Versatile Layout Understanding via Conjugate Graph

Animesh Prasad
National University of Singapore, Singapore
animesh.prasad@u.nus.edu

Hervé Déjean Jean-Luc Meunier
Naver Labs Europe, Meylan, France
firstname.lastname@naverlabs.com

***Abstract*— Recent advances in document understanding, especially text recognition, provide new opportunities to address the page segmentation problem. In this paper, we propose a method to groups text lines into semantic objects. We model a page as a graph where nodes represent text lines and the edges their geometric relations. The logical segmentation task then refers to identify all text lines belonging to some logical subdivision of the page. We model this task as categorizing edges as relevant or not to build the targeted sub-division (sub-graph). This edge categorization is performed using structured machine learning algorithms (graph Conditional Random Field and Edge Convolutional Network). We use a connected components-based approach following the edge classification for aggregating the nodes. This simple approach shows very robust results for various layout and various page sub-division. We experiment on table segmentation into multiple sub-divisions (rows, columns, and cells) and minutes segmentation into resolutions. Our subdivision and page-layout oblivious approach shows near-par performance as compared to task dedicated approaches and even outperforms them in certain setups.**

***Index Terms*— Document Analysis and Understanding, Page Segmentation, Graph Model, Conjugate Graph, Structured Machine Learning, Graph Conditional Random Field, Edge Convolutional Network, Edge Classification**

I. INTRODUCTION

With the advances in computer vision, spearheaded by deep learning compounded with the increased release of datasets, some traditional issues such as document page segmentation and layout analysis can be now performed with acceptable quality on documents considered very challenging five years ago. Text lines detection (reformulated as baseline detection), for instance, has become possible with very high quality on many archival (primarily handwritten) documents. The latest evaluations such as [1] on the cBAD dataset [2] show that baseline detection performance is now beyond 90% accuracy. To avoid the traditional top-down or bottom-up approaches, some recent work [3], [4] perform multi-task layout analysis by combining several tasks (essentially region, text line segmentation and labeling, along with baseline detection). While competitive and appealing these approaches marginally improve (at best) the individual solutions in terms of quality but may be more convenient at processing time for deployment.

Still, the approaches mentioned above focus on object categorization and do not entirely address the issue of the organization of the page objects targeting a more logical layout analysis. We propose in this article a method which allows organizing page objects into meaningful sub-divisions, depending on the given task (*e.g.*, layout structures, semantic structures). Our idea is to represent the page objects as a graph and to learn the binary relation of two objects (do they belong to the same structure or not). Based on machine learning, this approach applies seamlessly to various structuring tasks as our experiments and evaluations show. Considering that the *defacto* "basic" page elements such as text lines can be "easily" recognized allows us to tackle the (Logical) Layout Analysis problem as a graph categorization problem, and no longer as an image processing problem.

The rest of this article is organized as follows: Section II first covers related work on layout analysis and graph-based approaches. Section III describes the way we formulate the problem, as a graph edge classification, and then Section IV details the machine learning approaches, graph Conditional Random Field and Edge Convolution Networks, used to solve it. Section V presents our different datasets and tasks used for our experiments. Section VI presents the various experimental scenarios along with the corresponding evaluations, showing that this approach overall performs well for all the tasks, even if specialized task-oriented methods may beat it in specific configurations. Finally, we discuss the intrinsic limitations of the technique and propose how to bypass them in the future.

II. RELATED WORK

While page segmentation [5] is revisited frequently in the last few years with the rapid adoption of neural networks, very few articles deal with logical layout analysis. Recently [4] presents a system which can simultaneously perform geometrical and logical analysis by segmenting and labeling in six page-zones. But identifying the relations among labeled elements is still not addressed.

[6] presents a fully convolutional neural network (CNN) approach for newspaper segmentation. They aim to segment a newspaper page into regions corresponding to article blocks. Since article border is very sensitive, they need to preprocess images to help the network learn the border pixel correctly. Non-rectangular regions are filtered out from the training data, and their future work will try to extend to arbitrary shapes. Though it addresses the complex layout segmentation, it does not explore the relations between article blocks.

As one of our primary use cases cover table understanding, we recount the recent work on this topic. A follow-up work of [7] proposes DeCNT [8], an approach based on a novel combination of deformable CNN with Faster R-CNN for table detection. Both papers present a comprehensive literature review and comparisons. [7] uses Faster R-CNN for the table recognition task, and also offers a complementary

system for the table understanding task (row and column detection) using Fully Connected Networks. It is interesting to note that the authors fail to use Faster R-CNN as such for segmentation. *High number of rows, columns, and very close proximity are the two factors that make segmentation task so tricky for FRCNN and demand for a different approach.* Their alternative method uses a fully-connected network, along with pre-processing consisting in stretching the images horizontally and vertically to improve the row and column identification (somehow similar to preprocessing used in [6]). The paper also argues that the image-based solution (converting any document, especially PDF into image format) allows for generic and robust solutions. Our approach shows that under reliable performance of upstream text line identification graph-based models are equivalently generic and robust.

