

Document: a Useful Level for Facing Noisy Data

Hervé Déjean
XRCE

6 chemin de Maupertuis
38240 Meylan

Firstname.Lastname@xrce.xerox.com

ABSTRACT

In this paper we will present a set of experiments using large digitalized collections of books to show that logical structures can be extracted with good quality when working at document level. The proposed solution relies on a twofold method: first specific logical elements are recognized by a given method. Then models for the recognized elements are generated by combining layout, content and labeling information. These inferred models combining several kinds of information are used to correct noisy data, typical zoning, OCR and labeling errors produced by previous processing steps. This method is illustrated with the extraction of page numbers and chapter headings, two navigating elements required by digital libraries.

Categories and Subject Descriptors

I.7.5 [Document and Text Processing]: Document Capture - Optical character recognition (OCR) - Document analysis

General Terms

Experimentation

Keywords

Logical Analysis, error correction, model

1. INTRODUCTION

Mass digitization of books is now a reality, and images of tens of millions of books are now available (Google alone announced 10 millions scanned books in October 2009). Additionally, uncountable PDF files (or any "unstructured" format) used to exchange data increase the need for robust Document Analysis systems. While in specific domains, such as form processing, some applications daily process thousands of pages, this recent amount of document requiring analysis challenges the Document Analysis and Understanding in terms of quality and robustness. Many libraries are now providing access to digitalized (scanned) documents, but the associated metadata, especially navigation metadata are almost always added manually at a high cost. One exception can be found in Google Books, where tables of contents

Jean-Luc Meunier
Jean-Luc Meunier
XRCE

6 chemin de Maupertuis
38240 Meylan

Firstname.Lastname@xrce.xerox.com

are automatically generated (as far as we can judge), and proposed to the reader. For scanned documents, OCR quality varies from document to document, and quality assurance, as discussed in [12, 17] is still an issue for large collections. The Document Analysis community is aware of this issue as mentioned by several articles [1, 18]. Some works addresses the impact of errors on later stages, such as [13] which studies the impact of OCR errors on Natural Language Processing applications, and [2] in the field of Information Retrieval. In this paper, we would like to present an alternative perspective: **Instead of considering the impact of errors on later stages, we can consider the later stages to correct errors.** How is this possible? Easily by changing the traditional level of processing, the page, up to the document itself, as pointed out by Gravenhorst [7, p. 13]:

"One of the interesting approaches to solving these problems [OCR errors] could be to look at the complete document instead of a single page, and to use this document structure information for improving the OCR results. Such structure information could also automatically recognise logical entities such as headings, captions etc... Since these are much more important for retrieval than running text, OCR correction could then focus on these elements. With the increasing mass digitisation, such a 'next level OCR' is very needed to increase the speed and lower the hardware cost".

In Logical Analysis, there is indeed a recent trend over the last few years to use methods that consider the document as processing level, such as in the following work: detection of page headers and footers [10], table of contents [3, 11], page numbers [5], page template [6]. These methods heavily rely on the document level. And even at the OCR level, some work uses the whole document such as the "whole book" approach [19] and the adaptive OCR [9]: *"The principle underlying Adaptive OCR is that it may be possible to improve OCR capabilities by creating an OCR engine that would 'tune' itself to each work being processed".*

Last, ABBYY released with FineReader 8.0 the adaptive document Recognition Technology (ADRT®): *"ABBYY ADRT enables re-creation of logical structure and formatting elements in multi-page documents. The technology analyses and processes a multi-page document as a single entity rather than a batch of individual and independent pages and converts paper documents, images and PDF files to the Microsoft® Word format more accurately than ever".*

All these examples argue that the document level allows for developing efficient methods to label some logical structures, but we also argue that it is also a natural and very good level for error correction at latter stages. This has been shown for the character

recognition task [9, 19], and we will illustrate it for some logical structures: even with noisy OCR, some logical structures can be robustly recognized using structural redundancy which appears at the document level. Recognition is possible but also **correction**: By inferring a **model** of a given logical structure, this model can be used in order to cope with noisy elements that were first missed by the method.

The next section will explain this proposal in more detail. In Section 3, we will illustrate it with two document structures: page numbering and page template. In the last section, we will discuss the results of the experiments and the current extensions under progress.

2. A PROPOSAL

Our proposal relies on the use of two characteristics: document as processing unit (and not page), and the inference of model for assessing and correcting logical labeling. Regarding a document as processing unit points out a mere but important fact: at this level emerges redundancy, in its basic and computational aspect: repetition (of structures). A partial recognition of elements of a given document structure allows for designing a model from the recognized elements and to use this model to find out missing elements, missed due to errors from previous steps for instance.

