DocBank: A Benchmark Dataset for Document Layout Analysis

Minghao Li[1*], **Yiheng Xu**[2*], **Lei Cui**[2], **Shaohan Huang**[2],
Furu Wei[2], **Zhoujun Li**[1], **Ming Zhou**[2]
[1]Beihang University
[2]Microsoft Research Asia
{liminghao1630,lizj}@buaa.edu.cn
{v-yixu,lecu,shaohanh,fuwei,mingzhou}@microsoft.com

Abstract

Document layout analysis usually relies on computer vision models to understand documents while ignoring textual information that is vital to capture. Meanwhile, high quality labeled datasets with both visual and textual information are still insufficient. In this paper, we present **DocBank**, a benchmark dataset that contains 500K document pages with fine-grained token-level annotations for document layout analysis. DocBank is constructed using a simple yet effective way with weak supervision from the LaTeX documents available on the arXiv.com. With DocBank, models from different modalities can be compared fairly and multi-modal approaches will be further investigated and boost the performance of document layout analysis. We build several strong baselines and manually split train/dev/test sets for evaluation. Experiment results show that models trained on DocBank accurately recognize the layout information for a variety of documents. The DocBank dataset is publicly available at https://github.com/doc-analysis/DocBank.

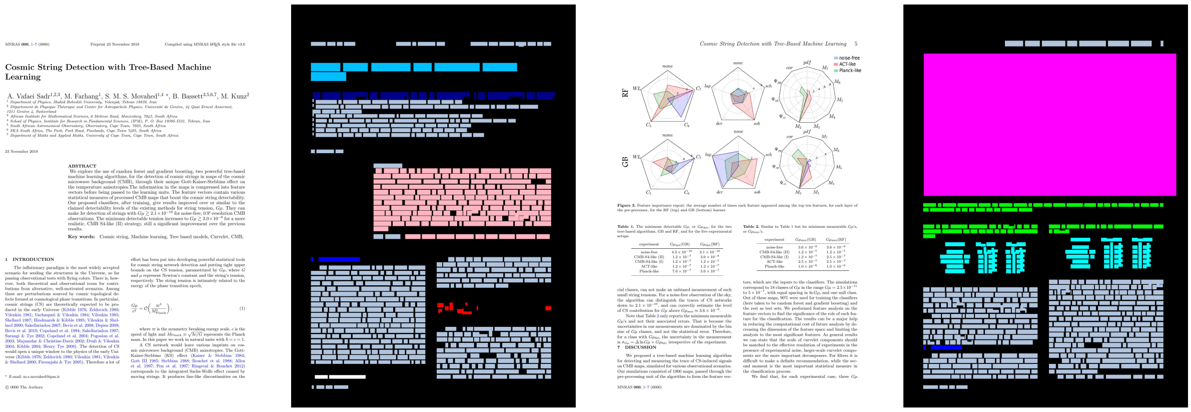
1 Introduction

Document layout analysis is an important task in many document understanding applications as it can transform semi-structured information into a structured representation, meanwhile extracting key information from the documents. It is a challenging problem due to the varying layouts and formats of the documents. Existing techniques have been proposed based on conventional rule-based or machine learning methods, where most of them fail to generalize well because they rely on hand crafted features that may be not robust to layout variations. Recently, the rapid development of deep learning in computer vision has significantly boosted the data-driven image-based approaches for document layout analysis. Although these approaches have been widely adopted and made significant progress, they usually leverage visual features while neglecting textual features from the documents. Therefore, it is inevitable to explore how to leverage the visual and textual information in a unified way for document layout analysis.

Nowadays, the state-of-the-art computer vision and NLP models are often built upon the pre-trained models (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018; Lample and Conneau, 2019; Yang et al., 2019; Dong et al., 2019; Raffel et al., 2019; Xu et al., 2019) followed by fine-tuning on specific downstream tasks, which achieves very promising results. However, pre-trained models not only require large-scale unlabeled data for self-supervised learning, but also need high quality labeled data for task-specific fine-tuning to achieve good performance. For document layout analysis tasks, there have been some image-based document layout datasets, while most of them are built for computer vision approaches and they are difficult to apply to NLP methods. In addition, image-based datasets mainly include the page images and the bounding boxes of large semantic structures, which are not fine-grained token-level annotations. Moreover, it is also time-consuming and labor-intensive to produce human-labeled and fine-grained token-level text block arrangement. Therefore, it is vital to leverage weak

[*]Equal contributions during internship at Microsoft Research Asia.

Proceedings of the 28th International Conference on Computational Linguistics, pages 949–960
Barcelona, Spain (Online), December 8-13, 2020



(a) (b) (c) (d)

Figure 1: Example annotations of the DocBank. The colors of semantic structure labels are: Abstract, Author, Caption, Equation, Figure, Footer, List, Paragraph, Reference, Section, Table, Title

supervision to obtain fine-grained labeled documents with minimum efforts, meanwhile making the data be easily applied to any NLP and computer vision approaches.

