerous algorithms published bring up a challenging issue: how to evaluate algorithms in different context. Currently, most work on table detection conducts experiments on their in-house dataset. Even the few sources of online datasets are targeted at image documents only. Moreover, Precision and recall measurement are usual practice in order to account performance based on human evaluation. In this paper, we provide a dataset that is representative, large and most importantly, publicly available. The compatible format of the ground truth makes evaluation independent of document medium. We also propose a set of new measures, implement them, and open the source code. Finally, three existing table detection algorithms are evaluated to demonstrate the reliability of the dataset and metrics.**

Keywords-dataset, ground-truth, performance metrics, performance evaluation, table detection

I. INTRODUCTION

Table detection---locating and separating tables from other elements within document pages---is an important task in the field of document analysis and understanding. Lots of works have been published on this topic since the past two decades, as summarized in surveys [1, 2]. However, there still exists a major issue to be solved --- “How to evaluate which algorithm is better in different applications?” In other words, performance evaluation is needed in order to compare and select the best-suited methods for a given application. Different algorithms have different shortcomings, and no single algorithm can provide optimal performance considering all evaluation metrics.

Specifically, the issue of table detection evaluation is a challenging problem because of the following sub-problems:

1. Lack of a standard dataset. Most of table detection algorithms are evaluated on their in-house and small datasets. The dataset should be publicly accessible and large-sized, meanwhile, maintain good variety in both document layout and table layout.

2. Lack of ground-truth. Ground-truth containing sufficient and detailed data in multiple aspects is necessary as a benchmark to evaluate experiment results. A user-friendly ground-truthing tool is also meaningful in saving time cost of manual ground-truth labeling.

3. Lack of performance metrics. Proper metrics should be proposed to firstly find out matches between the recognized results and the ground-truth, and then evaluate and compare effectiveness of algorithms.

Most of existing papers analyze experiment results using precision & recall, which are widely used in classification and information retrieval field. However, in the case of table detection where error types are more than false positive and false negative, P&R are not the most appropriate. Table areas may be partially detected, or expanded to non-table area, etc. Fully mining the error types help to understand weakness of one specific algorithm.

In this paper, we provide a general dataset, ground-truth and propose evaluation metrics for table detection algorithms. Both the ground-truthed dataset and the source code of evaluation metrics are publicly available through the website [1]. They will benefit researchers to conduct experiments and carry out evaluations.

The rest of the paper is organized as follows. Section II reviews existing relevant studies in dataset, ground-truth and performance metrics. Section III describes our ground-truthed dataset and performance metrics. Then in Section IV, we describe the experiment on three existing table detection algorithms, and also analyze the corresponding results. Finally, conclusion and future work are given in Section V.

II. RELATED WORK

A. *Dataset and ground-truth*

Most of existing table detection algorithms tested their experiments on small sized dataset. For instance, the dataset used by [3] was composed of 26 Wall Street Journal articles in text format and email messages. A.C. e. Silva [4] gathered 22 PDF financial statements as their dataset.

[1] http://www.founderrd.com/marmot_data.htm

* Zhi Tang is the corresponding author

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.29

The UW-III dataset [5] is most widely used for evaluating document understanding and segmentation tasks. It is also used for table detection because there are 215 marked table zones distributed over 110 pages. Its drawbacks reflect in: *i)* no detailed structure information is provided but only a bounding box of table region is provided; *ii)* dataset is in small size, and only adaptive to scanned images. To overcome its weakness, Y. Wang [6] developed a software tool package, and generated a total of 1125 document image ground-truths with 565 table entities and 10298 cell entities. However, only the software package is downloadable, while the dataset and ground-truth are not publicly available.

B. *Performance metrics*

It is well known that, precision & recall has been widely used to evaluate classification and information retrieval algorithms. Most of table evaluation works published also utilized them as experiment measurements, and judged the false positive and false negative occasions manually. The problem with table detection evaluation is that it is usually not easy to tell whether an obtained table region is correct or wrong, since partially recognition cases cannot be neglected. Therefore, simple manual judgments without quantitative description are subjective and hard to reproduce.