This work is related to the method explained in [4], which uses also a second step in order to improve results. But in this work the emphasis is put on the error correction, correction which can go back to previous stages down to zoning. The three main steps are:

1. Logical labeling
2. Inference of models
3. Correction based on inferred models

The first step consists in labeling a logical structure of a document (this can be achieved by any means). Then from this set of identified elements, models of these elements are inferred, possibly combining layout, content, and label information. Finally the inferred models are used in order to find some elements not recognized initially due to errors. Based on the information present in the models, errors can be corrected in some cases.

Two observations regarding the model: **First, since the labeled elements belong to the same document, they share some regularities that make model inference possible under two conditions: the amount of initial errors is not too high and the labeled structures have to be frequent enough in the document.** One issue of his approach concerns structures that only occur a few times in a document. By experience, structures occurring less than 5 times are difficult to capture with this method, the inference of the corresponding model becoming unreliable. **Second, the combination of several aspects (geometric layout, content, logical labels) allows the method to cope with noise occurring in these different aspects.**

Finally, a correction step is performed using this model. As said, combining several types of information in the model allows for correcting several types of errors: layout information may allow for zoning and OCR correction, content information may allow for OCR correction, and label information for label correction, but also OCR. We are currently investing how to correct a fourth type of error: segmentation (at word or line level).

3. ILLUSTRATIONS

We will now illustrate this methodology with two examples: the detection of page numbers and chapter headings. Many figures are provided for the experiments. They do not provide a decisive argument for our proposal, and a fine-grain discussion requires to know in detail the different algorithms used by our components. But we hope they are convincing enough in the sense that they show that the proposal improves results, and can be easily implemented on top of many components.

3.1 Collections

We present here a set of experiments using this method. Since we want to show that this kind of method can cope with noise, and in the context of the mass digitalization, we took several large corpora where character recognition was already performed. The next table gives the characteristics of these corpora. Even though we are far from a very large collection, the corpora size amounts for around 420,000 pages, which is a fairly large dataset. Collections are provided with some logical metadata.

Table 1: Description of the various collections

corpus	number of books	number of pages	Type	Description
GOOGLE BOOKS	44	15073	scanned and ocr (Tesseract)	Books from the 19th century. Noisy metadata
UGOE	83	47805	scanned and ocr (?)	Reports, journals, and proceedings from the 19th and beginning of the 20th centuries. Metadata: Page numbers and logical structure at page level (chapters, illustrations)
INEX	1000	344034	scanned and ocr (Live Search book project?)	Books from the 19th and the beginning of the 20th centuries. Metadata: logical structure
PDF	208	16857	text-PDF	various PDF files (recent documents). No metadata

The metadata provided with Google Books (hereafter GB) are unfortunately too noisy to be used (especially the start of chapter tag). The books of the UGOE collection were provided by the University of Göttingen in the framework of the EU SHAMAN project (<http://shaman-ip.eu/shaman/>). The INEX collection is the collection used by the INEX book track competition [8]. The PDF collection is composed of various recent documents (from forms to technical documents), and is used as a kind of reference for the experiment regarding the page number detection: we consider that zoning character recognition is perfect for this collection, and then compare the characteristic of this collection with the scanned collections.

The experiments carry on two types of logical elements: the page numbers and chapter heading. These structures are of interest for navigating inside a book, and are desired by libraries and more generally content providers [18]. Furthermore the associated logical metadata for some the UGOE and INEX collections allow us to perform some qualitative evaluation.

3.2 Page Numbering

This logical element is interesting in more ways than one: page numbers are usually small and isolated pieces of content. This challenges the zoning and character recognition tasks. A description of the method we used to recognize page numbers is given in [5]. Working at document level, the method searches for incremental sequences which cover at best the document. A set of predefined but generic pattern family is provided, and global optimization using a Viterbi algorithm is used to generate the best coverage of the document by a set of page number sequences. The method enumerates in a greedy manner all the longest possible sequences of text fragments occurring on consecutive pages and fitting one of the predefined numbering schemes.

