Logical Layout Analysis Applied to Historical Newspapers

Nicolas Gutehrlé, Iana Atanassova

Logical Layout Analysis Applied to Historical Newspapers

Nicolas Gutehrlé
Centre de Recherches Interdisciplinaires
et Transculturelles (CRIT),
Université de Bourgogne Franche-Comté
30 rue Mégevand, 25000 Besançon, France
nicolas.gutehrle@univ-fcomte.fr

Iana Atanassova
Centre de Recherches Interdisciplinaires
et Transculturelles (CRIT),
Université de Bourgogne Franche-Comté
Institut Universitaire de France (IUF)
30 rue Mégevand, 25000 Besançon, France
iana.atanassova@univ-fcomte.fr

Abstract

In recent years, libraries and archives led important digitisation campaigns that opened the access to vast collections of historical documents. While such documents are often available as XML ALTO documents, they lack information about their logical structure. In this paper, we address the problem of logical layout analysis applied to historical documents. We propose a method which is based on the study of a dataset in order to identify rules that assign logical labels to both block and lines of text from XML ALTO documents. Our dataset contains newspapers in French, published in the first half of the 20th century. The evaluation shows that our methodology performs well for the identification of first lines of paragraphs and text lines, with F1 above 0.9. The identification of titles obtains an F1 of 0.64. This method can be applied to preprocess XML ALTO documents in preparation for downstream tasks, and also to annotate large-scale datasets to train machine learning and deep learning algorithms.

1 Introduction

One important challenge in digital humanities is the efficient exploitation and processing of scanned textual documents (archives, documentary funds, ...). For example, historical documents such as newspaper archives are prime resources for historians (Tibbo, 2007). Thanks to the important digitisation campaigns led by libraries and archives, vast collections of historical documents have been made easily accessible. However, the majority of these documents are available only as scanned images (e.g. in PDF format) which makes them difficult to explore in a text processing perspective. Extracting the text content from such documents requires at least the following three steps: Optical Character Recognition (OCR), physical layout analysis (PLA) and logical layout analysis (LLA).

Physical layout analysis (PLA), which is also sometimes called *document layout analysis*, consists in identifying physical regions of the document, with their text content and boundaries. Such regions can correspond to sections and lines of text, but also to figures, tables, etc. PLA also defines the reading order of the document, which corresponds to the linear order in which the different regions appear. This is particularly important for documents that have multi-column layouts. One commonly used output format of PLA is the XML ALTO format[1]. *Logical layout analysis (LLA)*, sometimes called *logical structure derivation* and *structure understanding*, consists in identifying the document structure elements and their categories i.e. title, header, paragraph, table, etc. Such logical elements can integrate one or more regions in the document that have been identified by PLA.

Physical and logical layout analyses are necessary steps in the processing of documents for a large number of applications, including Information Retrieval, information extraction, Table of Content extraction, text syntheses, and more broadly document understanding.

In this article we focus on the problem of logical layout analysis (LLA). We describe a methodology for logical layout analysis, where logical labels are assigned to physical layout entities. The input of our processing pipeline is the physical layout analysis of documents in the XML ALTO format.

The rest of the article is organised as follows: the following section presents the related work on logical layout analysis. Section 3 present our train and test datasets. Section 4 presents the methodology that we propose and section 5 proposes an evaluation of the implemented processing pipeline. Finally, we propose a conclusion and a discussion.

[1] ALTO: Technical metadata for layout and text objects: https://www.loc.gov/standards/alto/

Proceedings of the Workshop on Natural Language Processing for Digital Humanities (NLP4DH), pages 85–94
co-located with ICON-2021, December 16-19, 2021, Silchar, India.
https://doi.org/10.26615/978-952-94-5833-2_010

2 Related works

An important body of research around physical layout analysis of printed documents has been produced in the end of the XXth century. Several algorithms have been proposed such as the X-Y Cut algorithm (Nagy et al., 1992), the Docstrum algorithm (O'Gorman, 1993) or the Voronoi diagram based algorithm (Kise et al., 1999). Furthermore, the processing of handwritten documents requires specific techniques, such as the "droplet" technique to identify text line by Bulacu et al. (2007), or neural networks as in Chen and Seuret (2017), where each pixel is labelled as text or not.

Existing logical layout analysis systems make use of various methods that go from heuristic systems to more recent architectures using neural networks. Some heuristic systems use grammars such as stochastic or attributed grammars, where the document is represented as a string of symbols, e.g. Namboodiri and Jain (2007). In their work, the grammar describes multiple production rules, each associated with a logical label. The string of symbols is then parsed by the grammar in order to extract logical labels. Other systems, such as LA-PDFText (Ramakrishnan et al., 2012) or DeLoS (Niyogi and Srihari, 1995), use rules that state the condition a physical block must meet to be given a logical label. For instance, DeLoS system uses first-order predicates in order to infer the logical category of a physical block.