To this end, we build the DocBank dataset, a document-level benchmark that contains 500K document pages with fine-grained token-level annotations for layout analysis. Distinct from the conventional human-labeled datasets, our approach obtains high quality annotations in a simple yet effective way with weak supervision. Inspired by existing document layout annotations (Siegel et al., 2018; Li et al., 2019; Zhong et al., 2019), there are a great number of digital-born documents such as the PDFs of research papers that are compiled by LaTeX using their source code. The LaTeX system contains the explicit semantic structure information using mark-up tags as the building blocks, such as abstract, author, caption, equation, figure, footer, list, paragraph, reference, section, table and title. To distinguish individual semantic structures, we manipulate the source code to specify different colors to the text of different semantic units. In this way, different text zones can be clearly segmented and identified as separate logical roles, which is shown in Figure 1. The advantage of DocBank is that, it can be used in any sequence labeling models from the NLP perspective. Meanwhile, DocBank can also be easily converted into image-based annotations to support object detection models in computer vision. In this way, models from different modalities can be compared fairly using DocBank, and multi-modal approaches will be further investigated and boost the performance of document layout analysis. To verify the effectiveness of DocBank, we conduct experiments using four baseline models: 1) BERT (Devlin et al., 2018), a pre-trained model using only textual information based on the Transformer architecture. 2) RoBERTa (Liu et al., 2019), a robustly optimized method for pre-training the Transformer architecture. 3) LayoutLM (Xu et al., 2019), a multi-modal architecture that integrates both the text information and layout information. 4) Faster R-CNN (Ren et al., 2015), a high performance object detection networks depending on region proposal algorithms to hypothesize object locations. The experiment results show that the LayoutLM model significantly outperforms the BERT and RoBERTa models and the object detection model on DocBank for document layout analysis. We hope DocBank will empower more document layout analysis models, meanwhile promoting more customized network structures to make substantial advances in this area.

The contributions of this paper are summarized as follows:

- We present DocBank, a large-scale dataset that is constructed using a weak supervision approach. It enables models to integrate both the textual and layout information for downstream tasks.

- We conduct a set of experiments with different baseline models and parameter settings, which confirms the effectiveness of DocBank for document layout analysis.

- The DocBank dataset is available at `https://github.com/doc-analysis/DocBank`.

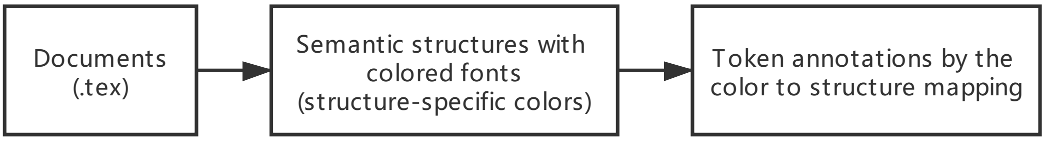

Documents (.tex)
Semantic structures with colored fonts (structure-specific colors)
Token annotations by the color to structure mapping

Figure 2: Data processing pipeline

2 Task Definition

The document layout analysis task is to extract the pre-defined semantic units in visually rich documents. Specifically, given a document $\mathcal{D}$ composed of discrete token set $t = \{t_0, t_1, ..., t_n\}$, each token $t_i = (w, (x_0, y_0, x_1, y_1))$ consists of word w and its bounding box (x_0, y_0, x_1, y_1). And $\mathcal{C} = \{c_0, c_1, .., c_m\}$ defines the semantic categories that the tokens are classified into. We intend to find a function $F : (\mathcal{C}, \mathcal{D}) \rightarrow \mathcal{S}$, where $\mathcal{S}$ is the prediction set:

$$\mathcal{S} = \{(\{t_0^0, ..., t_0^{n_0}\}, c_0), ..., (\{t_k^0, ..., t_k^{n_k}\}, c_k)\} \tag{1}$$

3 DocBank

We build DocBank with token-level annotations that supports both NLP and computer vision models. As shown in Figure 2, the construction of DocBank has three steps: Document Acquisition, Semantic Structures Detection, Token Annotation. Meanwhile, DocBank can be converted to the format that is used by computer vision models in a few steps. The current DocBank dataset totally includes 500K document pages, where the training set includes 400K document pages and both the validation set and the test set include 50K document pages.

3.1 Document Acquisition

We download the PDF files on arXiv.com as well as the LaTeX source files since we need to modify the source code to detect the semantic structures. The papers contain Physics, Mathematics, Computer Science and many other areas, which is beneficial for the diversity of DocBank to produce robust models. We focus on English documents in this work and will expand to other languages in the future.