Others [3], [9], [10] merge both representations, image and text (provided by OCR or PDF) to perform logical labeling or information extraction (for invoices). It is interesting to note that [9] also uses FRCNN to detect multiple line-items regions, but their task-oriented evaluation measure makes the comparison with [7] difficult (and most of the invoices in their dataset have only less than three items).

So, how to structure a page? By structuring, we mean establishing relations between objects of a target sub-division. The reviewed work implicitly performed some structuring by grouping objects (pixels) in the same structure, which can be defined by a rectangular zone, but we would like to go beyond this limitation by being able to group objects into arbitrary structures.

For analyzing complex layouts, literature provides some more traditional approaches by assuming first the detection of elementary objects (such as text lines), and then organize them primarily using some graph-based representation. It allows the computational representation of arbitrary structures owing to its greater expressiveness. The related problem of reading order computation can also capitalize on graph representations. [11] investigates the problem of detecting the reading order relationship between components of a logical structure using first-order logic theory. More recently, [12] and [13] formulate the reading order detection with a bipartite graph where reading transitions (edges) scores with a set of features encoding geometrical and textual information along with Hungarian algorithm for optimal matching.

We were not able to find more recent work tackling reading order problem or more generally detecting relationships between layout/ logical page objects (the topic was never intensively studied by the community).

[14] presents work very close to ours, but working at the image level. They represent a page as a graph where nodes correspond to (labeled) connected components and edges *link nodes that share a common edge in the area Voronoi diagram [sic]*. The segmentation problem is then formulated as an edge removal problem similarly to us. This classification is done using a Multilayer Perceptron resulting in subgraphs corresponding to layout objects (typically paragraphs). Note that they do not take into account the adjacencies of the graph and hence the expressive power of the graph representation, which as we show in our results is a critical advantage of our approach.

This brings us to the central hypothesis and contribution. We present in this paper a method which represents a page as a graph, the nodes being textual elements and where edges represent geometric relations between two nodes (see Fig. 1). We classify edges as relevant or not for a target layout structure. This classification is efficiently performed using either a graph Conditional Random Fields classifier or Edge Convolutional Networks introduced recently by [15]. Our in-depth study uses various open-data datasets, and we open-source [1] the methods and the evaluation algorithm.

III. PROBLEM FORMULATION

We present now a more formal and detailed description of various aspects of the problem.

A. Input

As already mentioned, our input data is not a page image, but the output of a text line detection algorithm. We use the one presented in [1] which is a state-of-the-art system achieving one of the best results in the literature. Polygon is used to represent a text line (we use PAGE [16] as input format). Note that for evaluation purpose, we use the text lines from the ground truth, which allows us to perform a fine-grained evaluation. Nevertheless, the text line detection setting is used in production successfully for the final Information Extraction task.

B. Problem Statement

Given a page containing text lines, we want to **partition** its set of text lines, so that each partition maps to one relevant **sub-division** of the page. Depending on the use-case, the sub-divisions can be a row, column, cell, resolution (in this paper), paragraph, news article or anything that has some geometrical relatedness on the page layout. We do not consider the case of one item that would span over multiple consecutive pages. If the scanned image contains a double-page, we consider it as one page (in other words, one image is a page and by extension one graph). Section IV-A describe how we build the graph and we come back and further discuss this limitation in Section VII.

C. Evaluation Metric

Since we produce a partition of the page objects, we need to evaluate a partition given a reference partition of objects. To match a predicted partition to a reference one, we use the Intersection over Union (IoU or Jaccard Index) similarity measure for two sub-divisions (one represented by candidate partition and other represented by reference partition), and a threshold to determine whether there is a match or not. We use two indicative thresholds for this evaluation – 100% which evaluates perfect match, and 80%.