While heuristic systems provide good results, they are often dedicated to specific layouts, and need to be adapted to work on other layouts. To tackle this problem, Klampfl and Kern (2013) created a system for logical layout analysis on scientific articles in PDF format that combines heuristic rules with unsupervised-learning models such as k-means or Hierarchical Agglomerative Clustering (HAC). This system is made up of several detectors, each learning geometrical and textual features from the document in order to identify a specific logical label. Some rules using text occurrences are also used to help the model, such as finding the keywords "Table" or "Fig." to identify table or figure blocks.

More recent works use neural networks for logical layout analysis. As noted by Akl et al. (2019), CNN or LSTM architectures work better than classical neural networks because of the sequential nature of the documents. This task also benefits from the use of word-embeddings such as fastText, Flair or GloVe which give a better encoding of textual data than simple one-hot encodings, as in Zulfiqar et al. (2019). Neural network systems can be trained on big datasets such as the Publaynet dataset (Zhong et al., 2019) or the Medical Articles Record Groundtruth (MARG) for physical and logical layout analysis purposes.

Considering the task of processing historical documents, several small datasets exist such as the DIVA-HISDB dataset (Simistira et al., 2016) which contains 150 annotated pages of three different medieval manuscripts or the European Newspapers Project Dataset (Clausner et al., 2015) which contains 528 documents. Other datasets in non-European languages exist, such as the PHIBD dataset (Hossein Ziaie Nafchi and Cheriet, 2012), which contains images of 15 Persian historical and old manuscript, and the HJDataset by Shen et al. (2020), which contains 2271 Japanese newspapers published in 1953, which was generated in a semi-automatic way. All of these datasets are too small to be used for machine learning or neural network approaches.

Hébert et al. (2014) deal with the task of article segmentation by a Conditional Random Field (CRF) model with heuristic rules to perform logical analysis. First the CRF model labels pixel as titles, text lines, or horizontal and vertical separators, then heuristics rules describing usual article layouts are applied to that classification. In both cases, bad results were caused by the quality of the scan or the quality of the OCR output. On the other hand, Riedl et al. (2019) deal with article segmentation by looking at the similarity between segments of texts. These segments are computed either by using the Jaccard coefficient and their word distribution or by computing the cosine similarity between word-embeddings. The similarity between blocks is then computed using the TextTiling algorithm (Hearst, 1997)

Most common approaches to LLA are not suited for historical documents because the document layout changes over time. For example, the layout and structure of an advertisement in the same newspaper can display important changes over several years. Logical layout analysis systems applied to historical documents must then account for the diachronic aspect of their layouts and adapt to the changes. Barman et al. (2020) propose a system that goes beyond usual logical labels by labelling physical block as either Serial, Weather Forecast,

Death Notice and Stock Exchange Table. To do so, their system combines visual and textual features using the word-embedding representation of each word and its coordinates on the page. Their results show that combining textual and visual features provide better results in most cases than using just one of them. Textual features are also more efficient to deal with the diachronic aspect of documents because they are more stable over time than visual features.

3 Dataset

We have processed a dataset of press and magazine documents published in the first half of the 20th century from the "Fond régional: Franche Comté" collection, available from the digital archive of Bibliothèque Nationale de France[2]. Figure 1 shows an example of the first page of a newspaper with more than 2 columns. It contains the header of the first page and several articles that contain titles and text content.

From this collection, we selected documents that had an OCR quality measure greater than 90%. This dataset was then split into a train and a test dataset. As shown in Table 2, our train set contains 15 collections of documents, which amount to a total of 48 documents, whereas the test set contains 6 collections and a total of 6 documents (Gutehrlé and Atanassova, 2021). The train and test datasets have been designed to cover as much as possible the various possible layouts that exist in the "Fond régional: Franche Comté" dataset. We have divided them into three layout types:

1c documents where the text is displayed in one column, as in books;

2c documents where the text is displayed into two columns;

3c+ documents where there are at least 3 columns of text, as in newspapers.

Table 1 shows the distribution of documents across the three layout types in our datasets.