3.2 Semantic Structures Detection

DocBank is a natural extension of the TableBank dataset (Li et al., 2019), where other semantic units are also included for document layout analysis. In this work, the following semantic structures are annotated in DocBank: {Abstract, Author, Caption, Equation, Figure, Footer, List, Paragraph, Reference, Section, Table and Title}. In TableBank, the tables are labeled with the help of the 'fcolorbox' command. However, for DocBank, the target structures are mainly composed of text, where the 'fcolorbox' cannot be well applied. Therefore, we use the 'color' command to distinguish these semantic structures by changing their font colors into structure-specific colors. Basically, there are two types of commands to represent semantic structures. Some of the LaTeX commands are simple words preceded by a backslash. For instance, the section titles in LaTeX documents are usually in the format as follows:

```
\section{The title of this section}
```

Other commands often start an environment. For instance, the list declaration in LaTeX documents is shown as follows:

```
\begin{itemize}
    \item First item
    \item Second item
\end{itemize}
```

The command $\backslash begin\{itemize\}$ starts an environment while the command $\backslash end\{itemize\}$ ends that environment. The real command name is declared as the parameters of the 'begin' command and the 'end' command.

We insert the ‘color’ command to the code of the semantic structures as follows and re-compile the LaTeX documents. Meanwhile, we also define specific colors for all the semantic structures to make them distinguishable. Different structure commands require the ‘color’ command to be placed in different locations to take effect. Finally, we get updated PDF pages from LaTeX documents, where the font color of each target structure has been modified to the structure-specific color.

```
\section{{\color{fontcolor}{The title of this section}}}
```

```
{\color{fontcolor}{\title{The title of this article}}}
```

```
\begin{itemize}
{\color{fontcolor}{
    \item First item
    \item Second item
}}
\end{itemize}
```

```
{\color{fontcolor}{
    \begin{equation}
        ...
        ...
    \end{equation}
}}
```

3.3 Token Annotation

We use PDFPlumber[1], a PDF parser built on PDFMiner[2], to extract text lines and non-text elements with their bounding boxes. Text lines are tokenized simply by white spaces, and the bounding boxes are defined as the most upper-left coordinate of characters and the most lower-right coordinate of characters, since we can only get the coordinates of characters instead of the whole tokens from the parser. For the elements without any texts such as figures and lines in PDF files, we use the class name inside PDFMiner and wrap it using two “#” symbols into a special token. The class names include “LTFigure” and “LTLine” that represent figures and lines respectively.

The RGB values of characters and the non-text elements can be extracted by PDFPlumber from the PDF files. Mostly, a token is composed of characters with the same color. Otherwise, we use the color of the first characters as the color of the token. We determine the labels of the tokens according to the color-to-structure mapping in the Section 3.2. A structure may contain both text and not-text elements. For instance, tables consist of words and lines. In this work, both words and lines will be annotated as the “table” class, so as to obtain the layout of a table as much as possible after the elements are tokenized.

3.4 Object Detection Annotation

The DocBank can be easily converted to the annotation format of the object detection models, like Faster R-CNN. The object detection models accept document images, the bounding boxes of semantic structures as input.

We classify all the token by the type of semantic structures on a page of the document. For the tokens of the same label, we use breadth-first search to find the Connected Component. We set an x-threshold and a y-threshold. If both of the x coordinates and the y coordinates of two tokens are within the thresholds, they are “Connected”. The breadth-first search is used to find all the tokens are connected to each other, which form an object of this label. Repeat the above steps to find all the objects. The bounding box of an object is determined by the most boundary tokens.

3.5 Dataset Statistics

The DocBank dataset consists of 500K document pages with 12 types of semantic units. Table 1 provides the statistics of training, validation, and test set in DocBank, showing that the number of every semantic unit and the percentage of pages with it. As these document pages are randomly drawn to generate training, validation, and test set, the distribution of semantic units in different splits are almost consistent.

We also show the distribution of document pages across years in Table 2. We can see that the number of papers is increasing year by year. To preserve this natural distribution, we randomly sample documents of different years to build DocBank without balancing them.

[1] https://github.com/jsvine/pdfplumber

[2] https://github.com/euske/pdfminer

Table 3 provides a comparison of the DocBank to the previous document layout analysis datasets, including Article Regions (Soto and Yoo, 2019), GROTOAP2 (Tkaczyk et al., 2014), PubLayNet (Zhong et al., 2019), and TableBank (Li et al., 2019). As shown in the table, DocBank surpasses the existing datasets in both size and number of types of semantic structures. All the listed datasets are image-based while only DocBank supports both text-based and image-based models. Meanwhile, DocBank are built automatically based on the public papers, so it is extendable, which is very rare in existing datasets.

