Personal and Ubiquitous Computing (2022) 26:933–939
https://doi.org/10.1007/s00779-020-01485-1

ORIGINAL ARTICLE

TableRobot: an automatic annotation method for heterogeneous tables

Guibin Wu[1] · Junjie Zhou[1] · Yongping Xiong[1] · Chaoyi Zhou[1] · Chong Li[2]

Received: 10 August 2020 / Accepted: 8 October 2020 / Published online: 8 January 2021

Abstract
Using deep learning networks to recognize the table attracts lots of attention. However, due to the lack of high-quality table datasets, the performance of using deep learning networks is limited. Therefore, TableRobot has been proposed, an automatic annotation method for heterogeneous tables. To be more specific, the annotations of table consist of the coordinates of the item block and the mapping relationship between item blocks and table cells. In order to transform the task, we successfully design an algorithm based on the greedy approach to find the optimum solution. To evaluate the performance of TableRobot, we check the annotation data of 3000 tables collected from the LaTex documents in arXiv.com, and the result shows that TableRobot can generate table annotation datasets with the accuracy of 93.2%. Besides, the table annotation data is feed into GraphTSR which is a state-of-the-art table recognition graph neural network, and the F1 value of the network has increased by nearly 10% compared with before.

Keywords Table recognition · Dataset · Deep learning · Annotation · TableRobot

1 Introduction

Table detection and recognition is an important task in document analysis which has been studied by researchers for a long time. With the continuous efforts, researchers had proposed some methods performed well in table detection. However, the researchers still face a lot of problems. For example, early researchers tried to use heuristic methods to recognize tables, but these methods lacked of generalization. Because of the various layouts and formats of the tables, heuristics may perform well on some specific documents, but are invalid for others.

With the rapid development of deep learning, researchers have proposed many deep learning models to solve the task of table recognition, and these methods perform better on heterogeneous tables, which can overcome the barrier for heuristic methods. However, the performance of deep learning models relies heavily on the annotation dataset. In the field of table recognition, because there is not any annotation dataset with heterogeneous tables, using the existing deep learning networks is not feasible in practical application.

In order to solve the problem mentioned above, researchers have published several datasets for table recognition, but they all have shortcomings. The ICDAR 2013 dataset is the official competition dataset with the annotations consist of detailed table structure information. Unfortunately, the ICDAR 2013 dataset contains less than 200 tables, which is not enough to train a feasible deep learning network, while TableBank is a large-scale table dataset, but it lacks item content and bounding boxes of cells.

In fact, there are a large number of heterogeneous tables from various documents on the Internet, but the detailed annotations for these tables are lacked. The annotations of ImageNet were created by crowdsourcing, which consumes a lot of resources, and the annotations of the table that contain hundreds of bounding boxes and detailed structure information of each cell are extremely complicated. Therefore, it will be much more expensive than ImageNet to build a table annotation dataset by crowdsourcing, which has become a major obstacle to the construction of table datasets.

✉ Yongping Xiong
ypxiong@bupt.edu.cn

[1] School of Computer Science, Beijing University of Posts and Telecommunications, 10th Xitucheng Road, Haidian District, Beijing 100876, China

[2] Computer Network Information Center, Chinese Academy of Sciences, No.4, Zhongguancun Nansijie, Haidian District, Beijing 100190, China

To address this research dilemma, we propose TableRobot, an automatic annotation method of heterogeneous tables. TableRobot is suitable for dealing with tables with varying layouts and formats, which helps us generate a high-quality table annotation dataset automatically.

We crawl 1927 LaTex documents from arXiv.com and feed them into TableRobot to generate table annotation dataset, then 3000 tables were selected, and we manually check the accuracy of the annotations. The results show that TableRobot is capable of annotating datasets with the accuracy of 93.2%. Moreover, we verify the feasibility of our dataset in training deep learning network, and result achieves state-of-the-art performance.