Fortunately, there are already some researches noticed the insufficient of precision & recall, and proposed better performance measurements. As far as it is concerned, J.Hu [3] proposed an edit distance based measurement for table detection. A. Shahab [7] addressed a color-encoding based evaluation method. A.C. e. Silva proposed absolute metrics completeness (C) and Purity (P) in [4]. The former refers to proportion of completely identified terms of the total number of original terms, while the latter means proportion of pure detected elements of all detected elements. It has been used as performance metrics in the table recognition competition organized by ICDAR 2011.

The problem is that, the measures on table detection mentioned above can be applied to compare an overall performance score of various algorithms, but lack the detailed error specification and cannot provide improvement clues. Strictly speaking, they are more like benchmarking rather than evaluation.

Due to the problems stated above, we create and make a large ground-truthed dataset publically accessible, address a set of novel and effective performance measures, and demonstrate the practicality through evaluating three existing table detection methods.

III. Proposed Method

A. *Dataset Collection*

In order to give an objective performance measurement of the algorithms, a large and high-quality dataset is required in all performance evaluation tasks. In this paper, we describe the creation of a large dataset, which could be a first step in standardizing the evaluation of table detection algorithms. Currently, 2000 PDF pages were collected and the corresponding ground truth data were extracted with our semi-automatic ground-truthing tool. 15 people participated in this labeling task. To minimize subjectivity, unified standard for labeling were set, and each ground-truth file is double-checked. The size of dataset is still increasing.

The e-document pages in the dataset show good variety in language types, page layouts, and table styles. First, it is composed of Chinese and English pages at the proportion of about 1:1. The Chinese pages were selected from over 120 e-Books with diverse subject areas provided by Founder Apabi digital library, and no more than 15 pages were selected from each book. The English pages were crawled from web. Over 1500 conference and journal papers were crawled, covering various fields, spanning from the year 1970, to the latest 2011 publications. The Chinese e-Book pages are mostly in one-column layout, while the English pages are printed in with both one-column and two-column layouts. Various kinds of tables are covered in this dataset, from ruled tables to partially and non-ruled tables, from horizontal tables to vertical tables, from inside-column tables to span-column tables, etc.

What's more, we build the dataset with "positive" and "negative" cases at the proportion of around 1:1. Hence, there are 1000 pages containing at least one table, while the other 1000 pages do not contain tables, but have complex layout where some page components may be mistakenly recognized as tables, such as matrices and figures. Thus, not only people can use the "positive" and "negative" cases to test machine-learning methods, but also fake detection errors can be sufficient collected.

In terms of presentation, each single page is composed of three parts: *i)* a labeled ground-truth (to be addressed in next subsection); *ii)* an image format of 600 dpi to represent their original appearance; *iii)* a physical xml description of page unit objects attributes. Basically, the ground-truth contains only necessary information for evaluation, such as bounding box and objects IDs. The physical file contains detailed data, which can also satisfy various extraction requirements. These two are connected by unique ID of each unit object.

In all, the dataset provides an opportunity to test algorithms on real data and do comparison at the first time, without additional cost for data collection.

B. *Ground-truth format*

A semi-automatic gound-truthing tool has been applied to each page of the dataset to generate ground-truths. The Ground-truth metadata is stored in XML format, as shown in Fig.1. XML format is ideal for representing ground-truth since it is the current industry standard. It allows researchers to easily understand and use it for evaluate algorithm or experiment with new metrics.