Split	Abstract	Author	Caption	Equation	Figure	Footer	List	Paragraph	Reference	Section	Table	Title
Train	25,387	25,909	106,723	161,140	90,429	38,482	44,927	398,086	44,813	180,774	19,638	21,688
	6.35%	6.48%	26.68%	40.29%	22.61%	9.62%	11.23%	99.52%	11.20%	45.19%	4.91%	5.42%
Validation	3,164	3,286	13,443	20,154	11,463	4,804	56,09	49,759	55,49	22,666	2,374	2,708
	6.33%	6.57%	26.89%	40.31%	22.93%	9.61%	11.22%	99.52%	11.10%	45.33%	4.75%	5.42%
Test	3,176	3,277	13,476	20,244	11,378	4,876	5,553	49,762	5,641	22,384	2,505	2,729
	6.35%	6.55%	26.95%	40.49%	22.76%	9.75%	11.11%	99.52%	11.28%	44.77%	5.01%	5.46%
All	31,727	32,472	133,642	201,538	113,270	48,162	56,089	497,607	56,003	225,824	24,517	27,125
	6.35%	6.49%	26.73%	40.31%	22.65%	9.63%	11.22%	99.52%	11.20%	45.16%	4.90%	5.43%

Table 1: Semantic Structure Statistics of training, validation, and test sets in DocBank

Year	Train		Validation		Test		All	
2014	65,976	16.49%	8,270	16.54%	8,112	16.22%	82,358	16.47%
2015	77,879	19.47%	9,617	19.23%	9,700	19.40%	97,196	19.44%
2016	87,006	21.75%	10,970	21.94%	10,990	21.98%	108,966	21.79%
2017	91,583	22.90%	11,623	23.25%	11,464	22.93%	114,670	22.93%
2018	77,556	19.39%	9,520	19.04%	9,734	19.47%	96,810	19.36%
Total	400,000	100.00%	50,000	100.00%	50,000	100.00%	500,000	100.00%

Table 2: Year Statistics of training, validation, and test sets in DocBank

Dataset	**#Pages**	**#Units**	**Image-based?**	**Text-based?**	**Fine-grained?**	**Extendable?**
Article Regions	100	9	✓	✗	✓	✗
GROTOAP2	119,334	22	✓	✗	✗	✗
PubLayNet	364,232	5	✓	✗	✓	✗
TableBank	417,234	1	✓	✗	✓	✓
DocBank	500,000	12	✓	✓	✓	✓

Table 3: Comparison of DocBank with existing document layout analysis datasets

4 Method

As the dataset was fully annotated at token-level, we consider the document layout analysis task as a text-based sequence labeling task. Under this setting, we evaluate three representative pre-trained language models on our dataset including BERT, RoBERTa and LayoutLM to validate the effectiveness of DocBank. To verify the performance of the models from different modalities on DocBank, we train the Faster R-CNN model on the object detection format of DocBank and unify its output with the sequence labeling models to evaluate.

4.1 Models

The BERT Model BERT is a Transformer-based language model trained on large-scale text corpus. It consists of a multi-layer bidirectional Transformer encoder. It accepts a token sequence as input and calculates the input representation by summing the corresponding token, segment, and position embeddings. Then, the input vectors pass multi-layer attention-based Transformer blocks to get the final contextualized language representation.

The RoBERTa Model RoBERTa (Liu et al., 2019) is a more powerful version of BERT, which has been proven successfully in many NLP tasks. Basically, the model architecture is the same as BERT except for the tokenization algorithm and improved training strategies. By increasing the size of the pre-training data and the number of training steps, RoBERTa gets better performance on several downstream tasks.

The LayoutLM Model LayoutLM is a multi-modal pre-trained language model that jointly models the text and layout information of visually rich documents. In particular, it has an additional 2-D position embedding layer to embed the spatial position coordinates of elements. In detail, the LayoutLM model accepts a sequence of tokens with corresponding bounding boxes in documents. Besides the original embeddings in BERT, LayoutLM feeds the bounding boxes into the additional 2-D position embedding layer to get the layout embeddings. Then the summed representation vectors pass the BERT-like multi-layer Transformer encoder. Note that we use the LayoutLM without image embeddings and more details are provided in the Section 4.2.

The Faster R-CNN Model Faster R-CNN is one of the most popular object detection networks. It proposes the Region Proposal Network (RPN) to address the bottleneck of region proposal computation. RPN shares convolutional features with the detection network using 'attention' mechanisms, which leads to nearly cost-free region proposals and high accuracy on many object detection benchmarks.

4.2 Pre-training LayoutLM

LayoutLM chooses the Masked Visual-Language Model(MVLM) and Multi-label Document Classication(MDC) as the objectives when pre-training the model. For the MVLM task,its procedure is to simply mask some of the input tokens at random keeping the corresponding position embedding and then predict those masked tokens. In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary. For the MDC task, it uses the output context vector of [CLS] token to predict the category labels. With these two training objectives, the LayoutLM is pre-trained on IIT-CDIP Test Collection 1.0[3] (Lewis et al., 2006), a large document image collection.

4.3 Training Samples in Reading Order

We organize the DocBank dataset using the reading order, which means that we sort all the text boxes (a hierarchy level higher than text line in PDFMiner) and non-text elements from top to bottom by their top border positions. The text lines inside a text box are already sorted top-to-bottom. We tokenize all the text lines in the left-to-right order and annotate them. Basically, all the tokens are arranged top-to-bottom and left-to-right, which is also applied to all the columns of multi-column documents.