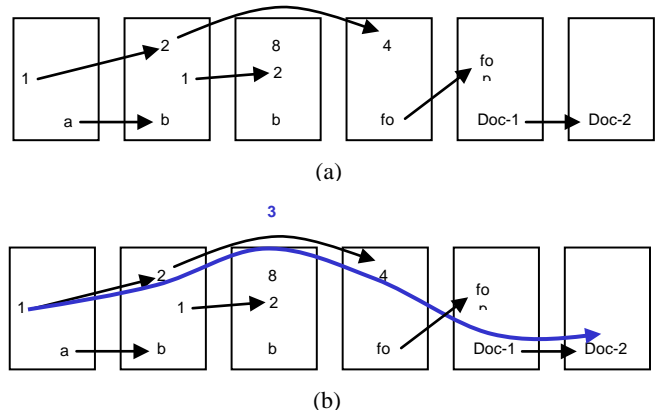


Figure 1 (a): All page number sequences are generated: (1, 2, x, 4), (a, b), (1, 2), (foo, fop), (Doc-1, Doc-2). (b): The best sequence is selected: (1, 2, 3, 4, Doc-1, Doc-2), and one guessed page number is generated: 3

Table 2: Examples of zones automatically extracted and associated output of Tesseract (GB/Volume_000)

Extracted zone				
Original OCR'ed text	VIH (Russian char !) I. THE YOUNG KING OF AIDH RUADH Gaelic	I 80 and they took down to meat	SS° his asian '!r·1t*..	NO TEXT FOUND
Targeted OCR (Tesseract applied to this zone)	viii NAME. I. THE YOUNG KING 'AIDH RUADB. Gulic n TL- 'lkl. -0 nu 111	180 md they M down to m»	3;% gz ag';.} gg \$n~☼☼☼u€\$us&s ifuwhiu	V`
OCR considering the whole page (Tesseract)	., ` ¥iii- KANE. fom nw , I. THE YOUNG KING	xoxo and took chwwn in	3;% gz ag';.} gg \$n~☼☼☼u€\$us&s ifuwhiu	NO TEXT FOUND

Figure 1 shows the list of sequences generated from an artificial 5-page document, and the final page number sequence.

An advantage of the method is to take into account during the optimization step missing page numbers in a sequence, characteristic required to deal with noisy text. A key parameter of the method is its *density*: the ratio of observed page numbers over the length of the page number sequence. When the density of a

sequence is smaller than the density threshold, the sequence is discarded.

Working at the document level, the method offers a very interesting and useful property: it guesses the value of the missing page numbers by considering holes in the sequences of page numbers. This absence can be due to conventional layout rule (first page of a chapter), but in the case of scanned documents, it is frequently due to zoning or OCR errors, as we will see it.

We ran the page number component over all of our collections (Table 2). Three main indicators are provided in this table:

1. the numbers of really observed numbers (which correspond to a correct piece of content in the page identified as page number),
2. the number of guessed elements, which correspond to missed elements in the page number sequence, and,
3. the number of pages without any associated number. They can correspond to document without page numbers, or pages without page number, such as front-matter pages or small missed sequences of page numbers.

Table 3: Indicators for the collections with different values for the density parameter

Collection	Density	observed numbers	Guessed numbers (ratio)	Pages with no number (ratio)
GB	0.10	8310	5110(38)	1653 (11)
GB	0.20	8353	4654 (36)	2066(14)
GB	0.30	8579	4225 (33)	2269(15)
GB	0.50	8169	2952 (27)	3952(26)
UGOE	0.10	31547	13456 (30)	2942(6)
UGOE	0.20	33711	11681(26)	2413(5)
UGOE	0.30	33691	11018(25)	3096(5)
UGOE	0.50	34200	9680(22)	3302(7)
INEX	0.10	252734	59382(19)	32166(9)
INEX	0.20	252603	58712(19)	32719(10)
INEX	0.30	248122	58912(19)	37000(11)
INEX	0.50	251923	54623(18)	36968(11)
PDF	0.10	12431	1120(8)	3346(20)
PDF	0.20	12422	1099(8).	3336(20)
PDF	0.30	12464	956	3437(20)
PDF	0.50	12505	808	3584(21)

Results are provided with different values of the *density* parameter. It shows that for noisy collections, this parameter has to be set up with a very low value. The noisiest collection, the Google books one, a density set up with 50 allows the method to find page numbers for only 74% of the pages, while a density set up at 10 allows for finding numbers for 89% of the pages, a figure which corresponds to a normal estimation for books. The INEX collection, similar in nature (books) has an equivalent ratio, whatever the density is. The UGOE collection is composed of documents, proceedings and journals, which have more numbered pages than "normal" books, and this explains its low ratio of pages without number.