2 Related work

2.1 Table detection

In the previous studies, researchers mainly focused on the task of table detection, which is less complicated than table recognition. They firstly used rule-based methods to detect table structures and to locate tables using bounding-boxes. There have been lots of researches, such as the T-Res system presented by Kieninger [1], the PDF2table method proposed by Yildiz [2], and the method combined with OCR released by Tupaj et al. [3], etc.

Subsequently, the data-driven method of deep learning began to appear with rapid development of artificial intelligence. Based on deep learning, researchers have proposed many methods which have great performance in table detection. For example, Schreiber et al. (2017) [4]) proposed a method based on Faster-RCNN network; Li et al. (2019) [5]) presented a method combined with GAN and so on.

2.2 Table recognition

Compared with the well-developed table detection, there is still no satisfactory method to complete the task of table recognition. Initially, researchers used rule-based methods to recognize table structures, which are mainly extracting tables from image-based documents based on handcrafted features. Kieninger et al. (1999) proposed the T-Res system. Oro et al. (2009) presented a bottom-up method to extract tables. These methods may perform well on some documents but always are invalid to generalize to other documents. In fact, tables have various layouts and format. For instance, there are less line tables which are more commonly in documents than full line tables due to the beautiful layout. They are more difficult to be recognized by rule-based method for lacking line feature. Therefore, the rule-based method is very dependent on the structural characteristics of the table itself, so its robustness and generalization ability are poor.

To generalize in different documents, researchers began to focus on deep learning methods based on data-driven. Wang et al., (2004) [6]) developed the first data-driven system to extract tables. Klamplf et al. (2014 [7]) presented an unsupervised learning method combined with some handcraft rules to recognize table structure in PDF documents. With the development of semantic segmentation and object detection, table detection and recognition tasks have become more intelligent. Schreiber et al. (2017) [4]) used the FCN to recognize rows and columns in tables. Chi et al. (2019) [8]) proposed a novel graph neural network for recognizing the table structure, named GraphTSR, which takes cells as input graph nodes, and reconstruct tables via predicting the relationship between the cells. Shah et al. (2019 [9]) presented a method that combined the advantages of convolutional neural network and graph neural network, so that the effect is better than the traditional neural network.

Although researchers are constantly exploring using new deep learning networks to table recognition, the accuracy of table recognition is still much lower than table detection. The performance of deep learning relies heavily on a large, accurate training dataset. At this stage, there is no dataset to meet these requirements. Researchers always build a small dataset for their experimentations, but the quality and completeness are uneven, which will cause the trained network to have a bad generalization and sometimes even not as accurate as the rule-based method. Therefore, the lack of high-qualified dataset has become the major bottleneck of deep learning methods for table recognition.

2.3 Existing datasets

- **ICDAR 2013 Dataset** [10]. The ICDAR 2013 Table Competition dataset contains 156 tables in PDF format which are from the European Union and US Government. Each PDF file has two annotation files which are used for table detection and table recognition, respectively. The table recognition annotation files consist of the structure information and the bounding boxes of each cell. To be more specific, the structure information includes cells' content, start-row, end-row, start-column, and end-column. We can reconstruct a table completely based on the annotations. However, the scale of the ICDAR 2013 dataset is too small to train deep learning model.
- **TableBank Dataset** [11]. The TableBank dataset includes 417 k tables and their position coordinates. It was built with weakly supervision from Word and Latex documents on the Internet. However, it only contains tables in image format; while coordinates information and item content are missed in its structure annotations. Therefore, we cannot reconstruct the table from their annotations, which means it cannot be used for model training.

TableBank shows that it is easy to build a dataset with huge amount of data; however, finding an effective and efficient annotation method for it has many difficulties.

- **SciTSR Dataset** [8]. The SciTSR dataset contains 15,000 tables in PDF format and image format. The table structure label is obtained directly from the LaTex source file. In addition, the SciTSR dataset annotates the adjacency relationships among cells for the training of a graph neural network. One thousand examples were randomly selected from the dataset and then checked manually; however, 62 of them were incorrect, which have a bad influence on the training of deep learning models.

