PubLayNet: largest dataset ever for document layout analysis

Xu Zhong
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
peter.zhong@au1.ibm.com

Jianbin Tang
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
jbtang@au1.ibm.com

Antonio Jimeno Yepes
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
antonio.jimeno@au1.ibm.com

arXiv:1908.07836v1 [cs.CL] 16 Aug 2019

Abstract—**Recognizing the layout of unstructured digital documents is an important step when parsing the documents into structured machine-readable format for downstream applications. Deep neural networks that are developed for computer vision have been proven to be an effective method to analyze layout of document images. However, document layout datasets that are currently publicly available are several magnitudes smaller than established computing vision datasets. Models have to be trained by transfer learning from a base model that is pre-trained on a traditional computer vision dataset. In this paper, we develop the PubLayNet dataset for document layout analysis by automatically matching the XML representations and the content of over 1 million PDF articles that are publicly available on PubMed Central™. The size of the dataset is comparable to established computer vision datasets, containing over 360 thousand document images, where typical document layout elements are annotated. The experiments demonstrate that deep neural networks trained on PubLayNet accurately recognize the layout of scientific articles. The pre-trained models are also a more effective base mode for transfer learning on a different document domain. We release the dataset (https://github.com/ibm-aur-nlp/PubLayNet) to support development and evaluation of more advanced models for document layout analysis.**

Index Terms—**automatic annotation, document layout, deep learning, transfer learning**

I. Introduction

Documents in Portable Document Format (PDF) are ubiquitous with around 2.5 trillion documents available in this format [1]. While these documents are convenient for human consumption, automatic processing of these documents is difficult since understanding the document layout and extracting information using this format is complicated.

Geometric layout analysis techniques based on an image representation of the document combined with optical character recognition (OCR) methods [2]–[4] were firstly used to understand these documents. More recently, image analytics methods based on deep learning are becoming available [5] and are used to train document layout understanding pipelines [1], [6].

Machine learning methods require training data to become successful. In addition, there is a large variety of document templates, which makes this task even more challenging since this increases the number of documents that need to be manually annotated. On the other hand, manual annotation is a slow and expensive process, which is a stepping curve when willing to use these techniques in new domains.

In this work, we propose a method to automatically annotate the document layout of over 1 million PubMed Central™ PDF articles and generate a high-quality document layout dataset called PubLayNet. The dataset contains over 360k page samples and covers typical document layout elements such as text, title, list, figure, and table. Then, we evaluate deep object detection neural networks on the PubLayNet dataset and the performance of fine tuning the networks on existing small manually annotated corpora. We show that the automatically annotated dataset is suitable to train a model to recognize the layout of scientific articles, and the model pre-trained on the dataset can be a more effective base in transfer learning.

II. Related work

Existing datasets for document layout analysis rely on manual annotation. Some of these datasets are used in document processing challenges. Examples of these efforts are available in several ICDAR challenges [7], which cover as well complex layouts [8], [9]. The US NIH National Library of Medicine has provided the Medical Article Records Groundtruth (MARG)[1], which are obtained from scanned article pages.

In addition to document layout, further understanding of the document content has been studied in the evaluation of table detection methods, e.g. [10], [11]. Examples include table detection from document images using heuristics [12], vertical arrangement of text blocks [13] and deep learning methods [14]–[17].

Overall, the datasets are of limited size, just several hundred pages, which is mostly due to the need for manual annotation. In the next section, we describe how multiple versions of the same document from PubMed Central™ are used to automatically generate document layout annotations for over 1 million documents.

III. Automatic annotation of document layout

To overcome the lack of training data, we used a large document collection from PubMed Central™ Open Access

---

[1]https://ceb.nlm.nih.gov/inactive-communications-engineering-branch-projects/medical-article-records-groundtruth-marg/

(a) PDF representation.

(b) XML representation.

(c) `PDFMiner` output on (a).