[1] code available at `https://github.com/Transkribus/TranskribusDU`

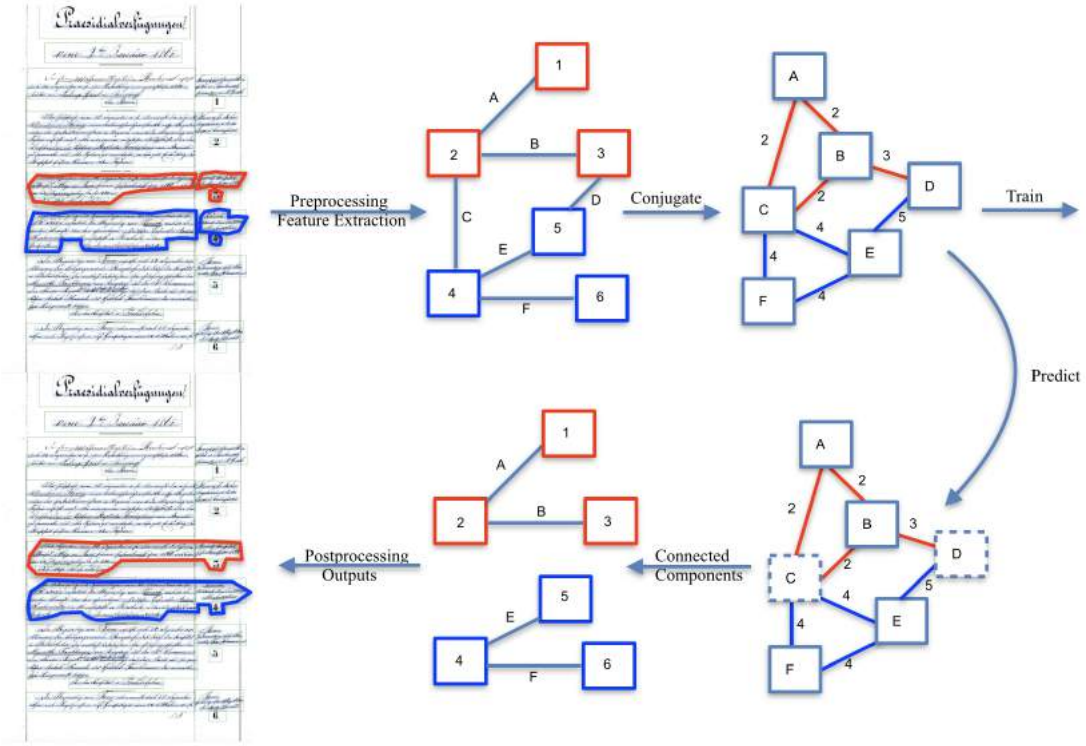


Fig. 1. Pipeline illustrating various steps on a graph made of text lines. Note that our method makes one graph for all the text lines per only a small page fragment is shown for illustration purpose.

Precision and Recall are computed, and we report their harmonic mean (F_1) as evaluation metrics.

IV. METHOD

Following are the different steps (see Fig. 1) of the proposed technique:

A. Building a graph for the page

We model each page as a graph, where each node reflects one text line. An edge in the graph reflects a neighboring relationship between two text lines, possibly long distance ones. More precisely, whenever there is horizontal (respectively vertical), significant and direct overlap between two bounding boxes of two text lines, we create a vertical (respectively horizontal) edge. *Significant* means that the overlap must be higher than a certain threshold. *Direct* means that the two bounding boxes must be in the line of sight of each other,*i.e.* without any obstructing block in-between.

B. Converting to Conjugate Graph

We create the conjugate graph (also called Line Graph), where nodes reflect the original edges. For graph G, its conjugate $L(G)$ is a graph such that each vertex of $L(G)$ represents an edge of G; and two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G. This results in a new graph with the features of the nodes as that of the edges and vice versa. The transformation is asymmetric in terms of graph topology and the features. It implies the new conjugate graph node receives features from both the original graph nodes at its endpoints (as concatenation) whereas only those nodes are transformed to edges which connect two edges in the original graph.

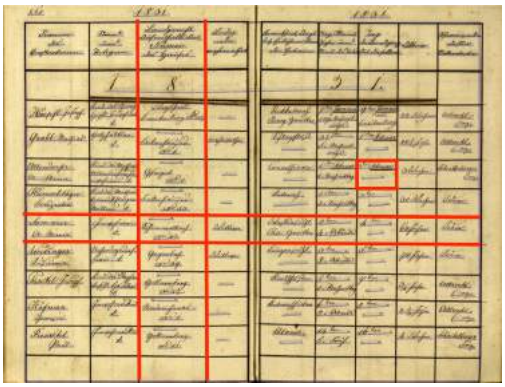
C. Training

During the training phase, we use the conjugate graph and train the classifier on this graph to learn the parameters to classify each node into binary classes. The label of a node (corresponding to an edge in the original graph) in the conjugate graph is *one* if it connects two text lines belonging to the same sub-division in the original graph, *zero* otherwise. This means for a particular page, the text line based graph construction will result in the same conjugate graph, but depending on the labels (which edges induce relation versus which do not) the model parameters can learn different sub-divisions.

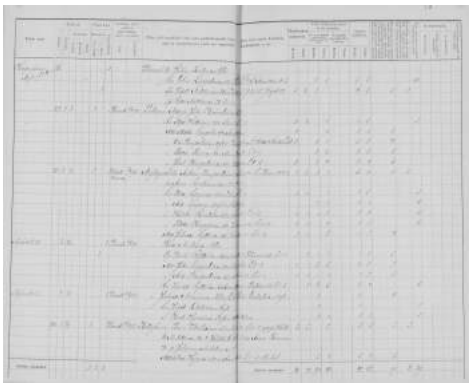
We use two structured classifiers, a graph Conditional Random Field (gCRF) and an Edge Convolution Network (ECN) which is the edge feature augmented modification of Graph Convolution Network (GCN). We refer the reader to [15] for a more detailed presentation of both methods, and we only sketch them rapidly since we replicate them as described in the original paper.