Dataset / layout	1c	2c	3c+	Total
Train	18	5	25	48
Test	2	2	2	6

Table 1: Document layouts in the train and test datasets

[2] https://gallica.bnf.fr

The documents in the corpus cover three general topics: Catholicism, Resistance and News. The documents of the Catholic topic were published between 1900 and 1918. Most of them, such as "Bulletin paroissial de Censeau" or "Petit Écho de Sainte-Madeleine", are bulletins of small parishes. As such, they focus mainly on the local religious life, although they sometimes discuss national and international events such as WWI. The documents from the Resistance topic, such as "La Haute-Saône libre" or "La Franc-Comtoise", were published between 1939 and 1945 by Resistance fighters. As such, their main goal is to relay information about the ongoing local and international events of WWII. Finally, the documents of the News topic were published in the 1930s and focus on local and national events. Some are apolitical such as "Le Franc-Comtois de Paris", while others have a political label. For instance, "Le Semeur" and "Le Front Comtois" are left-wing newspapers whereas "Vers l'Avenir" is a right-wing Catholic newspaper.

The French language used in these documents is not very different from modern French. However, we notice some variations in the written styles between the three topics. The written style in the Catholic document is formal and literary and uses many religious metaphors. On the other hand, the written style in the News document is mostly standard, although sometimes formal. Sentences are shorter and use simpler tenses than the Catholic documents. This simplification of the writing style is even more prominent in Resistance documents. The difference in the writing style between documents can first be explained by their domain: religious text should be more literary than newspapers or Resistance periodicals. This difference can also be explained by the size of the documents. Catholic documents are the longest in the corpus, with more than 10 pages on average. As such, their text can be more elaborate. On the other hand, News and Resistance documents are respectively four and two pages long on average. Their text is factual and concise in order to convey a lot of information in the limited space they have.

All the documents are stored in the XML ALTO format, which contains descriptions of their physical layout and the text content obtained from OCR. As such, the files already provide the physical layout analysis and the reading order of the documents.

The XML ALTO format provides the text con-

Figure 1: Excerpt of the first page of the second issue of the communist newspaper *Le Semeur* published the 23rd of April 1932

Dataset	Newspapers	Issues	Text blocks	Text lines	Words	Pages
Train	15	48	4 608	51 815	338 583	368
Test	6	6	1 445	8 836	63 343	52

Table 2: Train and the test datasets

tent and physical layout of documents in the following manner. The OCR output for the whole document is available in a PrintSpace tag. Lines of text are contained in TextLine tags, which in their turn contain String tags for words and SP tags for spaces. TextLine tags are grouped into blocks in TextBlock tags. Sometimes, TextBlock tags are also grouped into ComposedBlock tags. TextBlock and TextLine tags have the following attributes:

Id the tag's identifier

Height, Width the text height and width

Vpos the vertical position of the text on the page. The higher the value, the lower the word is on the page

Hpos the horizontal position of the text on the page. The higher the value, the further on the right the text is on the page

Language the language of the text (only for TextBlock tags).

Among the attributes listed above, some TextBlock tags also have a Type attribute. This attribute is useful as it contains the logical labels of the lines in the block. It appears most often for tables or advertisements. However, TextBlock with a Type attribute are rare in our dataset. As shown in Table 3, nearly 98% of the TextBlock tags in the train and the test datasets do not have a Type attribute.

	Train		Test	
Type attribute	Count	Perc.	Count	Perc.
No attribute	4 514	97.96	1 423	98.48
illegible	79	1.71	15	1.04
titre1	15	0.33	0	0
advertisement	0	0	4	0.28
table	0	0	2	0.14
textStamped	0	0	1	0.07

Table 3: Type attribute distribution on TextBlock tags in the train and test datasets

4 Methodology

Our algorithm aims to attribute logical layout labels to both TextBlock and TextLine tags in documents. In the following subsections, we present the tagset that is used, then we explain the general processing pipeline of the algorithm. Finally, we present in detail the features that are used by the algorithm and the sets of rules. The diagram in Figure 2 shows the processing pipeline as described in this section.

We defined sets of rules for the annotation of TextBlock and TextLine tags. They are applied to documents regardless of the layout category they belong to. These rules were designed using heuristics based on observations that we made in the train