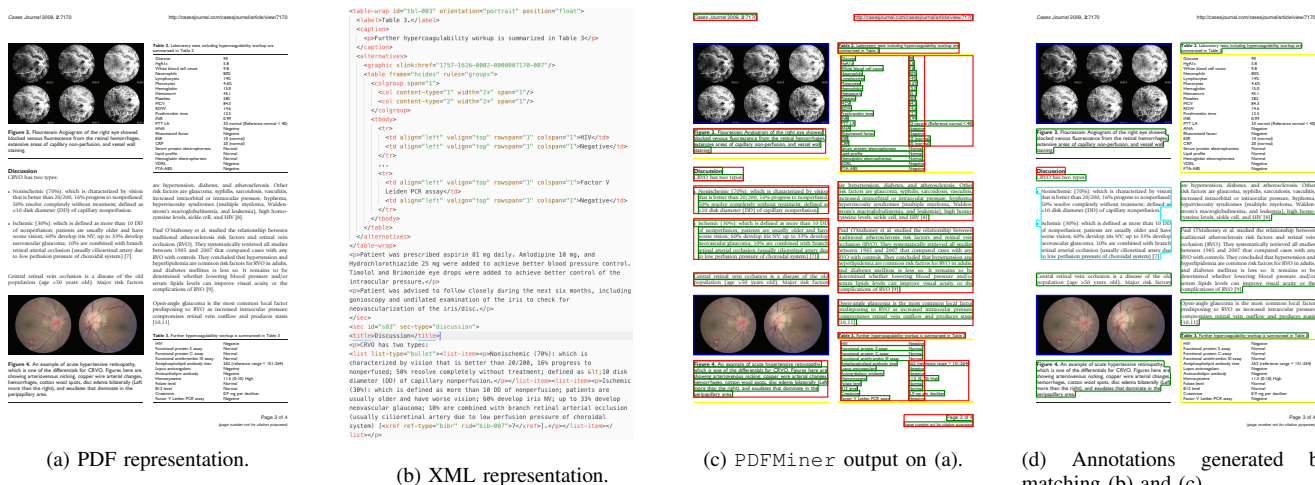(d) Annotations generated by matching (b) and (c).

Fig. 1: Parsing PDF page (a) using `PDFMiner` (c) and matching the layout with the XML representation (b) to generate annotation of page layout (d). The color scheme in (c) is red: textbox; green: textline; blue: image; yellow: geometric shape. The color scheme in (d) is red: title; green: text; blue: figure; yellow: table; cyan: list.

(PMCOA), provided under the Creative Commons license. The articles in PMCOA are provided in both PDF format (Fig. 1a) and in an XML format (Fig. 1b). The XML version of the PMCOA documents is a structured representation of its content and all XML documents follow the schema provided by the NLM for journals[2]. Since the content in the PDF version of the articles and their XML representation contain similar format, we have identified a method to use these two representations of the same article to identify document layout components. In this work, a total of 1,162,856 articles that have a complete XML representation were downloaded from ftp.ncbi.nlm.nih.gov/pub/pmc on 3 October 2018, and automatically annotated with the method described in the following sections.

A. Layout categories

The structured XML representation of the articles in the PMCOA dataset contains many different categories of nodes, which are difficult, even for humans, to distinguish based only on document images. We aggregated the categories of the nodes in the XML into the document layout categories shown in Table I, based on the following considerations:

- The differences between the categories are distinctive and intuitive for a visual model to capture learnable patterns.
- The categories are commonly found in documents in various domains.
- The categories cover most elements that are important for downstream studies, such as text classification, entity recognition, figure/table understanding, etc.

B. Annotation algorithms

Our annotation algorithm matches PDF elements (see Section III-B2) to the XML nodes. Then, the bounding box

TABLE I: Categories of document layout included in PubLayNet.

Document layout category	XML category
Text	author, author affiliation; paper information; copyright information; abstract; paragraph in main text, footnote, and appendix; figure & table caption; table footnote
Title	article title, standalone (sub)section title[a]; standalone figure & table label[b]
List	list[c]
Table	main body of table
Figure	main body of figure[d]

[a] When a section title is inline with the leading text of the section, it is labeled as part of the text, but not as a title.

[b] When a figure/table label is inline with the caption of the figure/table, it is labeled as part of the text, but not as a title.

[c] Nested lists (i.e., child list under an item of parent list) are annotated as a single object. Child lists in nested lists are not annotated.