2.4 Existing annotation methods

Researchers have proposed some methods or tools to annotate data efficiently. RectLabel [12] is an image annotation tool to label images for bounding box object detection and segmentation. LabelMe created by the MIT Computer Science and Artificial Intelligence Laboratory is an online annotation tool to help build image databases for computer vision research. It can label both images and videos, and the labeled data can be used for target detection, image classification, and image segmentation tasks. VGG Image Annotator (VIA) is a simple and standalone manual annotation tool for images, audio and video. The VIA software allows human annotators to define and describe spatial regions in images or video frames, and temporal segments in audio or video. However, the structure information of the table is far more complicated than the image category label and bounding box. Therefore, none of the above tools can be used for the construction of table recognition dataset, and it is not realistic to annotate a large table recognition dataset by human annotator.

(Li et al., 2019[10]) proposed a method to annotate tables automatically. They released a large dataset containing 417 k tables. The annotations are in HTML format where "<tdy>" denotes cells with content, while "<tdn>" represents cells without content, and "<tr>" indicates the arrangement of rows and columns. However, the above annotations are too simple for a table, since they lacked cell coordinates and text content, and this annotation structure cannot express the structure of complex tables such as the table with spanning cells. Therefore, this kind of annotation cannot be applied to the training of deep learning networks in practice.

3 TableRobot

TableRobot is proposed as an effective automatic annotation method that is suitable for heterogeneous tables. Firstly, we introduce the steps of acquiring and preprocessing table data. Then, a mathematical model of the table automatic annotating task was defined. Next, we design an algorithm based on greedy approach to complete the task. Finally, the annotation results that contain the table's detailed structure information were shown.

3.1 Data preparation

Our data come from the paper written by LaTex from arXiv.com. We compile the LaTex source code into the corresponding PDF. Then, we detect whether there are tables in a document by analyzing the LaTex source code. After that, we analyze the structure of the table in detail, including the rows and columns information, general content of each cell. Because the content of each cell in LaTex is not exactly identical with the item block displayed in the corresponding PDF. Up to now, we have got the structure information of the table, but we cannot be sure which page the table is on, what coordinates of each cell are, and the contents of each cells are still inaccurate.

Then, we parse the corresponding PDF file. PDF file is a document format that is not affected by the platform, and it contains the coordinate information of each character, so we can get the page and coordinate value of each character by parsing PDF. After the parse, we preprocess each character to generate item blocks which consist of several characters. Through a lot of experiments, it can be ensured that each item block will not have the text content of more than one cell, but one cell corresponding to multiple item blocks is allowed.

Moreover, we determine the serial number of the page where the table is located and design an evaluation function, which combines the cell content and the position relationship among cells to speculate the position for the table.

3.2 Modeling

Next, we map the item block in the PDF page to the cells in the table. As mentioned earlier, the content of each cell we get from parsing LaTex is not exactly the same as the item block displayed in the corresponding PDF, which is a challenge in the mapping task. Therefore, we abstract the task as an optimization problem.

(1) Suppose the number of item blocks in the page is n, and all item blocks on each page form a set $B = \{b_i \mid i = 1, 2, 3, \ldots, n\}$.
(2) Define functions $X(b_i)$, $Y(b_i)$, b_i is any item block in set B. $X(b_i)$ represents the abscissa value of the center point of b_i, and $Y(b_i)$ represents the ordinate value of the center point of b_i. Define the function $T(b_i)$ to represent the text content of b_i.
(3) Arrange the elements in B to generate a sequence $\{a_n\}$, which satisfies $Y(a_i) < Y(a_{i+1})$ or $Y(a_i) = Y(a_{i+1})$ and $X(a_i) < X(a_{i+1})$.
(4) Define functions $SR(c_i)$, $ER(c_i)$, $SC(c_i)$, $EC(c_i)$, which respectively represent the start row, end row, start column, and end column occupied by c_i.

(5) Define the symbol $\prec$, when $c_i \prec c_j$, satisfy $SR(c_j) > ER(c_i)$, or $SR(c_j) = ER(c_i)$ and $EC(c_j) > SC(c_i)$; Define the symbol ~, when $c_i \prec c_j$ does not hold and $c_j \prec c_i$ does not hold, then $c_i \sim c_j$.