The graph CRF model uses the PyStruct Open Source Python library [17]. We trained using the one-slack structured SVM method and ran inferences using AD3. We train for 1000 iterations using the default hyper-parameter values.

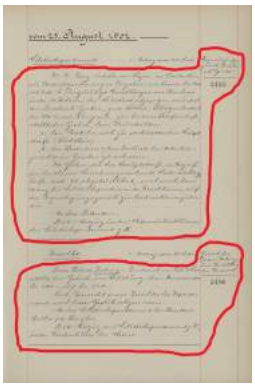
A GCN, in essence, is composed of two steps. First, compute some node representation by applying a transformation on the current feature representation; let us call



(a) A page from ABP.



(b) A page from NAF, showing sparse tax records and small cells.



(c) A page from BAR showing two resolutions.

Fig. 2. Exemplary pages from the datasets used in the experiments.

this representation a potential in a loose sense. Then, this node potential is convolved; it merely means that for each node, one takes a weighted average of the neighbor nodes' potential. Finally, this average is fed to the next layer of the neural network.

Let A be the adjacency matrix of the undirected graph G and D its degree matrix. The layer-wise propagation is defined as:

$$H^{(l+1)} = f\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \quad (1)$$

Here, $\tilde{A} = A + I_N$ is the self loop augmented adjacency matrix and $W^{(l)}$ is the layer-specific set of weight parameters. f denotes an activation function; in the GCN model, *ReLU* is preferred. $H^{(l)}$ is the matrix of activation in the l^{th} layer such that $H^{(0)} = X$, $X \in \mathbb{R}^{n,a}$.

The main idea of ECN is to learn graph convolutions which depend on edge features. If we can assign a score to each edge in the graph, we, therefore, have defined a parametrized adjacency matrix. In this way, the network can learn to filter out some edges or to find new scales to average neighboring nodes. Consider the source edge matrix S and target edge matrix T. S_{ij} is 1 if edge j has node i as source (respectively destination for T_{ij}) such that $\tilde{A} = ST^t$ Consider F is the feature matrix for the edges in the graph and w_E a parameter vector for defining a convolution on edges, a way to represent parametrized adjacency is $g(w_E) = Sdiag(w_E F)T^t$ where *diag* is the diagonal matrix. Therefore, the redefined layer-wise propagation for ECN is:

$$H^{(l+1)} = f\left(g(w_E)H^{(l)}W^{(l)}\right) \quad (2)$$

D. Partitioning

After prediction, we remove the edges in the original graph corresponding to the *zero* predicted nodes in the conjugate. The maximum connected components are the desired logical segmentation. This simple and naive method produces competitive results but is extremely sensitive to edge classification errors as the reader can imagine (one single error can lead to merging two parts). One of the tested methods, ECN, provides probabilities for both classes (the probability to be kept and to be removed). We investigated whether it is possible to find a better performance than taking the decision based on raw probability scores. These raw scores translate to a remove probability of above 0.5, which remains the best choice in our experiments (although more conservative remove probability like above 0.4 may improve results for some datasets/tasks).

V. Datasets and Use cases

We now detail the different datasets and tasks we used for evaluation. A common aspect of the different datasets is that they are all handwritten, making them fairly challenging for the tasks we selected.

A. Minute – BAR

The Schweizerisches Bundesarchiv (BAR) – Digitalisierte Bundesratsprotokolle [2] is the Digitized Federal Council minutes from the Swiss Federal Archives. The collection spans the years 1848 to 1882 (300,000 handwritten pages). The task for this collection is to segment it into resolutions. A resolution is composed of a number and optionally a summary (both occurring in the margin) and a body composed of one or more paragraphs (see Fig. 2(c)).

B. Table - ABP and NAF

We also tested our approach with a challenging problem, mainly table understanding. The task is to identify table rows, columns, and cells. Datasets are rare for this task, and we use one public dataset, the ABP [18], (say ABP_small), and two new datasets – an extension dataset of the ABP [3] (containing not only death records but birth and marriage records; say ABP_large), and another from National Archive Finland (NAF) [4], corresponding to tax records. A specificity of NAF dataset is its very narrow columns containing digits and its sparsity *i.e.* most cells are empty, while for ABP

[2] https://www.infoclio.ch/de/node/133651

[3] https://github.com/Transkribus/TranskribusDU/tree/master/usecases/: folder ABP and NAF

[4] https://www.arkisto.fi/en/frontpage

datasets, most of the cells are filled in. Table I describes the different datasets in terms of pages, text lines, rows, columns, and cells.