[d] When sub-figures exist, the whole figure panel is annotated as a single object. Sub-figures are not annotated individually.

and segmentation of the PDF elements are calculated. The XML nodes are used to decide the category label for each bounding box and segmentation. Finally, a quality control metric is defined to control the noise of the annotations at an extremely low level.

1) PMCOA XML pre-processing and parsing: Some of the nodes in the XML tree are not considered for matching, such as *tex-math*, *edition*, *institution-id*, and *disp-formula*. These nodes are removed, as the content of these nodes may interfere with the matching of other nodes. The placement of list, table and figure nodes in the XML schema is not consistent across the articles. We standardized the XML tree by moving list, table, and figure nodes into the floats-group branch. Then, the nodes in the XML tree are split into five groups:

- Sorted: including paper title, abstract, keywords, section

titles, and text in main text. The order of sorted XML nodes matches the reading order in the PDF document.

- Unsorted: including copyright statement, license, authors, affiliations, acknowledgments, and abbreviations. The order of unsorted XML nodes may not match the reading order in the PDF document.
- Figures: including caption label (e.g., 'Fig. 1'), caption text, and figure body
- Tables: including caption label (e.g. Table I), caption text, footnotes, and table body
- Lists: including lists.

2) PMCOA PDF parsing: Fig. 1c illustrates an example of the layout of a PDF page parsed using the `PDFMiner`[3] package, where three layout types are extracted:

- textbox (red): block of text, consisting of textlines (blue). Each textbox has three attributes: the text in the textbox, the bounding box of the textbox, and textlines in the textbox. Each textline has two attributes: the text in the textline and the bounding box of the textline.
- image (green): consisting of images. Each image is associated with a bounding box.
- geometric shape (yellow): consisting of lines, curves, and rectangles. Each geometric shape is associated with a bounding box.

3) String pre-processing: The strings from XML and PDF are Unicode strings. The Unicode standard defines various normalization forms of a Unicode string, based on the definition of canonical equivalence and compatibility equivalence. In Unicode, several characters can be expressed in various ways. To make the matching between XML and PDF more robust, the strings are normalized to the KD normal form[4] (i.e., replacing all compatibility characters with their equivalents).

4) PDF-XML matching algorithms: There are frequent minor discrepancies between the content of PDF parsed by `PDFMiner` and the text of XML nodes. Thus fuzzy string matching is adopted to tolerate minor discrepancies. We use the fuzzysearch[5] package to search for the closest match to a target string in a source string, where string distance is measured by the Levenshtein distance [18]. The maximum distance allowed for a match (d_{max}) is adaptive to the length of the target string (l_{target}) as,

$$d_{max} = \begin{cases} 0.2 * l_{target} & \text{if } l_{target} \leq 20 \\ 0.15 * l_{target} & \text{if } 20 < l_{target} \leq 40 \\ 0.1 * l_{target} & \text{if } l_{target} > 40 \end{cases}$$

As one textbox may cover multiple XML nodes, we sequentially search the textlines of a textbox in the text of a XML node. If the textline of the textbox cannot be found in the text of the XML node, we skip to and search the next textbox. If the end of the XML node is reached, but the textline is not the end of the textbox, the textbox is divided into two textboxes

[3]https://github.com/euske/pdfminer
[4]http://unicode.org/reports/tr15/
[5]https://github.com/taleinat/fuzzysearch

at the current textline. Then the former textbox is appended to the list of matched textboxes of the XML node. When all the content of the XML node is covered by matching textlines, we start searching in the next XML node. This matching procedure is applied to all the text XML nodes, including the 'Sorted', 'Unsorted', and 'Lists' groups; the captions in the 'Tables' and 'Figures' groups; and the footnotes in the 'Table' group.

Depending on the template of specific journals, section/subsection titles may be inline with the first paragraph in the section. A title is treated as inline titles if the last line of the title does not cover a whole textline. Inline section titles are annotated as part of the text, rather than individual instances of titles. The same principle is also applied to the caption labels of figures and tables.

