CTE: A Dataset for Contextualized Table Extraction

Andrea **Gemelli**[1,*,†], Emanuele **Vivoli**[1,†] and Simone **Marinai**[1,†]

[1]Dipartimento di Ingegneria dell'Informazione (DINFO) Università degli studi di Firenze, Italy

Abstract

Relevant information in documents is often summarized in tables, helping the reader to identify useful facts. Most benchmark datasets support either document layout analysis or table understanding, but lack in providing data to apply both tasks in a unified way. We define the task of Contextualized Table Extraction (CTE), which aims to extract and define the structure of tables considering the textual context of the document. The dataset comprises 75k fully annotated pages of scientific papers, including more than 35k tables. Data are gathered from PubMed Central, merging the information provided by annotations in the PubTables-1M and PubLayNet datasets. The dataset can support CTE and adds new classes to the original ones. The generated annotations can be used to develop end-to-end pipelines for various tasks, including document layout analysis, table detection, structure recognition, and functional analysis. We formally define CTE and evaluation metrics, showing which subtasks can be tackled, describing advantages, limitations, and future works of this collection of data. Annotations and code will be accessible at https://github.com/AILab-UniFI/cte-dataset.

Keywords

Dataset, Table Extraction, Scientific Paper Analysis, Document Layout Analysis, Benchmark

1. Introduction

Nowadays, large collections of documents require a huge amount of human work to annotate documents and extract important information. In the last thirty years, the community of Document Analysis and Recognition (DAR) tried to overcome this challenge, exploiting suitable algorithms and artificial intelligence techniques to automatize the analysis of documents and reduce its costs. Among others, Document Classification (DC), Layout Analysis (DLA), and Table Understanding (TU) more broadly attracted the interest of researchers and companies. DC is the first step of many DAR pipelines, since different kinds of documents require different strategies: given a document, either scanned or digital-born, the aim is to classify it into a specific category, e.g. invoice or magazine. DLA [1] aims at recognizing homogeneous regions within the document, grouping smaller components close to each other such as regions of text, and, if required, assigning it a category (e.g. a title or an image caption). Finally, TU [2] is an

umbrella term for table detection and recognition: tables summarize important information within documents and their detection along with the recognition of their structure is crucial to automatically query collections of documents.

During the past years, the interest in the detection and recognition of tables raised significantly, leading to the automation of important processes such as information extraction. In particular, for scientific literature, it is crucial to extract tabular data, e.g. to make the research comparable and help scholars to reconstruct the SOTA of the different fields of study [3]. Moreover, collections of scientific papers such as arXiv and PubMed opened to the possibility of accessing a large number of documents along with their structural information represented in standard formats such as LaTeX and XML. That is why scientific literature parsing and scientific table analysis rapidly became one of the most prominent areas of research in DAR: large datasets have been released [4, 5], allowing the community to develop deep learning models. Unfortunately, as we will describe in the next sections, these datasets come with partial information that forces the experimentation of layout analysis and table extraction separately. From this identified lack, we define Contextualized Table Extraction, a broad task that comes along with novel annotations for a collection of 75k scientific pages containing more than 35k tables, encouraging the development of new systems capable of tackling a multitude of tasks at once.

In this paper, we introduce a new task called Contextualized Table Extraction that is a framework, which involves detecting tables, recognizing their structure, and performing functional analysis in an end-to-end manner. CTE is formulated as a token and link classification task, which allows for multiple tasks to be addressed simultaneously overcoming common limitations such as being performed separately or lacking a comprehensive dataset. CTE is built on top of well known tasks in DAR. CTE is designed to be suitable for methods employing Graph Neural Networks, which are widely used in applications where the structure and layout in documents matter. We provide a new set of labels structured in a way that allows us to merge information of selected scientific publications from other well known benchmark datasets. In this way we obtain a comprehensive dataset for the task of CTE. We believe that the combination of methods applied to process the labeled documents and produce the merged information collected is a novel contribution to the field of document analysis as well.