TABLE I
DATASET DESCRIPTION

Dataset	# pages	# rows	# columns	# cells
ABP_small	180	528	457	
ABP_large	1098	1551	1275	15074
NAF	488	3203	3159	33831
Dataset	**# pages**	**# text lines**	**# resolutions**	**# res. per page**
BAR	107	3386	77	1.4

VI. EXPERIMENTS

On ABP_small, ABP_large, NAF and BAR datasets, we create *four*, *nine*, *four*, and *three* folds respectively. For all datasets, we test on the *first* fold and use all the remaining folds for training and validation. Experimental evaluations are given Tables II, III, IV for the table datasets and Table V for the minute dataset.

Besides the F_1 measure for the edge classification *per se*, we report the metric discussed in Section III-C at 100% and 80% thresholds (one predicted sub-division matches a ground-truth sub-division if its IoU ratio is above the threshold value). Note that the 100% threshold is very strict as it requires an exact match between the reference partition and the produced one. Some work cited in Section II used a threshold of simply 50% for table detection (comparing geometrical regions by using IoU). In addition to the performance of the present proposal with gCRF and ECN (with different number of layers), we report:

- "Oracle", which uses the ground-truth to label the edges and then apply the maximum connected component method to partition. This is somehow an evaluation of the method we use to build the partitions from the edge labels and gives us upper bound of the proposed conjugate graph method equivalent to when gCRF or ECN perform at 100% accuracy.
- a task-oriented method if available (different for each task).
- baseline linear classifiers (LR, SVM) only using the edges features (without any neighborhood context). We mention them only for ABP_small and BAR (result for edge classification are around 80% F_1, producing degenerate values for the final decision).

We must also describe how spanning elements are processed since they introduce noise in the evaluation which currently does not support this phenomenon. When a text line spans several rows (respectively columns), we arbitrarily consider it to be on the first of the rows (respectively columns) it spans. So the row (respectively column) spanning is not dealt with ideally. This is especially true for the NAF collection where the table header part is highly structured with up to 4 rows with spanned elements. The ABP collections have almost no row-spanned texts and a few column-spanned texts.

ECN is the most robust method in general, while gCRF slightly outperforms it for some datasets/tasks (row segmentation for both ABP datasets and BAR), but may really fail for others (row and column segmentation for NAF, cell segmentation for ABP_large). We experiment with 1 and 8 layers for ECN. In ECN formalism a model with n layers aggregate features from n-hop neighborhood. An ECN architecture with only 1 layer under-performs compared to an 8 layers architecture, which can be interpreted as a need for larger context than the immediate neighborhood. Looking at the second (edge classification) and third columns (perfect match at structure level) of Tables II–V, we see that the second measure acts as a magnifying effect *i.e.* a small improvement of the first measure may significantly increase the performance on the second. We also show the result for SVM and LR which do not aggregate features from any node and is equivalent to zero neighborhood context, similar to [14].

We analyze here the two types of errors we face in our approach and comment on their impact with regard to the target structure:

- If an edge is wrongly predicted positive (False Positive), we wrongly merge two structures. This situation leads to a structure which is often incoherent: this merge leads to consider edges which were categorized as negative (not connected) as positive ones (since the elements belong to the same structure). As future work, we will investigate whether this incoherence may be easily detected and the False Positives automatically detected.
- If an edge is wrongly predicted negative (False negative), we may split a structure if and only if no other edge keeps the connectivity. If the graph of a structure is mostly connected, there is redundancy that protects against that particular error (as noticed by [14]). This behavior is particularly true for the *row* structure, but less relevant for the *column* one whose graphs are often similar to a minimal spanning tree (or even a sequence) (see Fig. 2(a) and Fig. 2(b)).

A. Table Segmentation

On the table datasets, we can partition text lines into rows, columns, or cells. We report on all the above sub-divisions. We exclude from the input any text that is outside tables.

1) Rows Segmentation: As the task-oriented method for rows, we use a strong method [19], which learns to categorize horizontal separators (skewed lines) in order to partition the tables into rows (candidate separators are first generated, then a learning algorithm is used in order to classify).

The proposed method shows decent results compared to this task-oriented method, which currently is our best method, but a highly specialized one. (It has a strong knowledge of the shape of the parts it is looking for, hence its lack of generality). We note that the Oracle is reasonable, around 90% (most of the errors are due to spanned cells). The ECN and gCRF methods are close to the Oracle results, which gives hope that a more accurate graph construction or a better method to use the edge labels can significantly

TABLE II

SEGMENTATION INTO **ROWS**

Dataset	F_1 (edges)	F_1@100%	F_1@80%	Method	Comment
ABP_small	99.7	87	92	ECN	8 layers
	99.2	78	85	ECN	1 layer
	99.0	83	91	gCRF	
	100	93	98	Oracle	
	-	91	97	Task-oriented	[19]
	81.5	0	0	LR	
	78.9	0	0	SVM	
ABP_large	98.2	77	81	ECN	8 layers
	96.4	64	69	ECN	1 layer
	95.4	74	79	gCRF	
	100	90	91	Oracle	
	-	78	86	Task-oriented	[19]
NAF	98.6	69	78	ECN	8 layers
	95.8	51	64	ECN	1 layer
	95.1	48	65	gCRF	
	100	82	87	Oracle	
	-	72	79	Task-oriented	[19]

improve the results of them. Some errors, typically a row split into two parts, could be fixed with by introducing some knowledge about the expected sub-division (see Section VII).