(6) Suppose that a table in the page has the number of content cells as m, and all cells in the table form a sequence $C = \{c_1, c_2, c_3 \ldots, c_m\}$ which satisfy $c_i \prec c_j$ or $c_i \sim c_j$ when $i < j$.

(7) Define the cells' item contents set W, $W_i = \{T(w) \mid w \in B, G(w) = i\}$, W_i represents the item block elements in the mapping result of c_i. Define the content set recognized by parsing LaTex $L = \{L_i \mid i = 1, 2, 3, \ldots, m\}$.

(8) Define the matching mapping G: $B \rightarrow C$ satisfies $c_{G(a_i)} \prec c_{G(a_j)}$ or $c_{G(a_i)} \sim c_{G(a_j)}$ when $i < j$, $G(a_i)$ is the cell's serial number that the a_i mapping to, a_i is in the cell $c_{G(a_i)}$. In addition, $B \rightarrow C$ must be surjective.

(9) Define the functional editing distance ρ: L × W→R which represents the editing distance between all item contents mapped in a cell and the cell's content recognized by parsing LaTex.

The objective function is

$$f(G) = \sum_{i=1}^{m} \rho(L_i, W_i) \quad (1)$$

Our objective is to solve the optimal mapping G that minimizes objective function $f(G)$.

3.3 Algorithm

We use an algorithm based on greedy approach to solve the optimal mapping G. First, we use some handcrafted rules to solve part of the mapping, which can help us reduce the search space. The pseudo code is shown in the following part, and the function R aims to verify whether the mapping satisfies the constraints.

Algorithm 1: Reduce the searching space

Input: content set L; item block set B; sequence C; initial mapping G; constraint function $\mathcal{R}$; objective function f
Output: intermediate mapping G_1

```
while(i≤m)
  if(strcmp(L_i, b_k)==0 && strcmp(L_i, b_{l(l≠i)})!=0)
    if(R(b_k → c_i)==True)
      G = G + b_k → c_i
while(True)
    v = f(G)
  while(i≤m)
    if(strcmp(L_i, b_k)==0 and (strcmp(L_i, b_{l(l≠i)})!=0 or
          R(b_{l(l≠i)} → c_i)==False))
      if(R(b_k → c_i)==True)
        G = G + b_k → c_i
  if(f(G)≥v)
    break
G_1 ← G
```

Next, we map the remaining item blocks with cells to find the local optimal solution each time. When searching for local optimum, an item block may correspond to multiple local optimal mappings. We use the method with backtracking to traverse all local optimal chains to find the global optimal solution that minimizes the value of the objective function.

3.4 Annotation format

Annotations are stored in xml format. Figure 1 shows a simple table with its annotations. The root node records the PDF file and page where the table is located, and we add the bounding box of table region to the annotations for table detection. In addition, each cell in the table has annotations, including rows and columns information, coordinates, and contents, which is sufficient to reconstruct the table with above annotations. Therefore, our dataset can be directly used as the training set for table structure recognition in data-driven method.

4 Experiments

4.1 Metrics

4.1.1 Accuracy

We evaluate the accuracy of TableRobot from two perspectives. One is the area of each cell, and the other is the item content.

In order to evaluate the accuracy of area, we propose a modified method based on ICDAR2019 competition on table detection and recognition. We calculate the IoU of bounding boxes in annotations and area in ground truth for each cell. The formula is shown in function 2:

$$\text{IoU} = \frac{area(GCP_i \cap DCP_i)}{area(GCP_i \cup DCP_i)} \quad (2)$$

In this formula, GCP_i represents the area of the ith cell in the ground truth, and DCP_i represents the bounding box of the ith cell in the annotations. The precision, recall, and F1 of table area were calculated in IoU thresholds (such as 0.6, 0.7, 0.8, 0.9). The higher the IoU threshold is, the higher the evaluation standard is.