1.1. Related Work

Despite the advances in the field, several challenges strongly limited the generalization of methods developed until a few years ago. In particular, we can mention: (i) data quality (e.g. scanned documents or images captured in-the-wild); (ii) contents, due to different languages and/or scripts; (iii) document layouts (which differentiate in, e.g. magazines, scientific papers, and invoices). To address these challenges a large number of data need to be collected in order to fully exploit the power of Deep Learning models that achieve the state-of-the-art for the aforementioned tasks. Unfortunately, creating such datasets is nothing but trivial since accurate annotations come at a high cost in terms of time and human effort [6, 7]. On the other hand, automatic annotation techniques are not always applicable since they require a large number of documents shared together with their source files in standard formats such as LaTeX, XML, or HTML [8, 4]. Additionally, these techniques usually generate weakly labeled collections and are more error-prone than manually annotated ones.

Table 1

Comparison of CTE with related datasets: ♣ denotes the datasets used to generate the new annotations. A dataset is $S4G$ (suitable for graphs) if a graph can be constructed directly with no further preprocessing. DLA (Document Layout Analysis), TD (Table Detection), TSR (Table Structure Recognition), and TFA (Table Functional Analysis) show which tasks models can be trained for.

Dataset	#pages	#tables	#classes	DLA	TD	TSR	TFA	S4G
PubLayNet (♣)	358k	107k	5	✓	✓	✗	✗	✗
PubTables-1M (♣)	574k	948k	7	✗	✓	✓	✓	✗
DocBank	500k	417k*	12	✓	✓	✓*	✗	✓**
SciTSR	0	15k	-	✗	✗	✓	✗	✓
CTE	75k	35k	13	✓	✓	✓	✓	✓

*DocBank is an extension of TableBank, from which we gathered these information

**If tokens used as graph nodes, no information on edges

Since online archives of scientific papers are freely and publicly available along with the corresponding source information (e.g. arXiv and PubMed) several datasets have been proposed so far in the field of scientific literature parsing. Among others, we summarize in Table 1 some of the most important datasets proposed for layout analysis and table extraction. PubLayNet and DocBank have been widely used to train object detectors [9, 10] and transformers [11] for DLA. Overall, these datasets contain around half a million pages labeled into five and twelve different classes, respectively. PubLayNet has been constructed merging the information extracted from PDFMiner (bounding box regions) and the XML files shared by the publishers (containing the region labels). DocBank is built gathering the LaTeX source files and assigning labels taking into account the section tags. For the Table Extraction task, a recent dataset has been released (PubTables-1M) which counts nearly one million tables, labeled to perform not only TD and TSR but also Table Functional Analysis (TFA) that provides additional information on table cells like table headers. Even if it is smaller, SciTSR [12] introduced a collection of 15k tables generated from LaTeX to perform TSR, mainly using a Graph Neural Network (GNN). Despite this contribution, GNNs also have the advantage of being lightweight compared to transformer-based architectures while still retaining good performance, as shown in the framework Doc2Graph [13] for document analysis.

As it is possible to notice in Table 1, all these datasets lack a comprehensive and broader set of annotations, forcing the community to develop multiple systems that, in application scenarios, would lead to heavy and large pipelines.

1.2. Contributions

Our ongoing work brings several novelties, that are discussed throughout the paper and are summarized as follows:

- We define the task of Contextualized Table Extraction, an extended version of table extraction as defined in [5] that adds layout information and encourages the development of end-to-end systems that can tackle multiple tasks at once;

- Novel annotations are created by merging subset of [4, 5] that can be found in our repo[1]. Our collection comprehends 75k scientific pages and more than 35k tables. Tokens at the basis of annotations correspond to words extracted from PDFs using PyMuPDF and labeled according to the region they belong to; table structure information is encoded as links between tokens;
- The dataset encourages the use and development of graph methods on documents, providing to the community a new set of labeled data to experiment with GNN-based techniques. The annotations do not require any further processing (either in labels or data themselves) to construct a graph over the scientific pages.