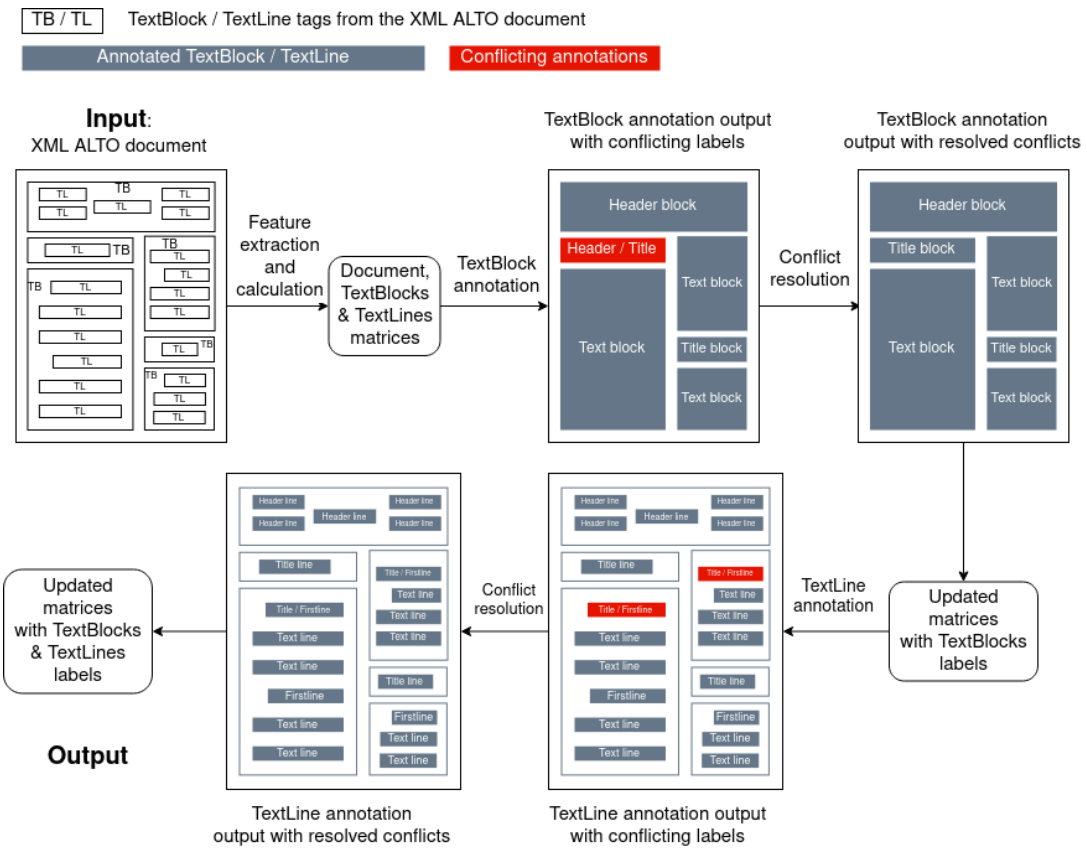

Figure 2: Diagram representation of the main stages of the algorithm

dataset. For instance, we observed that the biggest titles in the documents start with a capital letter and are surrounded by important spaces. Then, we translated these patterns into rules that we could use. This required us to extract features from the XML ALTO document, and to compute other features that are not available directly, such as the space between lines or the case of the first letter of a line.

4.1 Logical Layout Tagset

To perform the Logical Layout Analysis of the documents, we define the following annotation tagset:

- TextBlock labels: Text, Title, Header, Other;
- TextLine labels: Text, Firstline, Title, Header, Other.

The label "Firstline" must be understood as "first line of the paragraph". Thus, any TextLine tag labelled Firstline will indicate the beginning of a paragraph.

The whole dataset has been manually annotated by a single annotator, then split into a train and a test dataset. Table 4 shows the label distribution in the datasets. The train set was used to develop the rules presented in Tables 6 and 7. The test set was kept blind until the final evaluation of the system.

A small portion of the TextBlock and TextLine tags correspond to elements that are not relevant for our study, such as images, tables or advertisement. Those elements were labelled as "Other" and are ignored for the evaluation. Our system will assign the label "Other" to a TextBlock or a TextLine tag only if no other label has been assigned to it already.

4.2 General processing pipeline

The first step of our processing pipeline extracts features from the XML ALTO document at the TextLine, TextBlock and Document levels. The exact features extracted for each level are presented in Section 4.3. These features are grouped into three categories: geometric, morphological and semantic, as in Rangoni et al. (2011), Bitew (2018), Abreu et al. (2019), Tomas Hercig (2019) and Giguet and Lejeune (2019). Geometric features correspond to the physical attributes of the tags such as its height, width, or position in the document. Morphological features concern aspects of the text inside the tags,

		Train		Test	
	Label	Count	Percentage	Count	Percentage
TextLine	Text	36 272	70.138	6 648	75.881
	Firstline	9 785	18.921	1 563	17.840
	Title	1 820	3.519	234	2.670
	Header	740	1.430	115	1.312
	Other	3 098	5.989	201	2.293
TextBlock	Text	2 064	45.724	1 102	80.203
	Title	429	9.503	90	6.550
	Header	333	7.377	53	3.857
	Other	1 686	37.35	128	9.314

Table 4: TextBlock and TextLine tags label distribution over the train and test datasets

for instance finding if a line starts with a capital letter or a digit. Finally, semantic features concern the content of the tag, like the presence of a specific keyword. We store these features into two matrices for TextLine and TextBlock features and in a dictionary for Document features. Each row in the matrices represents either a TextLine or a TextBlock tag and each column is a corresponding feature.