4.4 Fine-tuning

We fine-tune the pre-trained model with the DocBank dataset. As the document layout analysis is regarded as a sequence labeling task, all the tokens are labeled using the output with the maximum probability. The number of output class equals the number of semantic structure types.

5 Experiment

5.1 Evaluation Metrics

As the inputs of our model are serialized 2-D documents, the typical BIO-tagging evaluation is not suitable for our task. The tokens of each semantic unit may discontinuously distribute in the input sequence. In this case, we proposed a new metric, especially for text-based document layout analysis methods. For each kind of document semantic structure, we calculated their metrics individually. The definition is as follows:

[3] https://ir.nist.gov/cdip/

$$Precision = \frac{\text{Area of Ground truth tokens in Detected tokens}}{\text{Area of all Detected tokens}},$$
$$Recall = \frac{\text{Area of Ground truth tokens in Detected tokens}}{\text{Area of all Ground truth tokens}},$$
$$F1\ Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}.$$

5.2 Settings

Our baselines of BERT and RoBERTa are built upon the HuggingFace's Transformers (Wolf et al., 2019) while the LayoutLM baselines are implemented with the codebase in LayoutLM's official repository[4]. We use 8 V100 GPUs with a batch size of 10 per GPU. It takes 5 hours to fine-tune 1 epoch on the 400K document pages. We use the BERT and RoBERTa tokenizers to tokenize the training samples and optimized the model with AdamW. The initial learning rate of the optimizer is 5×10^{-5}. We split the data into a max block size of $N = 512$. We use the Detectron2 (Wu et al., 2019) to train the Faster R-CNN model on DocBank. We use the Faster R-CNN algorithm with the ResNeXt (Xie et al., 2017) as the backbone network architecture, where the parameters are pre-trained on the ImageNet dataset.

5.3 Results

Models	Abstract	Author	Caption	Equation	Figure	Footer	List	Paragraph	Reference	Section	Table	Title	Macro average
$\text{BERT}_{\text{BASE}}$	0.9294	0.8484	0.8629	0.8152	1.0000	0.7805	0.7133	0.9619	0.9310	0.9081	0.8296	0.9442	0.8770
$\text{RoBERTa}_{\text{BASE}}$	0.9288	0.8618	0.8944	0.8248	1.0000	0.8014	0.7353	0.9646	0.9341	0.9337	0.8389	0.9511	0.8891
$\text{LayoutLM}_{\text{BASE}}$	**0.9816**	0.8595	0.9597	0.8947	1.0000	0.8957	0.8948	0.9788	0.9338	0.9598	0.8633	**0.9579**	0.9316
$\text{BERT}_{\text{LARGE}}$	0.9286	0.8577	0.8650	0.8177	1.0000	0.7814	0.6960	0.9619	0.9284	0.9065	0.8320	0.9430	0.8765
$\text{RoBERTa}_{\text{LARGE}}$	0.9479	0.8724	0.9081	0.8370	1.0000	0.8392	0.7451	0.9665	0.9334	0.9407	0.8494	0.9461	0.8988
$\text{LayoutLM}_{\text{LARGE}}$	0.9784	0.8783	0.9556	0.8974	**1.0000**	0.9146	0.9004	0.9790	0.9332	0.9596	0.8679	0.9552	0.9350
X101	0.9717	0.8227	0.9435	0.8938	0.8812	0.9029	0.9051	0.9682	0.8798	0.9412	0.8353	0.9158	0.9051
$\text{X101+LayoutLM}_{\text{BASE}}$	0.9815	0.8907	**0.9669**	0.9430	0.9990	0.9292	**0.9300**	0.9843	**0.9437**	0.9664	0.8818	0.9575	0.9478
$\text{X101+LayoutLM}_{\text{LARGE}}$	0.9802	**0.8964**	0.9666	**0.9440**	0.9994	**0.9352**	0.9293	**0.9844**	0.9430	**0.9670**	**0.8875**	0.9531	**0.9488**

Table 4: The performance of BERT, RoBERTa, LayoutLM and Faster R-CNN on the DocBank test set.