The paper is organized as follows: in Section 2 we describe in detail how the dataset has been created and how the annotations are presented, along with some limitations we aim to address in the near future. Section 3 formalizes the CTE task by means of token and link classification. Finally, in Section 4 and 5 we discuss future work and draw conclusions.

2. Dataset Description

Contextualized Table Extraction (CTE), as we describe deeply in Section 3, involves not only detecting tables, recognizing their layout and functional structure, but also takes into consideration their surrounding information. We formalize CTE to be accomplished through token and link classification, allowing multiple tasks to be tackled at once. The F1 score for CTE is defined as the average of F1 scores for token and link classification.

Although it is easy to freely access large collections of scientific papers (i.e. from arXiv or PubMed Central) it is difficult to find documents labeled with complete information. Most benchmark datasets support either DLA or TU. However, as our aim is encouraging the development of systems capable of tackling more tasks at once, a new dataset is needed. The proposed dataset for CTE is obtained by merging data and annotations given by PubLayNet and PubTables-1M datasets, both based on PubMed Central publications. As depicted in the next sections, firstly we identify the pages of scientific papers annotated in both datasets, then we merge the information and add two novel classes (captions and page information) and finally use PyMuPDF to extract text and position of tokens. We used a preliminary small version of this collection in [14], applying a GNN to tackle CTE. After the release of PubLayNet test set we updated the version of CTE dataset, now containing more annotated data.

2.1. Subset of PubLayNet and PubTables-1M

PubLayNet is a collection of $358, 353$ PDF pages with five types of regions annotated (*title, text, list, table, image*) [4]. PubTables-1M [5] is a collection of $947, 642$ fully annotated tables, including information for table detection, recognition, and functional analysis (such as identifying *column headers, projected rows*, and *table cells*). The datasets are built to address different tasks, as summarized in Table 1.

To merge the datasets, we first identify the papers belonging to both collections. From this subset, we keep pages with tables fully annotated in PubTables-1M and pages without

---

[1]https://github.com/AILab-UniFI/cte-dataset

tables: this filters out even more pages, since we found some PubTables-1M annotations to have only one annotated table in pages containing two or more tables. Following this step, we obtain approximately 75k pages. The resulting merged dataset contains objects labeled into 13 different classes, having in addition to the regions annotated in PubLayNet the table annotations described in PubTables-1M (*row, column, table header, projected header, table cell*, and *grid cell*). Moreover, we added two classes: *caption* and *other*. *Captions* are heuristically found taking into account the proximity with images and tables, while the *other* class contains all the remaining not-labeled text regions (e.g. page headers and page numbers).

The GitHub repository of our dataset is at its second version, after adding the test-set released by PubLayNet [2]. We followed PubLayNet for the train/val/test splits.

2.2. Annotation procedure

Once a complete annotated list of pages is selected from the two datasets, we leverage an external tool to extract page tokens. After comparing several tools, we opted for PyMuPDF [15] which is a Python open-source library backed by a large community and constantly maintained. Each element, visible or not visible, present in the PDF page is extracted and annotated based on the annotation bounding-box it appears in, as depicted in Figure 1: tokens are labeled according to their enclosing labeled region (upper part); links, instead, are presented as groups of tokens for visualization purposes (bottom part), but encoded as couples as described in details in the next Section and in Table 2. By doing so, the resulting page is composed by extracting page tokens along with their position (bounding boxes coordinates) and their textual content (mostly single words). This process heavily depends on original versions of the PDF files: even if the document name is the same along the two datasets annotations (PubLayNet and PubTables-1M) the PDF version of PubLayNet documents could differ. This is due to the two years gap between the datasets release date. To obtain reliable information, in our approach we discard all the pages (and tables) in which the content of the two sources does not correspond anymore.