The second step attributes logical labels to TextBlock tags. Labelling TextBlock before TextLine is important because the presence of a Type attribute in TextBlocks can help label the lines inside these blocks. The goal of this step is to add a Type attribute to every TextBlock. To do so, we process the TextBlock feature matrix from the previous step by applying sets of annotation rules, one for each possible logical label. A TextBlock is only processed if it doesn't already have a Type attribute. Because the sets of rules are applied independently from each other, a same TextBlock can obtain multiple labels. Another set of rules is then applied to solve such conflicts and keep only one possible logical label for each TextBlock, which is then set as the value of the Block's Type attribute in the feature matrix. The complete sets of rules to annotate TextBlock tags and solve conflicts are presented in Section 4.4.

The third step attributes logical labels to TextLine tags. Every TextLine is by default labelled as Text. The system then applies rules to identify the other labels. First, any TextLine in a Title or a Header block inherits the same label. Then, any TextLine contained in a TextBlock is processed by a set of rules in order to identify Firstlines and possible missing Titles. Similarly to the previous step, rules are applied independently from each other, resulting sometimes in conflicting predictions. The TextLine feature matrix is processed a second time to solve conflicting predictions and keep only one possible label for each TextLine tags. This step also controls that any line that follows a Title is labelled as Firstline and that the first line of the document is labelled as Title if it not already labelled as Header. The complete sets of rules to annotate TextLine tags and solve conflicts are presented in Section 4.5.

The algorithm finally outputs the three feature matrices, where the TextBlock and TextLine matrices have been updated with the annotations of both steps 2 and 3.

4.3 TextBlock, TextLine and Document features

Our algorithm uses sets of features that are extracted and calculated from the XML ALTO document at three different levels: TextLine, TextBlock and Document level. Table 5 presents all the features with their descriptions and levels. The information on these features for all document elements, in the form of matrices, is the input of the annotation rules that are described in the following subsections.

The header words set, which is used for the calculation of the *simHeaderSet* feature, is made up of the following words or phrases: *Rubrique Locale, Gérant, Publicité, Abonnement, Envoyez les fonds, Conservez chaque numéro, Rédacteur, Directeur, Numéro, Chèque postal, Dépôt, Achat-Vente-Echange, Annonce, Imprimerie, En vente partout, Paraissant*. This list is necessary for the annotation of the TextBlocks that represent the headers of the newspaper pages. It has been created by observing the different types of headers that exist in the datasets.

4.4 TextBlock annotation rules

The annotation rules that we have defined use sets of conditions that must be verified on the features of the TextBlock elements. All the rules are applied

Feature	Description	TextLine	TextBlock	Document
page	page number of the page containing the element	X	X	
blockType	type of the block	X	X	
wordCount	number of words	X	X	
precedingSpace, followingSpace	spaces between the element and those before and after it	X	X	
capitalPro, digitProp	proportion of capital letters and digits	X	X	
height, width	height and width values of the line	X		
hpos, vpos	coordinates of the line on the page, i.e. its horizontal and vertical position	X		
diffHpos	the difference between *hpos* and the median *hpos* value in the block	X		
stwCapital, stwDigit	True if the line starts either by a capital letter or a number, False otherwise	X		
headerMark1	True if the element contains the word "Page" or a dash sign. False otherwise.	X	X	
headerMark2	True if the element contains a date, a currency, an address. False otherwise.	X	X	
simTitle	similarity of the line with the title of the document, calculated by the Levenshtein distance	X		
simHeaderSet	highest similarity of the line with the words contained in the header words set, calculated by the Levenshtein distance	X		
firsthpos, firstvpos	coordinates of the first line of the block		X	
lasthpos, lastvpos	coordinates of the last line of the block		X	
linecount	number of lines		X	
medHeight, medWidth	median line height and line width		X	X
medHpos, medVpos	median *hpos* and *vpos* values in the block		X	
medWordCount, medLineSpace	median number of words by line and the median space between lines in the block		X	X
wordRatio	number of words by line		X	
medBlockHeight, medBlockWidth	median line height and block height and width			X
medBlockSpace	median space value between blocks			X
thirdQuartileLineSpace	third quartile of line space values in the document			X
medWordRatio, medLineCount	median number of words by line and median number of line by block in the document			X

Table 5: List of features used by the algorithm

to all TextBlock tags in the documents. Identifying Text and Title blocks relies on geometric and morphological features, whereas identifying Header blocks relies on semantic features.