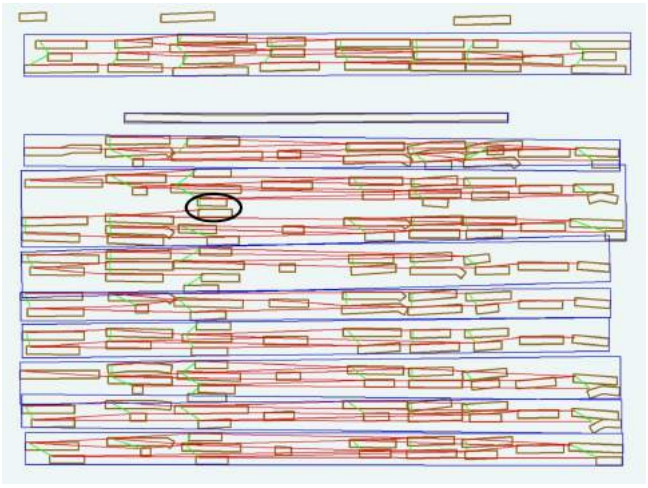
2) Column Segmentation: As the task-oriented method for column segmentation, we use an unsupervised projection profile method, since it is currently one of the state-of-the-art choices. In brief, we shrink the text lines by 33% and then look for vertical cuts, which do not cut too much through texts. This is an old good method which works well on the ABP datasets, poorly on the challenging NAF one (as explained, columns are already thin and skewed).

TABLE III

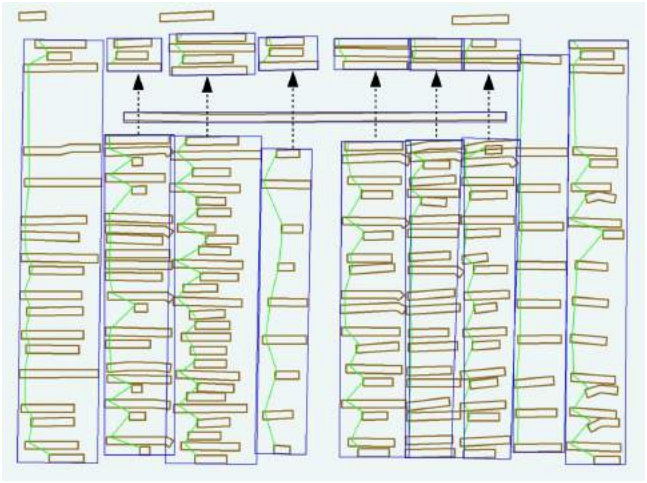
SEGMENTATION INTO **COLUMNS**

Dataset	F_1 (edges)	F_1@100%	F_1@80%	Method	Comment
ABP_small	99.8	93	95	ECN	8 layers
	99.3	90	91	ECN	1 layer
	99.2	87	89	gCRF	
	100	94	96	Oracle	
	-	93	96	Task-oriented	Unsupervised Projection Profile
ABP_large	98.6	76	80	ECN	8 layers
	98.8	76	79	ECN	1 layer
	98.7	70	75	gCRF	
	100	85	85	Oracle	
	-	88	94	Task-oriented	Unsupervised Projection Profile
NAF	99.0	72	80	ECN	8 layers
	97.5	52	64	ECN	1 layer
	96.8	43	61	gCRF	
	100	82	87	Oracle	
		27	41	Task-oriented	Unsupervised Projection Profile

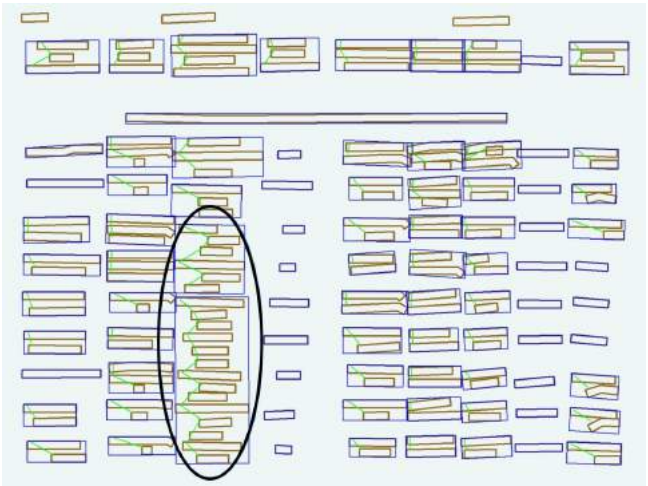
The main difference with the row segmentation is the poor score achieved by the Oracle. Fig. 3(b) illustrates this main issue. In the shown table a long text line breaks a set of columns (the text line is spanned over several columns). We will comment on this issue in Section VII. On NAF, where columns content is skewed and short (digits), we observe



(a) Row segmentation. Only a single edge is wrongly classified (encircled in black), culminating in merging of two rows.