2.3. Dataset structure and format

After the merging procedure, we end up with three JSON files (subset of the original PubLayNet one) splitting the data into train, val, and test. Each one contains information regarding tokens extracted by PyMuPDF, their links and the regions that group them (larger objects). Tokens have these information: *token id, bounding box coordinates, text, class id*, and *object id* (larger region to which it belongs). Links between tokens (belonging to the same row, column or grid cell) have information such as *link id, class id*, and *token id* (list of tokens linked together). Finally, objects contain information such as *object id, bounding box coordinates* and *class id*. A representation of the aforementioned annotation format is represented in Tables 2.
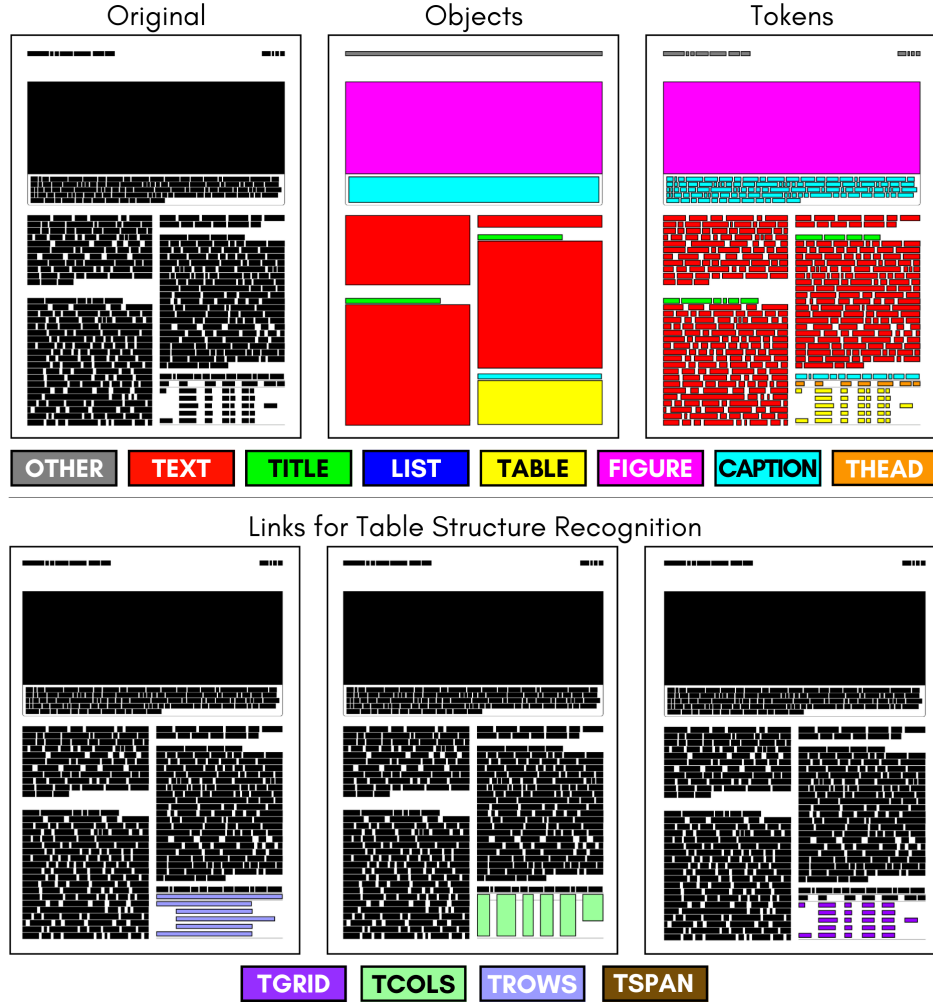

Figure 1: Example page (content is intentionally concealed in 'Original') and corresponding CTE annotations. Objects represent the layout regions. Tokens contain the word tokens labeled according to the class in the top of the figure. Acronyms for table annotations are: THEAD (table headers), TSPAN (table sub-headers spanning along different columns), TGRID (table cells), TCOLS and TROWS (respectively columns and rows of the tables).