After all the text XML nodes are processed, the margin between annotated text elements in the PDF page is utilized to annotate the body of figures and tables. Fig. 2 illustrates an example of the annotation process for a figure body. First, the bounding box of the main text of the article (green box) is obtained as the smallest bounding box that encloses all the annotated text elements in the article. Then the potential box (blue box) for the figure is calculated as the the largest box that can be fit in the margin between the top of the caption box (brown box) and the annotated text elements above the caption. The last step is to annotate the figure body with the smallest box (red box) that encloses all the textboxes, images, and geometric shapes within the potential box. Table bodies are annotated using the same principle, where it is assumed that table bodies are always below the caption of the tables.

Fig. 2: Annotation process for an example figure. The main text box and the potential box are determined from the annotations of the caption of the figure and surrounding text elements. The final annotation is made as the smallest box that encloses all the textboxes, images, and geometric shapes within the potential box.

5) Generation of instance segmentation: For text, title, and list instances, we automatically generate segmentations from the texelines of the PDF elements, which allows us to train the Mask-RCNN model [5]. As shown in Fig. 3, the top edge of the top textline and the bottom edge of the bottom textline in the PDF elements forms the top and bottom edge of the segmentation, respectively. The right edge of the textlines are scanned from top to bottom to form the right side of the segmentation. ⌐-shape edges are inserted if the right edge of a textline is on the left of the right edge of the textline above it. Otherwise, ⌐-shape edges are inserted. The left side of the segmentation is generated by scanning the left edge of the textlines from bottom to top using the same principle. For figure and table instances, the bounding box is reused as the segmentation, since almost all these instances are rectangular. Fig. 1d illustrates the annotations for the PDF page in Fig. 1a.
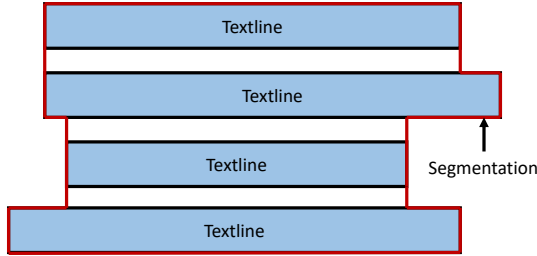
Fig. 3: Example of generating layout segmentation based on textlines. The segmentation is a regular polygon, consisting of only horizontal and vertical edges. The shape of the segmentation is decided by the position of adjacent textlines.

6) Quality control: There are several sources that can lead to discrepancies between the PDF parsing results and the corresponding XML. When discrepancies are over the threshold d_{max}, the annotation algorithm may not be able to identify all elements in a document page. For example, `PDFminer` parses some complex inline formulas completely differently from the XML, which leads to a large Levenshtein distance and failure to match PDF elements with XML nodes. Hence we need a way to evaluate how well a PDF page is annotated and eliminate poorly annotated pages from PubLayNet. The annotation quality of a PDF page is defined as the ratio of the area of textboxes, images, and geometric shapes that are annotated to the area of textboxes, images, and geometric shapes within the main text box of the page. Non-title pages of which the annotation quality is less than 99% are excluded from PubLayNet, which is an extremely high standard to control the noise in PubLayNet at a low level. The format of title pages of different journals varies substantially. Miscellaneous information, such as manuscript history (dates of submission, revision, acceptance), copyright statement, editorial details, etc, is often included in title pages, but formatted differently from the XML representation and therefore missed in the annotations. To include adequate title pages, we set the threshold of annotation quality to 90% for title pages.

C. Data partition

The annotated PDF pages are partitioned into training, development, and testing sets at journal level to maximize the differences between the sets. This allows better evaluation of how well a model generalizes to unseen paper templates.

The journals that contain ≤ 2000 pages, ≥ 320 figures, ≥ 140 tables, and ≥ 20 lists are extracted to generate the development and testing sets. This avoids the development and testing sets from being dominated by a particular journal with a large number of pages, and ensures the development and testing sets have an adequate number instances of figures, tables, and lists.

Half of these journals are randomly drawn to generate the development set. The development set consists of all pages with a list in these journals, as well as 2000 title pages, 3000 pages with a table, 3000 pages with a figure, and 2000 plain pages, which are randomly drawn from these journals. The testing set is generated using the same procedure on the rest half of the journals. To further reduce the noise in the development and testing sets and make more valid evaluation of models, the development and testing sets are curated by human, where profound erroneous pages are removed and moderate erroneous pages are corrected.