A set of tags were defined based on the elements in the ground truth data, which consists of two main parts --- *Leafs* and *Composites* elements. The former are the smallest page units corresponding to the parsed text, image and graph content streams together with their associated attributes. The latter refer to distinct logical components labeled by our ground-truthing tool. Each component records its "children", and vice versa. Each table is composed of three parts --- *table caption*, *table footnote* and *table body*. The first two are

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:include schemaLocation="basic_data_type_20110928.xsd"/>
    <xs:simpleType name="Label">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Char"/>
            <xs:enumeration value="Image"/>
            <xs:enumeration value="Path"/>
            <xs:enumeration value="Matrix"/>
            <xs:enumeration value="Formula"/>
            <xs:enumeration value="Figure"/>
            <xs:enumeration value="Textline"/>
            <xs:enumeration value="List"/>
            <xs:enumeration value="TableCaption"/>
            <xs:enumeration value="TableFootnote"/>
            <xs:enumeration value="TableBody"/>
            <xs:enumeration value="Table"/>
            <xs:enumeration value="Paragraph"/>
            <xs:enumeration value="Footnote"/>
            <xs:enumeration value="Body"/>
            <xs:enumeration value="Header"/>
            <xs:enumeration value="Footer"/>
            <xs:enumeration value="Decoration"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="Content" abstract="true" mixed="true">
        <xs:attribute name="Label" type="Label" use="required"/>
        <xs:attribute name="BBox" type="Box" use="required"/>
        <xs:attribute name="LID" type="LayoutID" use="required"/>
        <xs:attribute name="PLID" type="LayoutID" use="required"/>
    </xs:complexType>
        <xs:complexType name="Leaf" mixed="true">
            <xs:complexContent mixed="true">
                <xs:extension base="Content">
                    <xs:attribute name="PID" type="PhysicalID" use="required"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    <xs:complexType name="Composite">
        <xs:complexContent>
            <xs:extension base="Content">
                <xs:attribute name="CLIDs" type="IDArray" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="Leafs">
    </xs:complexType>
    <xs:complexType name="Composites">
    </xs:complexType>
    <xs:complexType name="Contents">
    </xs:complexType>
    <xs:complexType name="Page">
    <xs:element name="Page" type="Page"/>
</xs:schema>
```

Figure 1. Ground-truth schema

optional. Each of them contains *textline* elements. In normal text regions, *textline* means a whole character line not across page column, while in table body regions, *textline* refers to text phases. Text phase is a concept equal to or smaller than table cell, which cannot cover multiple lines. Then, hieratically, textline contains the smallest units— text characters. The hieratical relation is kept based on the parent and children id binding.

C. Performance Metrics

In page segmentation area, the most common categories of errors are over-, under-, and miss segmentation, which appeared in early works of solving segmentation evaluation problem such as [8]. Similarly, table detection could also be treated as one kind of segmentation issue. Tables may be correctly, falsely, partially located, expanded to non-table area, splitted/merged into/by several parts, or entirely missed. Hence, we define six general error types: *fake*, *reduced, amplified, splitted*, *merged*, and *missed* accordingly. Similar ways could also be found in [9], which classify error types into over-segment, under-segment, miss, false positive according to region overlap proportion. Unlike that, we put more emphasis on application-oriented requirements, and adopt both content-based and region-based strategies.