2.4. Limitations of the Dataset

We are aware that the proposed dataset, even if it is proposing a new benchmark to tackle CTE, has room for improvement. As such, in the following we list the limitations of the dataset:

1. There is a small amount of data and tables compared to other datasets. Considering that adding more annotated data would be nothing but trivial, we believe this point could be addressed in two ways: i) as a starting pool of data to train generative models and getting new samples automatically labeled (e.g. using techniques similar to [16]); ii) using the CTE collection as a challenging benchmark to compare lightweight models, such as GNNs, along with state-of-the-art transformers (notably anger of huge amount of data).

Table 2
Annotation Format: each line contains different information in case of Objects, Tokens, or Links.

Object annotations

Index	0	1,0	1,1	1,2	1,3	2
Content	object id	x0	y0	x1	y1	class id

Tokens annotations

Index	0	1,0	1,1	1,2	1,3	2	3	4
Content	token id	x0	y0	x1	y1	text	class id	parent object id

Link annotations

Index	0	1	2,0	2,1	2,n-1
Content	link id	class id	1st token id	2nd token id	n-th token id

2. The heuristics used for the the classes *caption* and *other* could affect the generalization of trained models, highly dependent on the paper format used in PubMed Central. On the other hand, we are enriching information about tables by recognizing captions, that contain valuable table descriptions and that otherwise would be discarded.

3. We still lack additional information such as author, keywords, and equations. We are going to add these additional labels in the near future, considering Grobid [17] in the annotation procedure, since it is a machine learning library for extracting technical information from scientific publications, from PDF to XML/TEI structured documents.

4. The first attempts to define a baselines are reported in [14], in which the task of TE and DLA are treated end-to-end. This paper aims at sharing the CTE dataset in a way that the scientific community can further propose baselines on this work.

3. Contextualized Table Extraction

Contextualized Table Extraction (CTE) is the broader task of extracting tables (meaning their detection) recognizing their structure and performing functional analysis, along with other page layout information. To do so, CTE is formulated as a token and link classification tasks, similarly to [8], since fine-grained objects like tokens permit to tackle multiple tasks at once. For instance, recognizing the table headers and grid cells allows us to detect the tables (grouping tokens together through links) and add functional information. In addition, through token and link classification the need for more components would be reduced since a method capable of successfully solving CTE would require to train only one model, extracting more information at once.

Given Precision and Recall for token and link classification, namely Token Precision (TP), Token Recall (TR), Link Precision (LP), and Link Recall (LR). We can define the $F1_{CTE}$ metric as follows:

$$F1_{CTE} = \frac{F1_{Token} + F1_{Link}}{2} = \frac{TP \cdot TR}{TP + TR} + \frac{LP \cdot LR}{LP + LR}. \tag{1}$$

Token classification

The first step required to tackle CTE is the classification of tokens, extracted from PDF pages using PyMuPDF. Tokens contain textual and positional information, along with class information inherited from the larger region they belong to (details in Table 2, tokens annotations). This subtask exposes these properties:

1. Through token classification it is possible to achieve DLA, TD, and TFA at once.
2. If tackled along with link classification to achieve CTE the $F1_{CTE}$ metric (Eq. 1) should be used. Instead, if tackled alone the metric proposed in [8] can be used as well.

Link Classification
In order to group together tokens belonging to tables into columns, rows, or grid cells, additional information on links among pairs of tokens is added. This subtask exposes these properties:

1. Through link classification it is possible to perform TSR.
2. Similarly to token classification, F1 is preferred to evaluate link classification if tackled alone.
3. Links connecting non-tables items should be considered as an additional class '*none*'.