The journals that do not satisfy the criteria above are used to generate the training set. To ensure diversity of the training data, from each of the journals, we randomly drawn at most 200 pages with a list, 50 pages with a table, 50 pages with a figure, 50 title pages, and 25 plain pages. The statistics of the training, development, and testing sets are depicted in detail in Table II.

TABLE II: Statistics of training, development, and testing sets in PubLayNet. PubLayNet is one to two orders of magnitude larger than any existing document layout dataset.

	Training	Development	Testing
Pages			
Plain pages	87,608	1,138	1,121
Title pages	46,480	2,059[+]	2,021[+]
Pages with lists	53,793	2,984	3,207
Pages with tables	86,950	3,772[+]	3,950[+]
Pages with figures	96,656	3,734[+]	3,807[+]
Total	340,391	11,858	11,983
Instances			
Text	2,376,702	93,528	95,780
Title	633,359	19,908	20,340
Lists	81,850	4,561	5,156
Tables	103,057	4,905	5,166
Figures	116,692	4,913	5,333
Total	3,311,660	127,815	131,775

[+] These numbers are slightly greater than the number of pages drawn, as pages with lists may be title pages or contain tables or figures.

IV. RESULTS

Three experiments are designed to investigate 1) how well the established object detection models Faster-RCNN (F-RCNN) [19] and Mask-RCNN (M-RCNN) [5] can

recognize document layout of PubLayNet; 2) if the F-RCNN and M-RCNN models pre-trained on PubLayNet can be fine-tuned to tackle the ICDAR 2013 Table Recognition Competition[6]; 3) if the F-RCNN and M-RCNN models pre-trained on PubLayNet are better initializations than those pre-trained on the ImageNet and COCO datasets for analyzing documents in a different domain.

A. Document layout recognition using deep learning

We trained a F-RCNN model and a M-RCNN model on PubLayNet using the Detectron implementation [20]. PDF pages are converted to images using the `pdf2image` package[7]. Each model was trained for 180k iterations with a base learning rate of 0.01. The learning rate was reduced by a factor of 10 at the 120k iteration and the 160k iteration. The models were trained on 8 GPUs with one image per GPU, which yields an effective mini-batch size of 8. Both models use the ResNeXt-101-64x4d model as the backbone, which was initialized with the model pre-trained on ImageNet.

The performance of the F-RCNN and the M-RCNN models on our development and testing sets are depicted in Table III. The evaluation metric is the mean average precision (MAP) @ intersection over union (IOU) [0.50:0.95] of bounding boxes, which is used in the COCO competition[8]. Both models can generate accurate (MAP > 0.9) document layout, where M-RCNN shows a small advantage over F-RCNN. The models are more accurate at detecting tables and figures than texts, titles, and lists. We think this is attributed to more regular shapes, more distinctive differences from other categories, and lower rate of erroneous annotations in the training set. The models perform worst on titles, as titles are usually much smaller than other categories and more difficult to detect.

TABLE III: MAP @ IOU [0.50:0.95] of the F-RCNN and the M-RCNN models on our development and testing sets. M-RCNN shows a small advantage over F-RCNN.

Category	Dev		Test	
	F-RCNN	M-RCNN	F-RCNN	M-RCNN
Text	0.910	0.916	0.913	0.917
Title	0.826	0.840	0.812	0.828
List	0.883	0.886	0.885	0.887
Table	0.954	0.960	0.943	0.947
Figure	0.937	0.949	0.945	0.955
Macro average	0.902	0.910	0.900	0.907

Fig. 4 shows some representative examples of the document layout analysis results of testing pages using the M-RCNN model. As implied by the high MAP, the model is able to generate accurate layout. Fig. 5 illustrates some of the rare errors made by the M-RCNN model. We think some of the errors are attributed to the noise in PubLayNet. We will continue improving the quality of PubLayNet.

---

[6]Other ICDAR competitions require annotations of document layout categories not available in PubLayNet and the size of the development sets is too small for effective training.