Text blocks contain relatively more lines and more words than other blocks in the document. Title blocks are TextBlock tags that contain few lines, usually not more than 3. The role of a title is to introduce the topic of a text section, thus a Title block should be surrounded by Text blocks. The space around that block should also be important, in order to stand out with the surrounding blocks. A Text block should have a smaller height than a Title block. As such, if there is a confusion between Text and Title block, we use the height of the block to distinguish between the two.

Headers contain very specific information about the document, such as its title, its price, a date or the publisher's name. This information is displayed with keywords and sentences that are recurrent across multiple pages and documents. As Header blocks are only located at the top of a page, we only look for this information in the first four lines of a page. Small blocks at the top of a page are most likely Headers. Considering the first page, we look for the header in the first 30 lines, because the first page's header contains more information.

Table 6 presents all annotation rules for TextBlock tags and their corresponding annotation labels, where B is a TextBlock in a document D. The last two rules, 6 and 7, solve conflicting annotations.

4.5 TextLine annotation rules

Naturally, TextLine tags that are contained in a Title or Header block inherit this annotation. TextLine tags that appear between two Header lines are also annotated as Header. To find Firstline and missing Title lines, we apply sets of rules that rely on geometric and morphological features.

TextLines inside Text blocks are processed in order to identify Firstlines and possible missing Titles. The first line of a paragraph always starts with a capital letter, and most of the FirstLine are

Rule	Condition	Label
1	*(B.linecount > D.medLineCount)* or *(B.wordCount > D.medWordCount/3)*	Text
2	Previous and next TextBlocks are Text and *(B.linecount < D.medLineCount)* and *(B.medHeight < D.medBlockHeight)*	Text
3	Previous and next TextBlocks are Text and B is not Text and *(B.linecount < 4)* and *(B.precedingSpace > D.medBlockSpace) or (B.followingSpace > D.medBlockSpace)*	Title
4	*B.page = 1* and for any of the first 30 lines of B: *simHeaderSet > 0.9* or *simTitle > 0.9* or *headerMark1* or *headerMark2* or *ctnTotal*	Header
5	*B.page > 1* and for any of the first 4 lines of B: *simHeaderSet > 0.9* or *simTitle > 0.9* or *headerMark1*	Header
6	Conflicting annotation: Header and (Text or Title):	
	(B.linecount < 15) and *(B.wordCount < 50)*	Header
	Otherwise	Text / Title
7	Conflicting annotation: Text and Title:	
	B.medHeight > D.medBlockHeight / 2	Title
	Otherwise	Text

Table 6: TextBlock annotation rules and conflict resolution rules

indented. For this reason, we select TextLines that have a Hpos value greater than the other TextLines in the block. The Firstlines that are not indented can be identified if the line that precedes them is shorter, indicating the end of the previous paragraph. Finally, the first line of a page or immediately after a Title is labelled Firstline, if it starts with a capital letter.

Like Title blocks, Title lines are surrounded by relatively more space in order to stand out from other text sections. The smaller the title is, the less important the space around it is. Small titles usually contain more capital letters and are center-aligned. Thus, all these criteria enter consideration for the identification of Titles.

Table 7 presents the TextLine annotation rules and their corresponding annotation labels, where L is a TextLine in a document D and B is the TextBlock that contains L. The last two rules, 11 and 12, solve conflicting annotations.

5 Evaluation

To evaluate our method and the proposed annotation rules we have run the algorithm through the test dataset. Table 8 shows the Precision, Recall and F1 scores for the TextBlock and TextLine classification steps.

TextBlock annotation is an intermediary step in the algorithm. TextBlock annotation rules perform best on documents from the 2c layout category. Title classification for TextBlocks performs with F1 score of 0.61 on average and 0.94 on documents from the 2c category. Header classification for TextBlocks provides a good precision score (0.726) but with a low recall (0.298).

Similarly to TextBlock annotation rules, TextLine annotation rules perform best on documents from the 2c category. Title identification performs worse on 1c documents, and obtains overall F1 score of 0.639 for all layouts. Firstline identification performs fairly well with an F1 score above 0.9. Header identification obtains a good precision score (0.803) but with a recall of 0.348. This means that header identification rules are insufficient and need to be completed to capture the various types of headers.