The accuracy of the content of each cell is calculated based on the accuracy of area, because the area of cells is generated by the mapping of blocks and cells. When the IoU of a cell does not reach the threshold, its content must also be wrong, and when the IoU of a cell with multiple blocks reaches the threshold, its content may still be wrong, because the lack of several blocks may also make the cell area error free. Therefore, we compare the number of characters between ground truth and annotations for each cell to judge the accuracy of cell content.

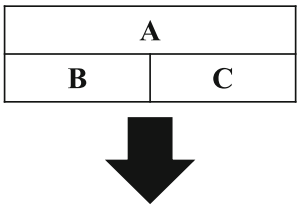

Fig. 1 A simple example of annotation format

```
<?xml version="1.0" encoding="utf-8"?>
<document filename="xxx.pdf" page="x">
    <table id="x" page="x">
        <Coords points="X1,Y1 X1,Y2 X2,Y2 X2,Y1"/>
        <cell id="0" start-row="0" start-col="0" end-row="0" end-col="1">
            <Coords points="x1,y1 x1,y2 x2,y2 x2,y1"/>
            <content>A</content>
        </cell>
        <cell id="1" start-row="1" start-col="0" end-row="1" end-col="0">
            <Coords points="x1,y1 x1,y2 x2,y2 x2,y1"/>
            <content>B</content>
        </cell>
        <cell id="2" start-row="1" start-col="1" end-row="1" end-col="1">
            <Coords points="x1,y1 x1,y2 x2,y2 x2,y1"/>
            <content>C</content>
        </cell>
    </table>
</document>
```

4.1.2 Feasibility

For the dataset created by TableRobot, it is necessary to evaluate its feasibility in the actual training of deep learning network. We also calculate the precision, recall, and F1 in the same way with as in (Göbel et al., 2013 [10]). We use the GraphTSR network as the baseline. GraphTSR was presented in the 15th international conference on document analysis and recognition, which is highly recognized among researchers for its feasibility in practice.

The GraphTSR network uses a graph neural network with attention mechanism. Before training the network, it extracts the coordinates and contents of item block from PDF and generates an undirected graph according to the location relationship among item blocks. Then, the generated graph is input into the network, and the network predicts the relationship (vertical, horizontal or no relationship) between two-item block nodes. Thus, an undirected graph containing relational markers is output from the network. Finally, the table structure is reconstructed from the labeled graph.

The baseline is trained using Intel(R) Xeon(R) Silver 4110 CPU with a mini batch size of 1 image. For parameters, we use the default values in its open source code.

4.2 Data acquisition

We crawl 1927 papers from arXiv.com. The fields of papers include physics, computer science, mathematics, management, business, and so on, which ensures the diversity of the sources of tables. We use TableRobot to automatically discover and annotate tables from these documents and randomly select 3000 tables of them. Then, we visualize the annotations on the image of the document page, which helps us to quickly carry out evaluating.

4.3 Evaluation

4.3.1 Accuracy

For the evaluation of the accuracy of the table area, we set the IoU to 0.9. For table area and item content, the table annotation can be considered right only when the precision and the recall equal to 1 at the same time. Finally, 2796 tables with completely correct annotations were selected. The overall accuracy rate reached 0.932, which shows that TableRobot has great performance in the task of automatic annotation.

Table 1 shows the statistics of our dataset which contains 2796 tables with completely correct annotations, and it is compared with the ICDAR 2013 Dataset. Firstly, thanks to the automatic annotating capability of TableRobot, the scale of our dataset is much larger than the ICDAR 2013 dataset. Secondly, TableRobot can effectively annotate the complex table which contains spanning cells. Finally, the rate of less-line table in our dataset is 82.9% while the ICDAR 2013 dataset is 46.8%. In fact, the less-line tables are more common, but the task of less-line table recognition is still a

Table 1 Statistics of the Our Dataset and the ICDAR Dataset

Type	Our Dataset	The ICDAR 2013 Dataset
Total of tables	2796	156
Table with spanning cells	645	71
Table without spanning cells	2151	85
Less-line table	2318	73
Full-line table	478	83

challenge for researchers. Therefore, TableRobot can generate correct annotations for heterogeneous tables, which is useful for the construction of a diversified dataset.