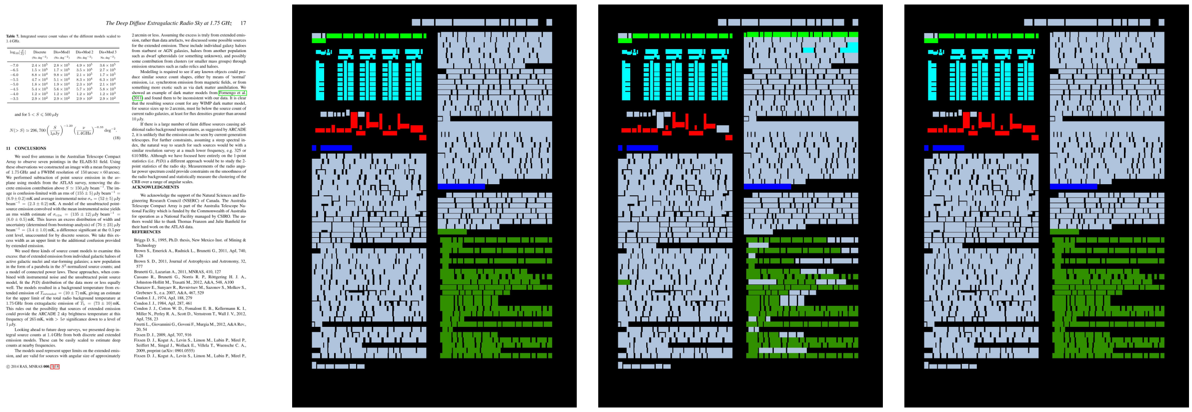
The evaluation results of BERT, RoBERTa and LayoutLM are shown in Table 4. We evaluate six models on the test set of DocBank. We notice that the LayoutLM gets the highest scores on the {abstract, author, caption, equation, figure, footer, list, paragraph, section, table, title} labels. The RoBERTa model gets the best performance on the "reference" label but the gap with the LayoutLM is very small. This indicates that the LayoutLM architecture is significantly better than the BERT and RoBERTa architecture in the document layout analysis task.

We also evaluate the ResNeXt-101 model and two ensemble models combining ResNeXt-101 and LayoutLM. The output of the ResNeXt-101 model is the bounding boxes of semantic structures. To unify the outputs of them, we mark the tokens inside each bounding box by the label of the corresponding bounding box. After that, we calculate the metrics following the equation in Section 5.1.

6 Case Study

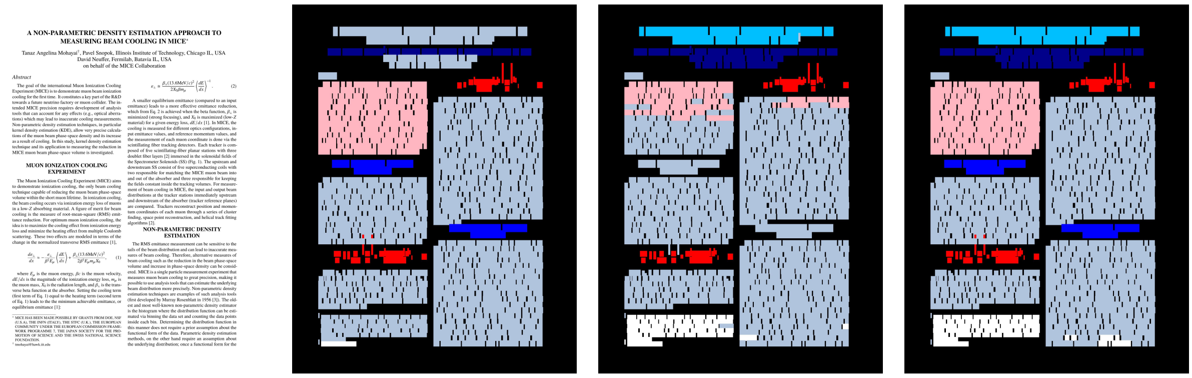
We visualize the outputs of pre-trained BERT and pre-trained LayoutLM on some samples of the test set in Figure 3 and Figure 4. Generally, it is observed that the sequence labeling method performs well on the DocBank dataset, where different semantic units can be identified. For the pre-trained BERT model, we can see some tokens are detected incorrectly, which illustrates that only using text information is still not sufficient for document layout analysis tasks, and visual information should be considered as well. Compared with the pre-trained BERT model, the pre-trained LayoutLM model integrates both the text and layout information. Therefore, it produces much better performance on the benchmark dataset. This is because the 2D position embeddings can model spatial distance and boundary of semantic structures in a unified framework, which leads to the better detection accuracy.

[4] `https://aka.ms/layoutlm`



(a) Original document page (b) Groundtruth (c) Pre-trained BERT (d) Pre-trained LayoutLM

Figure 3: Example output of pre-trained LayoutLM and pre-trained BERT on the test set



(a) Original document page (b) Groundtruth (c) Pre-trained BERT (d) Pre-trained LayoutLM

Figure 4: Example output of pre-trained LayoutLM and pre-trained BERT on the test set

7 Related Work

The research of document layout analysis can be divided into three categories: rule-based approaches, conventional machine learning approaches, and deep learning approaches.

7.1 Rule-based Approaches

Most of the rule-based works (Lebourgeois et al., 1992; Ha et al., 1995a; Simon et al., 1997; Ha et al., 1995b) are divided into two main categories: the bottom-up approaches and the top-down approaches.