[7]https://github.com/Belval/pdf2image

[8]http://cocodataset.org/#detection-eval

B. Table detection

The ICDAR 2013 Table Competition [21] is one of the most prestigious competitions on table detection in PDF documents from European government sources. We created a table detection dataset by extracting from our training set the PDF pages that contain one or more tables, and remove non-table instances from the annotations. We trained a F-RCNN model and a M-RCNN model on this table detection dataset under the same configuration described in Section IV-A. Then the models are fine-tuned with the 170 example PDF pages provided by the competition. For fine-tuning, we used a base learning rate 0.001, which was decreased by 10 at the 100th iteration out of 200 total iterations. The minimum confidence score for a detection is decided by a 5-fold cross-validation on the 170 training pages. The fine-tuned model was evaluated on the formal competition dataset (238 pages) using the official evaluation toolkit [9]. Table IV compares the performance of the fine-tuned models and published approaches. The fine tuned F-RCNN model achieves the state-of-the-art performance reported in [6], where the F-RCNN model was fine tuned with 1600 samples from a pre-trained object detection model. By fine tuning from a model pre-trained on document samples, we can obtain the same level of performance with much smaller training data (170 samples).

TABLE IV: Fine tuning pre-trained F-RCNN and M-RCNN models for the ICDAR 2013 Table Recognition Competition. Based on models pre-trained on table pages in PubLayNet, we obtained the state-of-the-art performance with only 170 training pages.

Input	Method	Precision	Recall	F1-score
Image	**F-RCNN**	0.972	0.964	0.968
	M-RCNN	0.940	0.955	0.947
	Schreiber et al. 2017 [6]	0.974	0.962	0.968
	Tran et al. 2015 [13]	0.952	0.964	0.958
PDF	Hao et al. 2016 [15]	0.972	0.922	0.946
	Silva 2010 [22]	0.929	0.983	0.955
	Nurminen 2013 [21]	0.921	0.908	0.914
	Yildiz 2005 [12]	0.640	0.853	0.731

C. Fine tuning for a different domain

In USA, there is a large number of private health insurance providers. Employees are provided with Summary Plan Description (SPD) documents typically in PDF, which describe the benefits provided by the private health insurers. There is a large variety in the layout of SPD documents provided by different companies. The layout of these documents is also distinctively different from scientific publications.

We manually annotated the texts, tables, and lists in 20 representative SPD documents that cover a large number of possible layouts. This domain specific dataset contains 2,131 pages, 9,379, 2,500, and 820 instances of text, tables, and lists, respectively. A 5-fold cross-document-validation[10] was