Object Recognition
Even if not required to do CTE, the annotations include area information of different regions in the paper (as common for object detection). Grouping together tokens belonging to the same class via edges can be exploited to find such areas, e.g. extracting sub-graphs from the whole document. A recent paper [18] exploited GNN to perform post-OCR paragraph recognition by grouping together similar items in the pages.

3.1. Limitation of the Task

While we acknowledge that CTE has some limitations, we believe that it represents a significant step towards a more comprehensive solution for table extraction in documents. In our previous work [14], we investigated different ways to achieve CTE through ablation studies, so as to analyze the impact of different components on the system's performances. In this paper, we define a metric, ($F1_{CTE}$), for the updated dataset regarding CTE. As the combination of two metrics, namely Token F1 and Link F1, they can be used to evaluate the performance of the system.

4. Future work

In addition to providing a new dataset for contextualized table extraction, the CTE task can also serve as a basis for future research. One area of research is to investigate the effectiveness of using graph neural networks (GNNs) versus transformer architectures for the CTE task. The models might be pre-trained and fine-tuned on all the original data from [11] and [4]. Comparing a lightweight network, GNN-based, with a heavy network, such as transformer-based, can help determine which approach is best suited for the CTE task. Another potential avenue for future work is to investigate the use of the CTE dataset for information extraction tasks, specifically in the context of scientific papers. Many papers include tables with important information that can be challenging to extract automatically, and incorporating external knowledge bases could

further improve performance. With the CTE dataset, it would be possible to explore how to effectively combine table structure information with external knowledge to answer questions based on scientific papers. Other open research questions that could be addressed using the CTE dataset include investigating cross-lingual performance, transfer learning, and developing techniques to handle different types of tables (e.g., nested tables, tables with merged cells).

5. Conclusions

In this work we presented a new dataset to tackle the task of Contextualized Table Extraction. The dataset is obtained by merging two well-known benchmark datasets (PubTables-1M and PubLayNet). Usually, table extraction pipelines involve several components to perform different tasks on tables, without considering other important information present in the document such as captions. Based on these limitations, the proposed collection of data aims at developing models capable of tackling more tasks at once, resulting in CTE. Moreover, the annotations format encourages the development of systems based on GNN, that lack of a common benchmark within the DAR community for tasks different from TSR. We are looking to extend the dataset by adding more information such as authors, keywords, and equations.

References

[1] S. Marinai, Learning algorithms for document layout analysis, in: C. Rao, V. Govindaraju (Eds.), Handbook of Statistics, volume 31 of *Handbook of Statistics*, Elsevier, ., 2013, pp. 400–419. doi:https://doi.org/10.1016/B978-0-444-53859-8.00016-3.

[2] K. A. Hashmi, M. Liwicki, D. Stricker, M. A. Afzal, M. A. Afzal, M. Z. Afzal, Current status and performance analysis of table recognition in document images with deep neural networks, IEEE Access 9 (2021) 87663–87685. URL: https://doi.org/10.1109/ACCESS.2021.3087865. doi:10.1109/ACCESS.2021.3087865.

[3] M. Kardas, P. Czapla, P. Stenetorp, S. Ruder, S. Riedel, R. Taylor, R. Stojnic, Axcell: Automatic extraction of results from machine learning papers, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 8580–8594. URL: https://doi.org/10.18653/v1/2020.emnlp-main.692. doi:10.18653/v1/2020.emnlp-main.692.

[4] X. Zhong, J. Tang, A. Jimeno-Yepes, Publaynet: Largest dataset ever for document layout analysis, in: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019, IEEE, 2019, pp. 1015–1022. URL: https://doi.org/10.1109/ICDAR.2019.00166. doi:10.1109/ICDAR.2019.00166.

[5] B. Smock, R. Pesala, R. Abraham, PubTables-1M: Towards a universal dataset and metrics for training and evaluating table extraction models, CoRR abs/2110.00061 (2021). URL: https://arxiv.org/abs/2110.00061. arXiv:2110.00061.