(b) Column segmentation. A typical error when the initial graph is not able to produce the expected structure. A long horizontal text line prevents columns elements to be linked. Note that the long horizontal line is considered as well as a column.



(c) Cell segmentation. Errors causing merge of multiple cells is encircled in black.

Fig. 3. Segmentation on the page displayed in Fig. 2(a). Resulting structures are in blue, vertical edges in green, horizontal edges in red. The top 3 elements are outside of the table and thus not considered.

that ECN largely outperforms our task-oriented method (the implemented projection profile method does not support skewed content). We currently do not understand the poor score obtained by gCRF for this task.

3) Cell Segmentation: As task-oriented method for the partitioning in cells, we simply combine the output of the row task-oriented method and column task-oriented method output *i.e.* the intersection of rows and columns produces a partition in cells.

TABLE IV
SEGMENTATION INTO **CELLS**

Dataset	F_1 (edges)	F_1@100%	F_1@80%	**Method**	**Comment**
ABP_small	99.1	97	97	ECN	8 layers
	98.9	94	94	ECN	1 layer
	98.8	93	93	gCRF	
	100	99	99	Oracle	
	-	95	95	Task-oriented	Intersection of rows and columns
ABP_large	99.1	96	96	ECN	8 layers
	97.2	89	89	ECN	1 layer
	96.6	86	86	gCRF	
	100	99	99	Oracle	
	-	91	91	Task-oriented	Intersection of rows and columns
NAF	99.5	97	97	ECN	8 layers
	98.5	94	94	ECN	1 layer
	96.9	93	93	gCRF	
	100	99	99	Oracle	
	-	56	56	Task-oriented	Intersection of rows and columns

The results are particularly promising since the partition into cells is very good. Compared to the column and row approach, we could deal properly with cell spanning multiple rows or columns. This calls for a bottom-up method to reconstruct the table from its cells, and tolerating some small level of noise.

B. Minute Segmentation

Minute segmentation is a good example of on-demand sub-division segmentation where (unlike fixed known partitions – rows, columns, cells) the partitions are collection defined and purpose-specific making designing task-oriented methods not only costly and time-consuming but also requiring expert geometrical analysis. On the minutes dataset (BAR), we currently have no other method to compare with. As reported in Table V, results are very good given rather homogeneous nature of the dataset. Nevertheless, this shows that the method can cover different structures in the same way. Furthermore, this dataset and task are interesting since they correspond to documents and structures (marginalia) which are extremely frequent in archival institutions (minutes).

TABLE V
SEGMENTATION INTO **RESOLUTIONS** (BAR)

F_1 (edges)	F_1@100%	F_1@80%	**Method**	**Comment**
97.6	85	88	ECN	8 layers
97.4	84	86	ECN	1 layer
97.3	85	92	gCRF	
100	100	100	Oracle	
	-	-	Task-oriented	No task-oriented alternative method!
87.2	5	13	LR	
86.9	5	13	SVM	

VII. CONCLUSIONS

In this work, we present a surprisingly simple and yet versatile approach to document logical segmentation. We convert the document component graph to its edge-to-vertex conjugate to facilitate learning the relationship between the document elements rather than prior component labeling approaches. Edges categorization is performed using structured machine learning classifiers which utilize the greater expressive power of graph representations. While not always achieving the best results, it allows for competitive results for all tasks, where dedicated methods were used before. We see currently two ways to improve it:

- Improving the way the graph is built as in some situation (columns) the current version does no cover correctly all the structures. The first try will be to test a fully connected graph (similarly to what is done in image processing as in [20]).
- Improving upon the naive connected component approach to partition text lines. For this, a very high accuracy level is required, which is achieved in our various datasets. Could this accuracy level be maintained for a more heterogeneous dataset? And the follow-up question – could we train a generic model for various structures (mixed layouts and structures)?

One interesting parallel can be drawn between our method and pixel-level categorization methods – after the pixel categorization (usually performed with high accuracy), a post-processing step (often based on connected components computation) is required in order to generate the targeted layout element (baseline, blocks, and so on). That post-processing is highly task-oriented. For efficiency purpose, we may end up following this strategy and design various post-processing depending on the tasks.

