Techniques for Detecting and Extracting Tabular Data from PDFs and Scanned Documents: A Survey

Aditya Kekare[1], Abhishek Jachak[2], Atharva Gosavi[3], Prof. P. S. Hanwate[4]

[1,2,3]B.E. Student, Dept. of Computer Engineering, NBN Sinhgad School of Engineering, Ambegoan, Pune- 411041, Maharashtra, India
[4]Professor, Dept. of Computer. Engineering, NBN Sinhgad School of Engineering, Ambegaon, Pune – 411041, Maharashtra, India

---***---

Abstract - *With the advent of smartphones and computers there has been an exponential rise in images, scanned documents and digital documents like PDFs. The amount of variations in the layout of each document makes it hard to find a single solution to extract the data. This paper focuses on the various techniques that have been implemented in detecting and extracting tabular data from the documents. The data in the tables is largely unstructured and thus poses a challenge to extract the data correctly. There are some open-source tools which succeed in recognizing and extracting data from bordered tables like Camelot and Tabula. These tools use natural language processing and machine learning techniques to extract the data. There is a lack of a universal open-source tool which works on textual formats like PDFs as well as scanned documents. Table detection techniques in scanned documents involve Optical Character Recognition and these techniques can vary from the methods used for PDFs. These open-source tools fail to correctly detect the data from borderless tables. There have been other approaches to detecting tables using rule-based methods which require heuristics and meta-data from PDFs. The more recent approaches have been using deep-learning models to detect tables and recognize layouts of the tables. The deep-learning approaches have been successful in detecting tables from both the scanned documents as well as PDFs. We explore a few of these methods and evaluate their performance based on conventional metrics. The paper also suggests some use cases for the problem.*

Key Words: **Natural Language Processing, Machine Learning, Deep Learning, Optical Character Recognition, PDF, Scanned documents**

1. INTRODUCTION

There are large amount of documents used in banking, finance and even education sectors. These documents are largely unstructured and thus developing an automated approach to extract data from them is challenging. The methods for analyzing these documents today are largely manual. This poses significant expenditures for companies and universities in labor costs. There is a need for robust systems which can automate the process of detecting and analyzing the information according to their layouts. Banking and financial domain-based companies need to extract valuable entities from forms and reports. Stock market funds and investment banking companies need to analyze balance sheet/ Profit & Loss statement information from the unstructured reports of the companies. Universities have large amount of data of their students which is trapped in documents like forms, marksheets and identification papers. Such data is usually present in a tabular format, bordered or borderless. A system to automate the process of Named Entity Recognition and value extraction from tabular data will be instrumental in reducing labor costs. We explore the various open-source tools available for table extraction. We use Python language as a filter to choose the tools. We also explore a few deep-learning techniques which have been used successfully for both table-detection and table structure recognition. Our aim is to study and evaluate different tools and approaches implemented till today.

2. RELATED WORK

2.1 Literature Survey on Open-Source tools

There is range of tools available for detecting and extracting tables from documents. Some of them provide APIs in Python to easily use the tools for extraction. They take the PDF as input and detect tables across the PDF. One can also give the page number as the input to detect specific tables inside the PDF. Some of the tools that we have studied are Tabula, Camelot and PyPDF2.

2.1.1 Tabula

Tabula is an open-source tool originally developed for Java. Tabula now also provides an API in Python which works smoothly. The Python API takes the PDF as the input as well as a number of arguments which help to detect different layouts of tables. One of the arguments is the “pages” argument which takes in the page number input and allows to extract specific tables. It also has a “stream” argument which takes a Boolean value. There are two formats to extract the PDFs: “stream” format is for extracting borderless tables while the “lattice” format is used for the extraction of clearly bordered tables.

Tabula also provides an application which has a UI to help the extracting of tables. Application provides the ability to select the area of the page which is then analyzed by Tabula for detecting a tabular structure. Tabula works well on tabular data which has a clearly defined border, but it gives erroneous results on borderless tables. We tried using Tabula on financial annual reports of few companies to extract balance sheets. Tabula failed to extract data from borderless tables but worked well on bordered tables.

2.1.2 Camelot

Camelot is another package available in Python for extracting tables. It has many similarities with Tabula. It also has arguments for page number and stream/lattice format. Camelot also provides a web-interface called Excalibur which is similar to Tabula UI in terms of functionality. The difference between the two is that Camelot works better on images. Camelot gave correct output on the annual report documents which we tested it on.

Both Camelot and Tabula work well on detecting well-defined tables, although their performance on unstructured layouts can be erroneous. Both the packages give a dataframe object as the output which can be converted to csv format or Excel format easily. These libraries are convenient when extracting a few numbers of known tables from the documents, but working on large datasets with unstructured tables can prove to be challenging without a robust logic to detect similar kind of tables.