A second figure shows that the GB collection may be very noisy: the number of **guessed page numbers** (fourth column) ranges from 27% to 38%. This number of guessed elements seems to be a very good indicator of the amount of noise for page numbers. As expected it is minimal with the PDF collection (around 7%), then increases from 19% for the INEX collection, to 38% for the GB collection. According to this criterion, the INEX collection, whose ratio of guessed elements is 19%, would be less noisy than the UGOE collection whose ratio ranges from 22% to 30% depending of the density value. A fact is consistent with this remark: the impact of the density parameter: the INEX collection is not sensitive to the different density values while it impacts the UGOE collection (the number of guessed elements goes from 22% up to 30 %). All this discussion assumes that the result of the page number detector is reliable. Evaluation with the UGOE collection corroborates this assumption.

We will develop this discussion in the final section and we can say here that the ratio of guessed elements can be considered as a reliable indicator of the OCR accuracy. We will now detail how errors are fixed with the model inferred from the observed numbers in order to reduce this ratio.

The page numbers are missed due to two kinds of errors:

1. zoning: the zone corresponding to the page number was not considered as text
2. OCR: the content of the page number was not properly recognized.

The page number **model** used by the method is simple but very efficient:

- content side: a sequence is defined by its counter schema (for instance, Arabic numbers, Roman numbers) and allows for accepting or not a given candidate.
- layout side: the most frequent positions of page numbers in a page are collected. In order to deal with page shift during scanning, the *page frame* (zone containing the foreground elements as defined by [14]) of the pages is computed and the relative position inside the page frame is used. The 4 most frequent positions of the observed page numbers over the document are used in order to deal with odd/even pagination.

Using this model, our objective here is twofold:

1. to make the page number detection more reliable by decreasing the number of guessed elements
2. to test if a *focused zoning* helps OCR: focusing on a specific zone of a page, can we improve character recognition?

We use the Tesseract OCR engine [16] in order to perform OCR on the selected page zones where no page number was found. No specific image pre-processing was performed on the page images, and we apply Tesseract directly on the zones without any tuning. Table 3 gives some examples of zones not properly recognized by the original OCR engine used for the collection, the Tesseract output for the targeted zone only providing the zone image, and the Tesseract output for this zone, but providing the whole page to Tesseract. The first and second cases are favorable cases where the new recognized text is correct. For the third case, the image quality is too poor for Tesseract. The last case is a white page where focused zoning is able to "invent" a text, but this content is not validated by the page numbers counters, and then discarded.

Table 4: Impact of reapplying OCR on indicators (observed, guessed elements) with two values of density (10, 20)

	dens ity	observed	Guessed	new numbers due to OCR	pages not covered
GB	0.10	8310	5110 (0.38)	na	1653
GB OCR	0.10	10568	3727 (0.25)	2339	778
GB	0.20	8353	4654 (0.36)	na	160
GB OCR	0.20	11324	2874 (0.20)	2352	875
UGOE	0.10	31757	13809 (0.3)	na	3002
UGOE OCR	0.10	34762	11152 (0.25)	1418	2654
UGOE	0.20	33711	11681 (0.26)	na	2413
UGOE OCR	0.20	34188	10502 (0.18)	1503	5

As shown by the different images, the zones are enlarged: the dimensions, computed for the observed elements are doubled as to cope with page shift during scanning. Other elements are also recognized by OCR, but usually discarded by the page numbers counters. The result of the OCR on these zones is filtered out using the possible page number counters of the document: only text matching these counters is kept. Taking the examples Table 3, among the first lines texts, only viii and 180 are kept while 3;% and V' are ignored.

Once new elements have been identified, the page number detector is re-applied considering old and new textual elements and the final page number list is generated still using the Viterbi algorithm to produce a global optimized solution.

What is the impact of this targeted OCR? Table 4 shows the evolution of our indicators after the OCR step. For the GB

collection: the number of guessed elements decreases significantly from 38% down to 24% (density=10), and from 36% to 20% (density=20), a reduction of 33% and 38% of the number of guessed elements. For the UGOE collection, the guessed elements decreased by 20% (density=10) and 10% (density=20). We expected a more significant improvement regarding the guessed elements. Looking at the focused zones, their quality seem correct, but looking at their OCR output, many OCR errors remain, as the third example Table 3 shows. Specific tuning for Tesseract seems to be required, especially for the ambiguity table (*DangAmbigs* file). Interesting could be to think in training Tesseract by associating the zones and the expected guessed values to build a training dataset, but currently the zones are too large to only contained page numbers.