A first type of error comes from errors in the Block classification step. As any line in a Title or Header block inherits that annotation, the precision of TextBlock annotation is an important factor for the overall performance of the algorithm.

A second type of error is the confusion between Titles and First lines. Most Titles mislabelled as Firstline are short subsection titles. As such, they are similar to other text lines in terms of typography, and are hard to detect with the features we use. This confusion happens mainly in documents from the 2c and 3c+ categories. Other mislabelled Titles are one-line paragraphs such as greetings or signatures, or the beginning of a text section. Such lines have properties similar to Titles, being surrounded by important spaces and being either center or right-aligned. Extracting features about the font style of the line (bold, italics) and its alignment (left, center, right-aligned) could help solve this confusion.

6 Conclusion and Discussion

In this article, we have presented a rule-based system for the Logical Layout Analysis of XML ALTO documents. Our system starts by extracting features from the document, then uses these features to add logical labels to TextBlock and TextLine tags. We have described the construction and the evaluation of the proposed annotation

Rule	Condition	Label
1	L.precedingSpace = 0 and L.followingSpace > D.medLineSpace and L.simTitle < 60 and L.simHeaderSet < 60 and L.stwCapital	Title
2	L.wordCount < B.medWordCount and L.precedingSpace > D.thirdQuartileLineSpace and L.followingSpace > D.thirdQuartileLineSpace	Title
3	L.capitalProp > 10 and L.wordCount < B.medWordCount and L.height < B.medHeight and (L.precedingSpace > D.thirdQuartileLineSpace or L.followingSpace > D.thirdQuartileLineSpace)	Title
4	L.diffHpos > 104 and L.capitalProp > 0 and L.precedingSpace > D.medLineSpace and L.followingSpace > D.medLineSpace	Title
5	L.hpos > B.medHpos and L.diffHpos < 105 and (L.stwCapital or L.stwDigit)	Firstline
6	L.width < B.medWidth and L.wordCount < B.medWordCount and L.Hpos < B.medHpos	Lastline
7	Previous TextLine is LastLine and L.stwCapital and L.followingSpace < B.medLineSpace	Firstline
8	Previous TextLine is not Lastline and L.stwCapital and L.precedingSpace > B.medLineSpace and L.followingSpace < B.medLineSpace	Firstline
9	Previous TextLine is not Lastline and L.stwCapital and L.hpos > B.medHpos	Firstline
10	None of the rules 1-9 above is True	Text
11	Conflicting annotation: Header and other label: Previous TextLine is Header and next TextLine is Header	 Header
12	Conflicting annotation: Title and FirstLine: L.followingSpace < B.medLineSpace and L.capitalProp < 15 Otherwise	 Title Firstline

Table 7: TextLine annotation rules

		Text			Title			Firstline			Header		
	Cat	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TextBlock	1c	0.947	0.938	0.942	0.312	0.357	0.333				0.679	0.373	0.476
	2c	0.973	0.989	0.981	0.899	1.000	0.947				1.000	0.271	0.411
	3c+	0.958	0.973	0.965	0.589	0.560	0.551				0.500	0.250	0.333
	Mean	0.959	0.966	0.962	0.600	0.639	0.610				0.726	0.298	0.406
TextLine	1c	0.979	0.986	0.983	0.354	0.720	0.473	0.943	0.854	0.895	0.909	0.598	0.721
	2c	0.961	0.995	0.978	0.746	0.765	0.747	0.955	0.859	0.902	1.000	0.118	0.197
	3c+	0.975	0.992	0.983	0.703	0.702	0.702	0.952	0.877	0.913	0.500	0.400	0.444
	Mean	0.969	0.991	0.979	0.595	0.733	0.639	0.949	0.861	0.902	0.803	0.348	0.435

Table 8: Precision, Recall and F1-score for TextBlock and TextLine annotation

rules. This methodology provides very good results for some categories like Text, Firstlines in most cases, but struggles with other labels such as Headers or Titles. Most errors in our system can be corrected by either adding new rules or by refining the already existing ones. The system could also benefit from adding new features such as font style and line alignment.

While recent methods in NLP use extensively machine learning and deep learning architectures, such approaches require large annotated datasets. To the best of our knowledge, no such datasets exist for the logical layout analysis of historical newspapers in French. For this reason, the algorithm that we propose in this paper is manually designed and rule-based. Its objective is, above all, to be able to produce annotated datasets that are large enough to envisage machine learning or deep learning approaches. The comparison between the performance of these rules and the results of recent deep learning architectures will be the object of our future work.