Some bottom-up approaches (Lebourgeois et al., 1992; Ha et al., 1995a; Simon et al., 1997) first detect the connected components of black pixels as the basic computational units in document image analysis. The main part of the document segment process is combining them into higher-level structures through different heuristics methods and labeling them according to different structural features. The spatial auto-correlation approach (Journet et al., 2005; Journet et al., 2008) is a bottom-up texture-based method for document layout analysis. It starts by extracting texture features directly from the image pixels to form homogeneous regions and will auto-correlate the document image with itself to highlight periodicity and texture orientation.

For the top-down strategy, (Jain and Zhong, 1996) proposed a mask-based texture analysis to locate text regions written in different languages. Run Length Smearing Algorithm converts image-background to image-foreground if the number of background pixels between any two consecutive foreground pixels is less than a predefined threshold, which is first introduced by (Wahl et al., 1982). Document projection profile method was proposed to detect document regions(Shafait and Breuel, 2010). (Nagy and Seth, 1984) proposed a X-Y cut algorithm that used projection profile to determine document blocks cuts. For the above work, the rule-based heuristic algorithm is difficult to process complex documents, and the

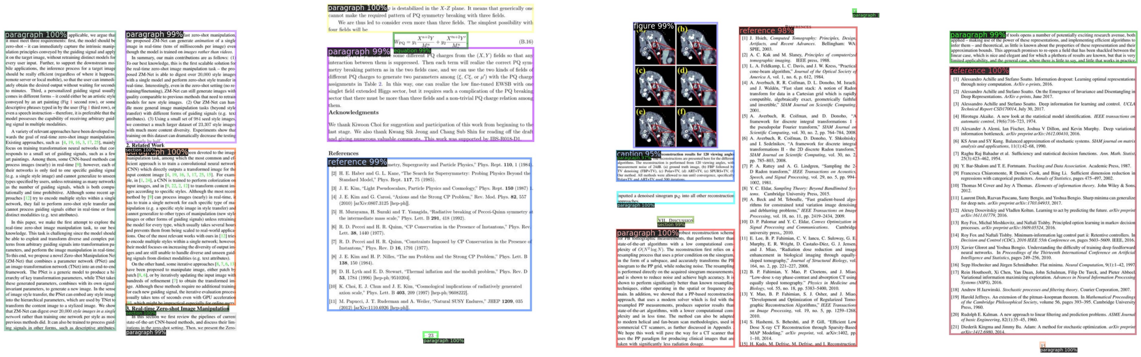


Figure 5: Example output of Faster R-CNN on the test set

applicable document types are relatively simple.

7.2 Conventional Machine Learning Approaches

To address the issue about data imbalance that the learning-based methods suffer from, a dynamic MLP (DMLP) was proposed to learn a less-biased machine model using pixel-values and context information (Baechler et al., 2013). Usually, block and page-based analysis require feature extraction methods to empower the training and build robust models. The handcrafted features are developed through feature extraction techniques such as Gradient Shape Feature (GSF) (Diem et al., 2011) or Scale Invariant Feature Transform (SIFT) (Garz et al., 2010; Garz et al., 2012; Garz et al., 2011; Wei et al., 2014a). There are several other techniques that use features extraction methods such as texture features (Chen et al., 2015; Mehri et al., 2013; Mehri et al., 2017; Mehri et al., 2015; Wei et al., 2013; Wei et al., 2014b) and geometric features (Bukhari et al., 2010; Bukhari et al., 2012). Manually designing features require a large amount of work and is difficult to obtain a highly abstract semantic context. Moreover, the above machine learning methods rely solely on visual cues and ignore textual information.

7.3 Deep Learning Approaches

The learning-based document layout analysis methods get more attention to address complex layout analysis. (Capobianco et al., 2018) suggested a Fully Convolutional Neural Network (FCNN) with a weight-training loss scheme, which was designed mainly for text-line extraction, while the weighting loss in FCNN can help in balancing the loss function between the foreground and background pixels. Some deep learning methods may use weights of pre-trained networks. A study by (Oliveira et al., 2018) proposed a multi-task document layout analysis approach using Convolution Neural Network (CNN), which adopted transfer learning using ImageNet. (Yang et al., 2017) treats the document layout analysis tasks as a pixel-by-pixel classification task. He proposed an end-to-end multi-modal network that contains visual and textual information.

8 Conclusion

To empower the document layout analysis research, we present DocBank with 500K high-quality document pages that are built in an automatic way with weak supervision, which enables document layout analysis models using both textual and visual information. To verify the effectiveness of DocBank, we conduct an empirical study with four baseline models, which are BERT, RoBERTa, LayoutLM and Faster R-CNN. Experiment results show that the methods integrating text and layout information is a promising research direction with the help of DocBank. We expect that DocBank will further release the power of other deep learning models in document layout analysis tasks.

9 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.U1636211, 61672081, 61370126), the Beijing Advanced Innovation Center for Imaging Technology (Grant No.BAICIT-2016001), and the Fund of the State Key Laboratory of Software Development Environment (Grant No.SKLSDE-2019ZX-17).