2.1.3 PyPDF2

PyPDF2 is an open-source package in Python which can extract data from PDFs in textual format. It does not recognize the layout of tables and it just extracts the data in text format. This package is useful for extracting text from many PDFs which can later be used Natural Language Processing applications.

2.2 Literature Survey deep learning approaches

2.2.1 DeepDeSRT

Sebastian et al. [1] propose a deep-learning based model for table detection in images as well as a deep-learning based approach for table structure recognition called DeepDeSRT. Instead of using heuristic-based approaches which have been used in the past, they present a data-driven approach by training a deep-learning model to extract the tables. DeepDeSRT works on both, scanned documents as well as digital documents like PDFs. They used the open-source Marmot dataset for training the model. The Marmot dataset is the largest publicly available dataset of PDFs for table extraction. This dataset was used for training and validating the model. They further evaluate the performance of the model by testing it on the ICDAR 2013 competition dataset. DeepDeSRT was also tested on a private dataset of a European company. This dataset contained tables which were very different from those present in the Marmot dataset. A sample from this dataset was taken for testing. The model provided a high accuracy on table-detection and table structure recognition in spite of the variation in data.

The model uses transfer learning and domain adaptation. A general-purpose object detection model is further train to detect tables from the digital documents. They provide two different approaches for table detection and table structure recognition.

2.2.2 TableNet

Shubham et al. [2] propose an end-to-end deep learning model for table detection as well as table structure recognition called TableNet. Unlike the previous approach where both these problems were treated differently, TableNet provides a single model for the tasks. It leverages the similarities between the tasks of detecting tables and recognizing the structure of rows, columns and headers. The model was trained on publicly available datasets: Marmot and ICDAR 2013. They also demonstrate that the performance of model can be improved by adding semantic features which allows the model to exhibit transfer learning. The model takes a single input and gives two semantically labelled outputs. The model generalizes to other datasets containing different table layouts by minimized parameter tuning and transfer learning. The model predicts the boundaries of rows and columns pixel-wise.

2.3 Literature Survey on dataset preparation

The datasets available for table detection are very few. Marmot dataset is one of the datasets that can be used for table detection, but this dataset also has only a few images.

The datasets available for table structure identification are also very few. The ICDAR 2013 competition dataset is one of the datasets available for table detection as well as table structure identification.[2]

2.3.1 Available datasets

ICDAR 2013 competition dataset: Contains 128 digital documents from EU and US governments.[3]

UNLV Table dataset: Contains 427 scanned documents from magazines, articles, reports, etc. [3]

Marmot dataset: Contains 2000 pages of PDF format from research papers.[3]

3. Analysis of Literature Work:

Components	Author	Methodology
DeepDeSRT	Sebastian et al. [1]	Two separate deep-learning models for table detection and table structure recognition
TableNet	Shubham et al. [2]	A single end-to-end deep-learning model for table detection and table structure recognition
TableBank	Minghao Li et al. [3]	Image-based table detection and recognition deep-learning model
Open-Source tools	-	Input: PDF Output: Dataframe objects A simple API is provided in Python Example: Camelot, Tabula

4. USE CASES

Extracting tables from PDFs and scanned documents has many use cases. One of the use cases is analyzing the marksheets and other personal identification documents submitted by job seekers to the company. The companies have to analyze these documents manually which incurs labor costs. This process can be automated by using the methodologies surveyed in this paper. A deep learning model can be trained on the open-sourced datasets and then applied on customized datasets using domain adaptation and transfer learning. This will help companies cut labor costs, and develop a secure system to ensure a smooth onboarding process for their employees.

Another use case is detecting and analyzing balance sheets and Profit and Loss statements of the companies from their annual reports. The system should be able correctly identify the balance sheets in the reports and ignore other tables present in them. The data should be securely extracted and stored which can be later used to perform statistical models to analyze the performance of the company.

5. CONCLUSION

Thus, we studied the impact of deep-learning and transfer learning on table detection and table structure recognition in this paper. We also studied some open-source tools which are available for the same task.

Based on this literature survey, we propose a use-case for the problem of table detection and table structure recognition. The use-case is that of training a deep learning model and using domain adaptation on it to detect entities in university marksheets. This approach will automate the process of student document analysing in the companies which hire these students. Thus, we propose a model for detecting, extracting tabular diagrams and structures using Natural Language Processing, Optical character Recognition, Image processing, machine learning.

REFERENCES

[1] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, Sheraz Ahmed, "Deepdesrt: deep learning for detection and structure recognition of tables in document images"/ https://www.dfki.de/fileadmin/user_upload/import/9672_PID4966073.pdf

[2] Shubham Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, Lovekesh Vig, "TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images"/ https://www.researchgate.net/publication/337242893_TableNet_Deep_Learning_model_for_end-to-end_Table_detection_and_Tabular_data_extraction_from_Scanned_Document_Images

[3] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, Zhoujun Li "Tablebank: table benchmark for image-based table detection and recognition"/ https://arxiv.org/abs/1903.01949