Table detection algorithms reflecting different emphases have been proposed for distinct application requirements. G. Nagy also addressed the importance of "application-oriented benchmarking" in [10]. Accordingly, we define a dictionary of error penalty score and allow users to modify the scores based on their own focuses. Although different scores lead to different results, the evaluation and comparison are solid because they are carried out on the same set of penalty scores. Besides, it helps to check which algorithm is better in each specific application.

Now we take mobile reading application as an example in our evaluation prototype. The error categories are further divided by the following rules:

- Splitting tables vertically are much more severe than horizontally, because the rows would be incomplete and confusing when displaying the separated parts in continuous screen pages.
- Merging tables across document page columns should be given more penalty than merging tables within the same column.
- Fake tables should be given more penalty than missed ones since the former would ruin the reading continuity of non-table regions.
- Mistakenly detected lists, matrixes, and certain figures, with similar layout with tables, are acceptable to some extent, if they are integrated and do not affect reading continuity.
- Amplified tables should also take account what kinds of components are wrongly merged to the table, etc.

More specifically, the six general error types are further divided into 13 subtypes as shown in Table 1, and each of the subtypes will be given a penalty score. On one hand, for *fake* and *amplified* types, we need to know the real logical component to which the falsely detected contents actually belong. In this way, mistakenly detected lists and matrixes can be given lower penalty scores. This is done by content-based strategy -- mapping the obtained results with our ground-truth using unique page objects id. On the other hand, for *splitted* and *merged* types, the split/merge direction can be determined by checking bounding box overlaps. This region-based strategy is novel and effective to obtain positional relations.

Note that, error types should not be the only factor to affect the final results. For instance, fake detected table is generally worse than amplified table area, but if a fake table is small while another amplified table wrongly merged much content of the page, the latter is more severe. Therefore, we define coefficients to those penalty scores. Let N denotes number, PO denotes the unit objects of a page, G denotes the

TABLE I. TABLE DETECTION ERROR TYPES FOR MOBILE READING APPLICATION

General error types	Subtypes for mobile reading
fake	fake_figure; fake_matrix; fake_list; fake_mix

amplified	amplified_tabaccessory;
	amplified_matrices;
	amplified_mix

splitted	splitted_horizontal; splitted_vertical
merged	merged_horizontal; merged _vertical
reduced	reduced
missed	missed

ground-truth table set $\{G_1, G_2, \ldots, G_i, \ldots, G_n\}$, and A denotes the analysis result set $\{A_1, A_2, \ldots, A_j, \ldots A_m\}$, the coefficients are calculated as follows.

- Coefficients for *fake* and *missed* table types:

$$coe_{fak} = \frac{N_{A_j}}{N_{PO}}, \; coe_{mis} = \frac{N_{G_i}}{N_{PO}} \qquad (1)$$

- Coefficients for *reduced* and *amplified* tables:

$$coe_{red} = coe_{amp} = \frac{N_{A_j \cap G_i}}{N_{A_j \cup G_i}} \qquad (2)$$

- Coefficients for *splitted* and *merged* tables, where N_s represents how many parts are one table divided into, and N_m represents how many tables are merged together.