We devised the rules to process documents regardless of their era. As stated earlier, the layout in historical documents evolves rapidly, especially in newspapers. In order to create sets of rules dedicated to the different publication periods, we plan in future works to apply rule learning algorithms to generalise the creation of rules.

Acknowledgments

This research is supported by the Région Bourgogne Franche-Comté, France, as part of the EMONTAL project (2020–2024).

References

Carla Abreu, Henrique Lopes Cardoso, and Eugénio Oliveira. 2019. Findse@fintoc-2019 shared task.

Hanna Abi Akl, Anubhav Gupta, and Dominique Mariko. 2019. FinTOC-2019 shared task: Finding title in text blocks. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 58–62, Turku, Finland. Linköping University Electronic Press.

Raphaël Barman, Maud Ehrmann, S. Clematide,

S. Oliveira, and F. Kaplan. 2020. Combining visual and textual features for semantic segmentation of historical newspapers. *ArXiv*, abs/2002.06144.

Semere Kiros Bitew. 2018. Logical structure extraction of electronic documents using contextual information.

Marius Bulacu, Rutger van Koert, Lambert Schomaker, and Tijn van der Zant. 2007. Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 357–361.

Kai Chen and Mathias Seuret. 2017. Convolutional neural networks for page segmentation of historical document images.

Christian Clausner, Christos Papadopoulos, Stefan Pletschacher, and Apostolos Antonacopoulos. 2015. The enp image and ground truth dataset of historical newspapers. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 931–935.

Emmanuel Giguet and Gaël Lejeune. 2019. Daniel@FinTOC-2019 shared task : TOC extraction and title detection. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 63–68, Turku, Finland. Linköping University Electronic Press.

Nicolas Gutehrlé and Iana Atanassova. 2021. Dataset for Logical-layout analysis on French historical newspapers.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64.

Reza Farrahi Moghaddam Hossein Ziaie Nafchi, Seyed Morteza Ayatollahi and Mohamed Cheriet. 2012. Persian heritage image binarization dataset (phibd 2012).

David Hébert, Thomas Palfray, Stéphane Nicolas, Pierrick Tranouez, and Thierry Paquet. 2014. Automatic article extraction in old newspapers digitized collections. *ACM International Conference Proceeding Series*.

K. Kise, M. Iwata, and Keinosuke Matsumoto. 1999. On the application of voronoi diagrams to page segmentation.

S. Klampfl and Roman Kern. 2013. An unsupervised machine learning approach to body text and table of contents extraction from digital scientific articles. In *TPDL*.

G. Nagy, S. Seth, and M. Viswanathan. 1992. A prototype document image analysis system for technical journals. *Computer*, 25:10–22.

Anoop Namboodiri and Anil Jain. 2007. *Document Structure and Layout Analysis*, pages 29–48.

D. Niyogi and S.N. Srihari. 1995. Knowledge-based derivation of document logical structure. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 472–475 vol.1.

L. O'Gorman. 1993. The document spectrum for page layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:1162–1173.

Cartic Ramakrishnan, Abhishek Patnia, Eduard Hovy, and Gully Burns. 2012. Layout-aware text extraction from full-text pdf of scientific articles. *Source code for biology and medicine*, 7:7.

Y. Rangoni, A. Belaïd, and Szilárd Vajda. 2011. Labelling logical structures of document images using a dynamic perceptive neural network. *International Journal on Document Analysis and Recognition (IJDAR)*, 15:45–55.

Martin Riedl, Daniela Betz, and Sebastian Padó. 2019. Clustering-based article identification in historical newspapers. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 12–17, Minneapolis, USA. Association for Computational Linguistics.

Zejiang Shen, Kaixuan Zhang, and Melissa Dell. 2020. A large dataset of historical japanese documents with complex layouts. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2336–2343.

Fotini Simistira, Mathias Seuret, Nicole Eichenberger, A. Garz, M. Liwicki, and R. Ingold. 2016. Divahisdb: A precisely annotated large dataset of challenging medieval manuscripts. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 471–476.

H. Tibbo. 2007. Primarily history in america: How u.s. historians search for primary materials at the dawn of the digital age. *American Archivist*, 66:9–50.

Pavel Král Tomas Hercig. 2019. UWB@FinTOC-2019 shared task: Financial document title detection. In *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*, pages 74–78, Turku, Finland. Linköping University Electronic Press.

Xu Zhong, J. 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.

Annus Zulfiqar, Adnan Ul-Hasan, and Faisal Shafait. 2019. Logical layout analysis using deep learning. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–5.