---

[9]https://github.com/tamirhassan/dataset-tools

[10]For each fold, the model is trained on 16 documents and tested on 4 documents
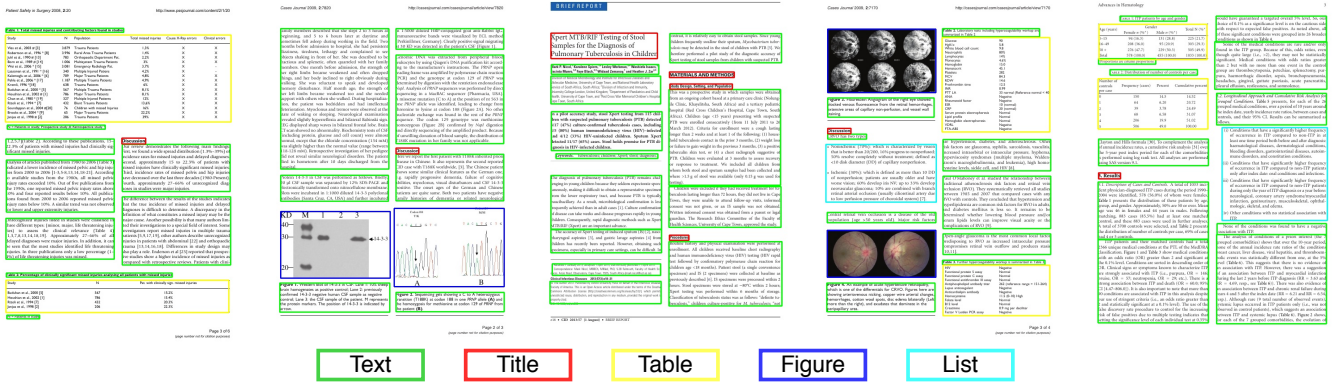
Fig. 4: Representative examples of the document layout analysis results using the M-RCNN model. As implied by the high MAP, the model is able to generate accurate layout.
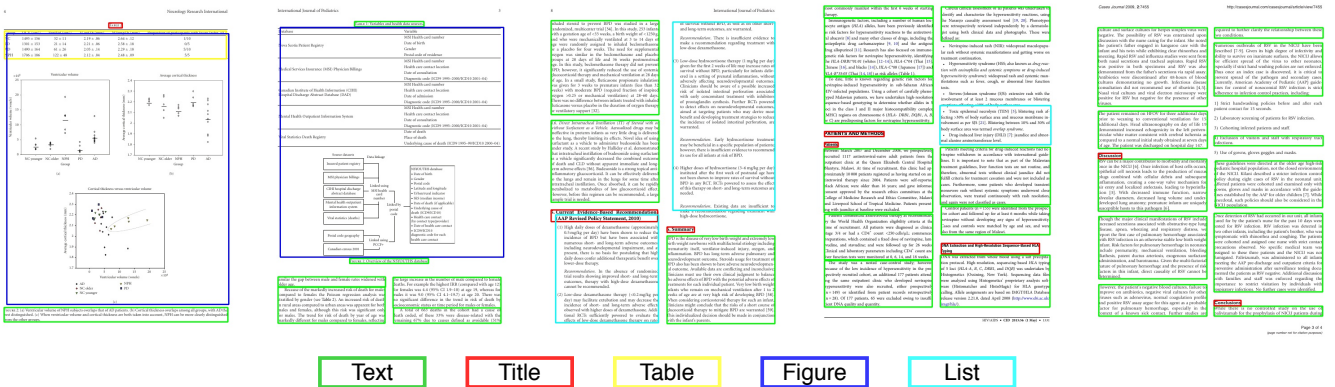

Fig. 5: Erroneous document layout predictions made by the M-RCNN model.

taken to compare different pre-trained F-RCNN and M-RCNN models for fine tuning.

We evaluated three fine tuning approaches: 1) initializing the backbone with pre-trained ImageNet model, 2) initializing the whole model with pre-trained COCO model, and 3) initializing the whole model with pre-trained PubLayNet model. We also tested the zero-shot performance of the pre-trained PubLayNet model. The comparison of the performance of the approaches is illustrated in Table V. The performance of the zero-shot PubLayNet model is considerably worse than the fine-tuned models, which demonstrates the distinct difference between the layout of SPD documents and PubMed Central™ articles. Fine tuning the pre-trained PubLayNet model can substantially outperform other fine-tuned models. This demonstrates the advantage of using PubLayNet for document layout analysis. The only exception is that fine tuning pre-trained COCO F-RCNN model detects tables more accurately than fine tuning pre-trained PubLayNet F-RCNN model. In addition, the improvement on table detection by fine tuning pre-trained PubLayNet MRCNN model is relatively low to that on text and list detection. We think this is because the difference of table styles between SPD and the PubMed Central™ articles and more substantial than that of text and list styles, and therefore

less knowledge can be transferred to the fine tuned model.

V. DISCUSSION

PMCOA provides a large set of documents available at the same time in PDF and XML format. The methodology proposed in this work to generate a large dataset of article pages automatically annotated with the location of document layout components using the PMCOA documents. The quality assurance has shown that the automatically generated annotations are of high quality. In addition, existing state-of-the-art object detection algorithms reproduce successfully the annotations from the automatically annotated set. The title category seems to be the weakest one. The identification of titles is challenging due to the different ways in which titles are present in the documents. On the other hand, titles are identified as text in ICDAR competitions and titles could be merged with the text category in this set up.

The documents in PubLayNet are all scientific literature, which is domain specific and limits the heterogeneity of layout. We took several methods in developing and partition PubLayNet to utilize as much variation in PMCOA as possible and prevent PubLayNet from being dominated by a certain journal. With over 6,500 journals included in PMCOA, our experiment shows that the training set is heterogeneous enough

TABLE V: Comparison of the performance of fine tuning different pre-trained F-RCNN and M-RCNN models on the SPD documents. The performance scores are MAP @ IOU [0.50:0.95] evaluated via a 5-fold cross-document-validation on 20 SPD documents. The results demonstrate the advantage of PubLayNet over general image datasets in domain adaptation for document layout analysis.