$$coe_{spl} = \frac{1}{N_s}, \; coe_{mer} = \frac{1}{Nm} \qquad (3)$$

The error types are independent to each other. For instance, a splitted table may also be reduced when some contents are indeed lost. Hence, unless the table is fake or missed, all the other error types will be calculated, and the maximum error score will be taken as final score. A threshold is heuristically selected to decide whether a detected table is acceptable. Finally, the number of each error type and acceptable table will be recorded across all the pages. The measurements presented in Table II are used to calculate a revised precision and recall as overall benchmark.

IV. CASE STUDY AND EXPERIMENT RESULT

A. Case study

Based on the performance measures defined in Section III, we evaluated the performance of three table detection algorithms, namely, *Pdf2table* [11], *TableSeer* [12] and our previous work [13]. The first two are open source projects. Therefore, we replaced their original input using data in our ground-truths, and modified the output format to be compatible with the ground-truth schema. In this way, these three algorithms are comparable. Short description of these algorithms is presented as below.

1) Pdf2table [11]

This project developed several heuristics to recognize and decompose tables in PDF files. In term of table detection, the method first merges text elements on the same line to line objects, then classifies single-line and multi-line objects and detects multi-line block objects. Finally, multi-line blocks objects that may belong to the same table are merged.

TABLE II. NOTATIONS FOR EVALUATION METRICS

Notation	Meaning
nr	number of real tables
nm	number of missed tables
na	number of acceptable tables
nfa	number of fake but acceptable tables
nfu	number of fake but unacceptable tables
Precison	na / (nr + nfa + nfu - nm)
Recall	na / (nr + nfa)

1) TableSeer [12]

TableSeer is a table search engine system. It crawls scientific PDF documents, identifies documents with tables, detects table regions, indexes them and enables end-users to search for tables. So far, it is a very complete system for table recognition and search. Specifically, the table detection part is implemented by labeling and merging sparse lines, which is defined as lines containing more than one text phase or shorter than a pre-defined width threshold.

2) Our previous method [13]

We proposed a table detection method via visual separators and geometric content layout information, targeting at PDF documents. The visual separators refer to not only the graphic ruling lines but also the white spaces to handle tables with or without ruling lines. Furthermore, we detect page columns in order to assist table region delimitation in complex layout pages.

B. Experiment results and discussion

After evaluation, the statistics of thirteen subtype error types is shown in Fig.2 and Fig.3, representing results on Chinese and English dataset respectively. Table III shows the overall performance measurements.

From the result figures and table, we observe that the three algorithms bear both advantages and shortcomings: Pdf2table miss least tables, TableSeer detected least fake tables, and our algorithm outperforms both of them in mobile reading application, in terms of most acceptable tables. It is possible that, when the penalty scores are reset, we may get different results.

Although the overall performance is still far from satisfactory, our ground-truth dataset as well as our experimental results provide valuable evaluation on multiple representative table detection algorithms, and help developers or researchers to figure our both advantages and disadvantages of their algorithms in certain application contexts.

I. CONCLUSION AND FUTURE WORK