We will also assess the method in a near future to see how it can be applied to various elements of the *hierarchical* layout structure (text lines to paragraph, paragraph to articles among some) and how to exploit the current framework to jointly learn the reading order by some multi-tasking approach.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 674943 (READ project).

REFERENCES

[1] T. Grüning, G. Leifert, T. Strauß, and R. Labahn, "A two-stage method for text line detection in historical documents," *CoRR*, vol. abs/1802.03345, 2018.

[2] M. Diem, F. Kleber, S. Fiel, T. Grüning, and B. Gatos, "cBAD: ICDAR2017 competition on baseline detection," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1355–1360, Nov. 2017. DOI: 10.1109/ICDAR.2017.222.

[3] Y. Xu, F. Yin, Z. Zhang, and C.-L. Liu, "Multi-task layout analysis for historical handwritten documents using fully convolutional networks," in *27th International Joint Conference on Artificial Intelligence (IJCAI)*, AAAI Press, Stockholm, Sweden, Jul. 2018, pp. 1057–1063. DOI: 10.24963/ijcai.2018/147.

[4] L. Quirós, "Multi-task handwritten document layout analysis," *CoRR*, vol. abs/1806.08852, 2018.

[5] D. Doermann and K. Tombre, Eds., *Handbook of document image processing and recognition*. London: Springer, 2014, ISBN: 978-0-85729-858-4. DOI: 10.1007/978-0-85729-859-1.

[6] B. B. Meier, T. Stadelmann, J. Stampfli, M. Arnold, and M. Cieliebak, "Fully convolutional neural networks for newspaper article segmentation," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan: IEEE, Nov. 2017, pp. 414–419. DOI: 10.1109/ICDAR.2017.75.

[7] 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 *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, Nov. 2017, pp. 1162–1167. DOI: 10.1109/ICDAR.2017.192.

[8] S. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "DeCNT: deep deformable CNN for table detection," *IEEE Access*, vol. 6, pp. 74 151–74 161, 2018. DOI: 10.1109/ACCESS.2018.2880211.

[9] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, "Chargrid: Towards understanding 2D documents," in *2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, 2018, pp. 4459–4469.

[10] R. Palm, "End-to-end information extraction from business documents," English, PhD thesis, 2018.

[11] D. Malerba, M. Ceci, and M. Berardi, "Machine learning for reading order detection in document image understanding," in *Machine Learning in Document Analysis and Recognition*, ser. Studies in Computational Intelligence, vol. 90, Springer, 2008, pp. 45–69, ISBN: 978-3-540-76279-9. DOI: 10.1007/978-3-540-76280-5_3.

[12] J. Fang, Z. Tang, and L. Gao, "Reflowing-driven paragraph recognition for electronic books in PDF," in *Document Recognition and Retrieval XVIII*, ser. SPIE Proceedings, vol. 7874, International Society for Optics and Photonics, 2011, 78740U.

[13] L. Gao, Y. Wang, Z. Tang, and X. Lin, "Newspaper article reconstruction using ant colony optimization and bipartite graph," *Applied Soft Computing*, vol. 13, no. 6, pp. 3033–3046, 2013. DOI: 10.1016/j.asoc.2012.07.012.

[14] A. L. L. M. Maia, F. D. Julca-Aguilar, and N. S. T. Hirata, "A machine learning approach for graph-based page segmentation," in *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, IEEE, 2018, pp. 424–431. DOI: 10.1109/SIBGRAPI.2018.00061.

[15] S. Clinchant, H. Déjean, J.-L. Meunier, E. M. Lang, and F. Kleber, "Comparing machine learning approaches for table recognition in historical register books," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, IEEE, Vienna, Austria, Apr. 2018, pp. 133–138. DOI: 10.1109/DAS.2018.44.

[16] S. Pletschacher and A. Antonacopoulos, "The PAGE (page analysis and ground-truth elements) format framework," in *2010 20th International Conference on Pattern Recognition (ICPR)*, IEEE, Istanbul, Turkey, Aug. 2010, pp. 257–260. DOI: 10.1109/ICPR.2010.72.

[17] A. C. Müller and S. Behnke, "PyStruct: Learning structured prediction in python," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2055–2060, 2014, ISSN: 1532-4435.

[18] H. Déjean, E. Lang, and F. Kleber, *READ ABP table datasets*, Apr. 2018. DOI: 10.5281/zenodo.1243098.

[19] J.-L. Meunier and H. Déjean, "Table rows segmentation," in *2019 15th IAPR International Conference on International Conference on Document Analysis and Recognition (ICDAR)*, to appear, Sydney, Australia, Sep. 2019.

[20] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr, "Higher order conditional random fields in deep neural networks," in *2016 14th European Conference on Computer Vision ECCV Part II*, Amsterdam, The Netherlands, Oct. 2016, pp. 524–540. DOI: 10.1007/978-3-319-46475-6_33.