[6] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, A. Farhadi, Figureseer: Parsing result-figures in research papers, in: European Conference on Computer Vision (ECCV), 2016.

[7] B. Pfitzmann, C. Auer, M. Dolfi, A. S. Nassar, P. W. J. Staar, Doclaynet: A large human-annotated dataset for document-layout analysis (2022). URL: https://arxiv.org/abs/2206.01062. doi:10.1145/3534678.353904.

[8] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, M. Zhou, Docbank: A benchmark dataset for document layout analysis, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, International Committee on Computational Linguistics, 2020, pp. 949–960. URL: https://doi.org/10.18653/v1/2020.coling-main.82. doi:10.18653/v1/2020.coling-main.82.

[9] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 91–99. URL: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

[10] K. He, G. Gkioxari, P. Dollár, R. B. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, IEEE Computer Society, 2017, pp. 2980–2988. URL: https://doi.org/10.1109/ICCV.2017.322. doi:10.1109/ICCV.2017.322.

[11] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, CoRR abs/1912.13318 (2019). URL: http://arxiv.org/abs/1912.13318. arXiv:1912.13318.

[12] Z. Chi, H. Huang, H. Xu, H. Yu, W. Yin, X. Mao, Complicated table structure recognition, CoRR abs/1908.04729 (2019). URL: http://arxiv.org/abs/1908.04729. arXiv:1908.04729.

[13] A. Gemelli, S. Biswas, E. Civitelli, J. Lladós, S. Marinai, Doc2graph: A task agnostic document understanding framework based on graph neural networks, in: L. Karlinsky, T. Michaeli, K. Nishino (Eds.), Computer Vision – ECCV 2022 Workshops, Springer Nature Switzerland, Cham, 2023, pp. 329–344.

[14] A. Gemelli, E. Vivoli, S. Marinai, Graph neural networks and representation embedding for table extraction in PDF documents, in: 26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022, IEEE, 2022, pp. 1719–1726. URL: https://doi.org/10.1109/ICPR56361.2022.9956590. doi:10.1109/ICPR56361.2022.9956590.

[15] PyMuPDF, J. X. McKie, Pymupdf: Python bindings for mupdf's rendering library., https://github.com/pymupdf/PyMuPDF, 2012.

[16] L. Pisaneschi, A. Gemelli, S. Marinai, Automatic generation of scientific papers for data augmentation in document layout analysis, Pattern Recognition Letters 167 (2023) 38–44. URL: https://www.sciencedirect.com/science/article/pii/S0167865523000247. doi:https://doi.org/10.1016/j.patrec.2023.01.018.

[17] GROBID, Grobid, https://github.com/kermitt2/grobid, 2008–2021. arXiv:1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c.