In this paper, we designed a generally representative and large dataset for table detection evaluation, which is also public accessible. The XML-based ground-truth contains hieratical content data sources of document pages, which make the evaluation independent of document mediums. We also addressed a set of performance metrics, which are mixture of application-oriented penalty scores and content-based quantitative calculation. In addition, we evaluated two open-source table detection projects as well as our previous

TABLE III. EXPERIMENTAL RESULTS AND COMPARISON

	English dataset			Chinese dataset		
Methods	**[11]**	**[12]**	**[13]**	**[11]**	**[12]**	**[13]**
nr	667	667	667	682	682	682
nm	51	208	140	63	249	91
na	261	232	344	223	192	547
nfa	22	1	41	5	0	4
nfu	111	27	23	18	8	19
Precision	0.35	0.48	0.58	0.35	0.44	0.89
Recall	0.38	0.35	0.49	0.34	0.28	0.80

algorithm to demonstrate the reliability of the dataset and the effectiveness of performance measurements. In the future, we would like to enlarge the dataset, evaluate not only results of table structure extraction but also other document components.

ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China (No.2010CB735908).

REFERENCES

[1] R. Zanibbi, D. Blostein, and J. Cordy, “A survey of table recognition: Models, observations, transformations, and inferences,” International Journal on Document Analysis and Recognition, vol. 7, pp. 1-16, March 2004.

[2] AC. e. Silva, A.M. Jorge, and L. Torgo, “Design of an end-to-end method to extract information from tables,” International Journal on Document Analysis and Recognition, vol. 8, pp. 144-171, February 2006.

[3] J. Hu, R. S. Kashi, D. Lopresti, and G. Wilfong. “Evaluating the performance of table processing algorithms”. International Journal on Document Analysis and Recognition, vol. 4, pp.140-153, March 2002.

[4] A.C. e .Silva, "New Metrics for Evaluating Performance in Document Analysis Tasks_Application to the Table Case," International Journal on Document Analysis and Recognition, vol. 1, pp.481-485, 2007.

[5] I.T. Phillips. “User's reference manual for the UW English/Technical Document Image Database III”. Technical report, Seattle University, Washington, 1996.

[6] Y. Wang, I.T.Phillips, R.M. Haralick, "Table Structure Understanding and Its Performance Evaluation", Pattern Recognition, vol. 37, pp. 1479-1497, July 2004.

[7] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel, "An open approach towards the benchmarking of table structure recognition systems", Proc. Document Analysis Systems, 2010, pp.113-120.

[8] F. Shafait, D. Keysers, T. Breuel."Performance Evaluation and Benchmarking of Six-Page Segmentation Algorithms". IEEE Trans. PAMI, vol. 30, pp. 941 – 954, 2008.

[9] F.Shafait, R.Smith, “Table Detection in Heterogeneous Documents”, Proc. Document Analysis Systems, 2010, pp. 65-72.

[10] G Nagy. “Twenty years of Document Image Analysis in PAMI”. IEEE Trans. PAMI, vol. 22, pp. 38-62, 2000.

[11] B. Yildiz, K. Kaiser, and S. Miksch, “pdf2table: A Method to Extract Table Information from PDF Files,” Proc. Indian International Conference on Artificial Intelligence, Pune India, 2005, pp. 1773-1785.

[12] Y. Liu, K. Bai, P. Mitra, and C.L. Giles, “TableSeer: automatic table metadata extraction and searching in digital libraries,” Proc. Joint Conference on Digital Libraries, Canada, 2007. pp. 91-100.

[13] J.Fang, L.Gao, Z.Tang, et al. "A Table Detection Method for Multipage PDF Documents via Visual Seperators and Tabular". Proc. International Conference on Document Analysis and Recognition, Beijing, China, 2011.pp. 779 – 783.

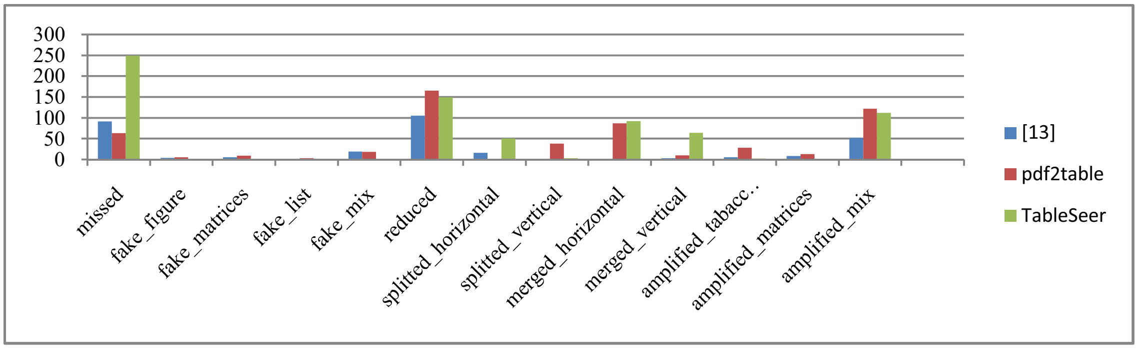

Figure 2. Chinese dataset error statistics

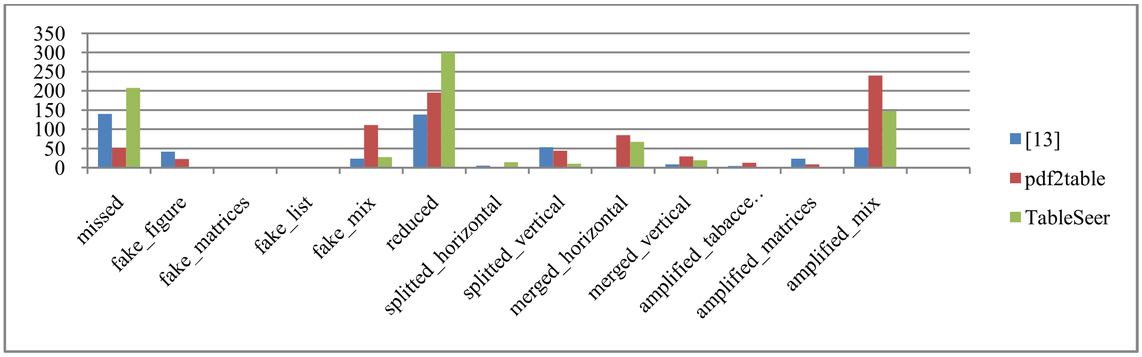

Figure 3. English dataset error statistics