Category	Method	Initialization	F-RCNN	M-RCNN
Text	zero-shot	PubLayNet	0.482	0.468
	fine tuning	PubLayNet	**0.701**	**0.708**
	fine tuning	COCO	0.651	0.661
	fine tuning	ImageNet	0.622	0.629
List	zero-shot	PubLayNet	0.508	0.510
	fine tuning	PubLayNet	**0.681**	**0.684**
	fine tuning	COCO	0.622	0.611
	fine tuning	ImageNet	0.603	0.603
Table	zero-shot	PubLayNet	0.422	0.419
	fine tuning	PubLayNet	0.541	**0.596**
	fine tuning	COCO	**0.560**	0.588
	fine tuning	ImageNet	0.528	0.573
Macro average	zero-shot	PubLayNet	0.470	0.465
	fine tuning	PubLayNet	**0.641**	**0.663**
	fine tuning	COCO	0.611	0.620
	fine tuning	ImageNet	0.584	0.602

to train deep learning models that can accurately recognize the layout of unseen journals. For documents in a distant domain, e.g., government documents and SPD documents, we demonstrated the value of using PubLayNet in a transfer learning setting.

VI. CONCLUSION

We automatically generated the PubLayNet dataset, which is the largest ever available document layout annotation dataset exploiting redundancy in PCMOA. This dataset allows state-of-the-art object detection algorithms to be trained delivering high performance layout recognition on biomedical articles. Furthermore, this dataset is shown to be helpful to pre-train object detection algorithms to identify tables and different document layout objects in health insurance documents. These results are encouraging since the developed dataset is potentially helpful for document layout annotation of other domains. PubLayNet is available from https://github.com/ibm-aur-nlp/PubLayNet.

As future work, we plan to exploit PMCOA for the automatic generation of large datasets to solve other document analysis problems with deep learning models. For example, PubLayNet does not contain relationships between the layout elements, e.g., a paragraph and a section title. Such information is available in the XML representation and can be exploit to automatically create a dataset of the logical structure of documents.

ACKNOWLEDGMENT

REFERENCES

[1] P. W. J. Staar, M. Dolfi, C. Auer, and C. Bekas, "Corpus conversion service: A machine learning platform to ingest documents at scale," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 774–782. [Online]. Available: http://doi.acm.org/10.1145/3219819.3219834

[2] R. Cattoni, T. Coianiz, S. Messelodi, and C. M. Modena, "Geometric layout analysis techniques for document image understanding: a review," *ITC-irst Technical Report*, vol. 9703, no. 09, 1998.

[3] T. M. Breuel, "Two geometric algorithms for layout analysis," in *International workshop on document analysis systems*. Springer, 2002, pp. 188–199.

[4] ——, "High performance document layout analysis," in *Proceedings of the Symposium on Document Image Understanding Technology*, 2003, pp. 209–218.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[6] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 1162–1167.

[7] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 296–300.

[8] C. Clausner, C. Papadopoulos, S. Pletschacher, and A. Antonacopoulos, "The enp image and ground truth dataset of historical newspapers," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 931–935.

[9] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "Icdar2017 competition on recognition of documents with complex layouts-rdcl2017," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 1404–1410.

[10] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel, "An open approach towards the benchmarking of table structure recognition systems," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 113–120.

[11] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 445–449.

[12] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *IICAI*, 2005, pp. 1773–1785.

[13] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim, and I. S. Na, "Table detection from document image using vertical arrangement of text blocks," *International Journal of Contents*, vol. 11, no. 4, pp. 77–85, 2015.

[14] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 254–261.

[15] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016, pp. 287–292.

[16] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 771–776.

[17] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv preprint arXiv:1804.06236*, 2018.

[18] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[20] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[21] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1449–1453.

[22] A. Silva, "Parts that add up to a whole: a framework for the analysis of tables," *Edinburgh University, UK*, 2010.