References

Micheal Baechler, Marcus Liwicki, and Rolf Ingold. 2013. Text line extraction using dmlp classifiers for historical manuscripts. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1029–1033. IEEE.

Syed Saqib Bukhari, Al Azawi, Mayce Ibrahim Ali, Faisal Shafait, and Thomas M Breuel. 2010. Document image segmentation using discriminative learning over connected components. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 183–190. ACM.

Syed Saqib Bukhari, Thomas M Breuel, Abedelkadir Asi, and Jihad El-Sana. 2012. Layout analysis for arabic historical document images using machine learning. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 639–644. IEEE.

Samuele Capobianco, Leonardo Scommegna, and Simone Marinai. 2018. Historical handwritten document segmentation by using a weighted loss. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 395–406. Springer.

Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. 2015. Page segmentation of historical document images with convolutional autoencoders. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1011–1015. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Markus Diem, Florian Kleber, and Robert Sablatnig. 2011. Text classification and document layout analysis of paper fragments. In *2011 International Conference on Document Analysis and Recognition*, pages 854–858. IEEE.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *ArXiv*, abs/1905.03197.

Angelika Garz, Markus Diem, and Robert Sablatnig. 2010. Detecting text areas and decorative elements in ancient manuscripts. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 176–181. IEEE.

Angelika Garz, Robert Sablatnig, and Markus Diem. 2011. Layout analysis for historical manuscripts using sift features. In *2011 International Conference on Document Analysis and Recognition*, pages 508–512. IEEE.

Angelika Garz, Andreas Fischer, Robert Sablatnig, and Horst Bunke. 2012. Binarization-free text line segmentation for historical documents based on interest point clustering. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 95–99. IEEE.

Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. 1995a. Document page decomposition by the bounding-box project. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 1119–1122. IEEE.

Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. 1995b. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955. IEEE.

Anil K Jain and Yu Zhong. 1996. Page segmentation using texture analysis. *Pattern recognition*, 29(5):743–770.

Nicholas Journet, Véronique Eglin, Jean-Yves Ramel, and Rémy Mullot. 2005. Text/graphic labelling of ancient printed documents. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1010–1014. IEEE.

Nicholas Journet, Jean-Yves Ramel, Rémy Mullot, and Véronique Eglin. 2008. Document image characterization using a multiresolution analysis of the texture: application to old documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 11(1):9–18.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Frank Lebourgeois, Z Bublinski, and H Emptoz. 1992. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, pages 272–276. IEEE.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 665–666, New York, NY, USA. ACM.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019. Tablebank: Table benchmark for image-based table detection and recognition. *arXiv preprint arXiv:1903.01949*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Maroua Mehri, Petra Gomez-Krämer, Pierre Héroux, Alain Boucher, and Rémy Mullot. 2013. Texture feature evaluation for segmentation of historical document images. In *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*, pages 102–109. ACM.

Maroua Mehri, Nibal Nayef, Pierre Héroux, Petra Gomez-Krämer, and Rémy Mullot. 2015. Learning texture features for enhancement and segmentation of historical document images. In *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*, pages 47–54. ACM.

Maroua Mehri, Pierre Héroux, Petra Gomez-Krämer, and Rémy Mullot. 2017. Texture feature benchmarking and evaluation for historical document image analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, 20(1):1–35.

George Nagy and Sharad C Seth. 1984. Hierarchical representation of optically scanned documents.

Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. 2018. dhsegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12. IEEE.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497.

Faisal Shafait and Thomas M Breuel. 2010. The effect of border noise on the performance of projection-based page segmentation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):846–851.

Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*.

Anikó Simon, J-C Pret, and A Peter Johnson. 1997. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277.

Carlos Soto and Shinjae Yoo. 2019. Visual detection with context for document layout analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3462–3468, Hong Kong, China, November. Association for Computational Linguistics.

Dominika Tkaczyk, Pawel Szostek, and Lukasz Bolikowski. 2014. Grotoap2 - the methodology of creating a large ground truth dataset of scientific articles. *D-Lib Mag.*, 20.

Friedrich M Wahl, Kwan Y Wong, and Richard G Casey. 1982. Block segmentation and text extraction in mixed text/image documents. *Computer graphics and image processing*, 20(4):375–390.

Hao Wei, Micheal Baechler, Fouad Slimane, and Rolf Ingold. 2013. Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1220–1224. IEEE.

Hao Wei, Kai Chen, Rolf Ingold, and Marcus Liwicki. 2014a. Hybrid feature selection for historical document layout analysis. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 87–92. IEEE.

Hao Wei, Kai Chen, Anguelos Nicolaou, Marcus Liwicki, and Rolf Ingold. 2014b. Investigation of feature selection for historical document layout analysis. In *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. `https://github.com/facebookresearch/detectron2`.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. Layoutlm: Pre-training of text and layout for document image understanding. *ArXiv*, abs/1912.13318.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. 2019. Publaynet: Largest dataset ever for document layout analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.