[18] R. Wang, Y. Fujii, A. C. Popat, Post-ocr paragraph recognition by graph convolutional networks, in: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022, IEEE, 2022, pp. 2533–2542. URL: https://doi.
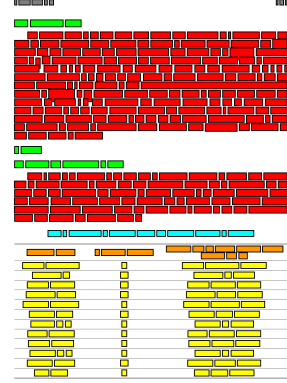
org/10.1109/WACV51458.2022.00259. doi:10.1109/WACV51458.2022.00259.
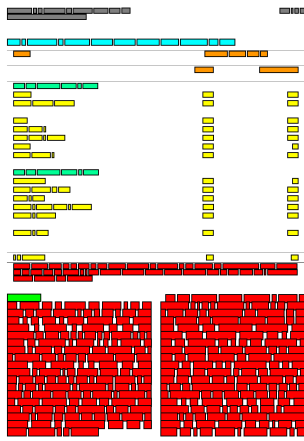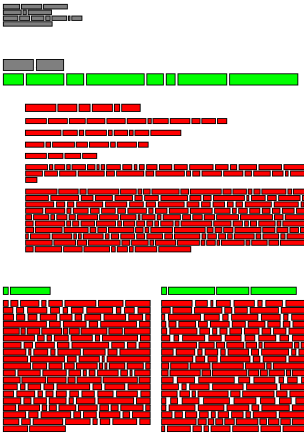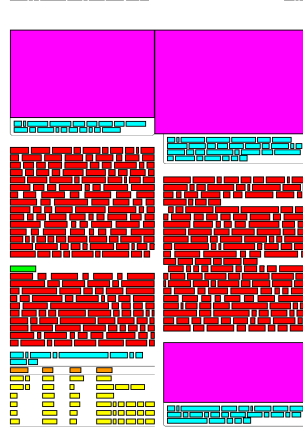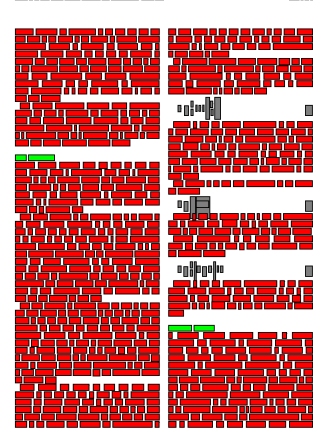
(a) Spanning rows.
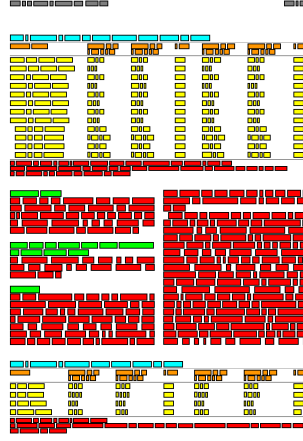
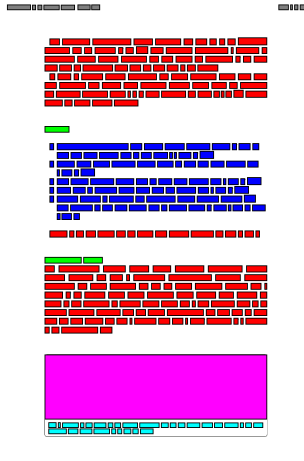(b) More out-column tables.

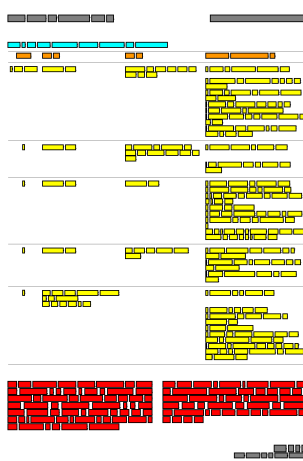(c) Single page layout.

(d) Title page.
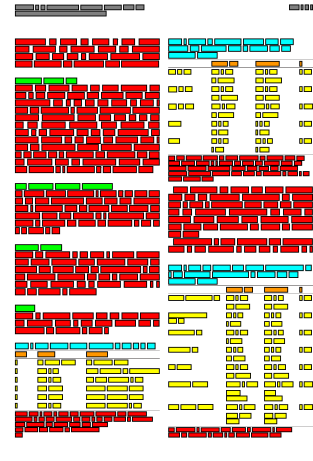
(e) More images per page.

(f) Formulas labeled as others.

(g) List example.

(h) Full page table.

(i) More in-column tables.

Figure 2: Examples of labeled pages showing the different layouts available in the dataset. There can be more tables and images per page (b, i, e) either aligned or not with columns. There are single-column pages (c). As mentioned in Section 2.4, there are some limitations: equations not labeled (f), missing keywords, authors, abstract information (d), and subtitles that are not distinguished from the paper title.