4.3.2 Feasibility

In order to verify whether our dataset is suitable for training deep learning network, we conduct two groups of experiments. In the existing dataset, the annotations of TableBank are not suitable for table recognition, since the scale of the ICDAR 2013 dataset is too small to be used as training set, but it can be used as test set.

We design four control experiments to inspect the quality and feasibility of our dataset and the SciTSR dataset is in the first group. Each experiment uses all samples of the ICDAR 2013 dataset as test set, and the results are shown in Table 2.

- We randomly sample 1600 tables from our dataset and randomly sample 1600 tables from the SciTSR dataset as two training sets to train the network. Under the same size of training set, the F1 value corresponding to our dataset has increased by 10 percentage points.
- We use the whole samples of each dataset as the training set, and the result shows that the F1 value of our dataset is the highest.

The above experimental results strongly prove that the dataset created by TableRobot is effective for using deep learning method.

Table 2 Evaluation results

Training set	GraphTSR		
	Precision	Recall	F1
SciTSR (1600)	0.928	0.818	0.858
Our (1600)	**0.947**	**0.924**	**0.933**
SciTSR (overall)	0.894	0.860	0.862
Our (overall)	**0.964**	**0.920**	**0.940**

5 Conclusion

In this paper, we propose TableRobot, an automatic annotation method for heterogeneous tables. We use TableRobot to process the LaTex documents crawled from the Internet, and generate a table annotation dataset. The results show that TableRobot is capable of annotating datasets with the accuracy of 93.2%. Moreover, we verify the feasibility of our dataset in training deep learning network, the network trained on our dataset achieves state-of-the-art performance, which proves that TableRobot improve the performance of deep learning networks. We believe that TableRobot will play an important role in the development of table recognition and document analysis.

Acknowledgement This work was supported by the Science and Technology Program of the Headquarters of State Grid Corporation of China under Grant No. SGTYHT/19-JS-215 (Research and development on intelligent consultation experts selection technology based on knowledge graph).

References

1. Kieninger T (1999) The t-recs table recognition and analysis system. Lect Notes Comput Sci 1655:255–269
2. Yildiz B, Kaiser K, Miksch S (2005) pdf2table: a method to extract table information from PDF files. Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Pune, India, December 20-22, 2005. DBLP
3. Tupaj S, Shi Z, Chang CH, Alam H (1996) Extracting tabular information from text files. EECS Department, Tufts University, Medford, USA
4. Schreiber S, Agne S, Wolf I, Dengel A, Ahmed S (2017) DeepDeSRT: Deep learning for detection and structure recognition of tables in document images. Iapr International Conference on Document Analysis & Recognition. IEEE Computer Society
5. Li Y, Gao L, Tang Z, Yan Q, Huang Y (2019) A GAN-based feature generator for table detection. 2019 International Conference on Document Analysis and Recognition (ICDAR)
6. Wang Y (2004) Table structure understanding and its performance evaluation. Pattern Recogn 37(7):1479–1497
7. Klampfl S, Jack K, Kern R (2014) A comparison of two unsupervised table recognition methods from digital scientific articles[J]. D-Lib Magazine 20(11):7
8. Chi Z, Huang H, Xu H, Yu H, Yin W, Mao X (2019) Complicated table structure recognition. arXiv: Information Retrieval

9. Qasim SR, Mahmood H, Shafait F (2019) Rethinking table recognition using graph neural networks. International conference on document analysis and recognition
10. Göbel M, Hassan T, Oro E, Orsi G (2013) ICDAR 2013 Table Competition. 2013 12th International Conference on Document Analysis and Recognition. IEEE
11. Li M, Cui L, Huang S, Wei F, Zhou M, Li Z (2019) Tablebank: table benchmark for image-based table detection and recognition. arXiv: computer vision and pattern recognition
12. Russell BC, Torralba A, Murphy KP, Freeman WT (2005) LabelMe: a database and web-based tool for image annotation. MIT AI Lab Memo AIM-2005-025