Table 5 compares evaluation before and after OCR (indicated by +OCR in the first column). The available UGOE metadata do not provide ground-truth at the element level, but at the page level: the logical number of a page is known, but we do not know which element in the page corresponds to the page number. Since we integrate the guessed elements as part of the elements to be evaluated, the OCR step does not improve the F-1 score. But an evaluation at the element level (recognizing the element which corresponds to the page number) should show that the method improves by 25% the recall (the numbers of observed increases by 25%). For instance, comparing line 1 (density=0.1) and line 6 (density=0.1 + OCR), the number of guessed elements decreases from 13809 down to 11152, while the number of observed elements increases from 31757 up to 34762. If we only consider now the 5 documents with the lower F-1 score, the gain is more visible, the F-1 score for these documents increases from 85.7 up to 87.8 (+2.2).

Table 5: While limited, the OCR step improves results, and validates the fact that new elements are correct.

UGOE Collection		Precision	Recall	F-1
Density	observed/guessed elements			
0.10	31757/13809	96.9	92.2	94.5
0.20	33711/11681	97.0	91.6	94.2
0.30	33691/11018	97.1	90	93.3
0.40	31541/11608	96.8	87.9	92.1
0.10+OCR	34762/11152	96.3	92.7	94.5
0.20/OCR	34188/10502	96.8	92.5	94.4
0.30+OCR	34204/10155	96.5	92.2	94.3
0.40+OCR	34200/9680	96.3	91.1	93.6

Despite the somehow disappointing improvements, the method is validated: new observed elements generated by focused OCR are correct. To resume, this experiment shows that even in a noisy collection, page numbers detection is possible, and that the corrections are reliable.

3.3 A Second Illustration with Page Template

Page numbers can be considered as a simple document structure, and we now illustrate the method with a more complex structure: page template.

The method used here aims at recognizing specific page templates of a document, and among them the page templates corresponding to the first pages of chapters will be used for evaluation purpose. A page template is modeled as a graph where nodes correspond to **labeled** elements and edges correspond to geometric relations. Figure 1 shows examples of such page templates for a book of the INEX collection.

The method is described in [6]. First a logical analysis is performed in order to identify common document elements such as page numbers, page headers, table of contents and corresponding body entries, captions. After the identification of these elements, geometric relations are computed between the labeled elements occurring in the same page, and page templates are generated using frequent related elements (second step). The generated page templates correspond to what we call *model* in this article. As usual, OCR errors and segmentation errors introduce noise, and pages with noisy elements are not associated with the correct page template. The third step, the correction step, consists in applying a fuzzy matching operation between templates and pages which allows for recovering pages that were not initially recognized by a template. This fuzzy matching step also allow for labeling missed elements (for instance a page header).

This task has similarity with the previous one (page numbers), associating an information (template or number) with a page, but the way this association is performed is very different: the page number detector has a pretty simple number pattern algorithm, but uses global optimization to assign a number to a page, while the page template detector uses the document level for the template generation, and a matching algorithm is used to assign a template to a page (no global constraint).

Another difference is the type of errors which mainly impact the matching step: since the processing workflow now includes logical analysis, this adds a new type of errors, besides zoning and OCR errors: labeling errors.

Table 6 shows the following indicators for the INEX collection, composed of 427 books (146017 pages). The method is able to generate templates for 411 books and fails to generate models for 16 books. During the second step, 27217 additional pages are associated with a template. To be associated with a template, each labeled element of the template has to be identified in the page. The new associated pages correspond to pages with one or more errors, which were fixed by the matching algorithm (noisy non labeled elements are labeled thanks to the page template).

The 16 books without template correspond to relatively short books (altogether 1674 pages), where templates were composed of only one element: the page frame. Since this method only

considers templates composed of at least 2 elements, no template is associated with these books. It is noteworthy that the page number detector completely fails to detect page numbers for these books, due to a too high level of noise for these elements (mostly ignored during zoning).

Similarly to the previous experiment, we have indicators of erroneousess: the number of pages not covered by templates and the number of pages associated to a template thanks to the correction step.

Table 6: Coverage of templates before and after fuzzy matching

Total number of pages (427 books)	146017
Books without models (% of total pages)	16 (1%)
Number of pages associated with templates	119412 (82%)
Number of new pages associated with templates after correction step	27217 (19% of the total pages)
Number of white pages (no associated template)	8674 (6%)

As for the first experiment, we need to evaluate whether the correction step improves the result or simply introduces more errors. We focus on a specific page template, which corresponds to the first page of chapter (see Figure 2). We are able to evaluate such a template thanks to the INEX corpus whose purpose is "*test[ing] and compar[ing] automatic techniques for deriving structural information from digitized books in order to build a hyperlinked table of contents that could then be used to navigate inside the books*". [8]

For this evaluation, we select pages associated with a page template containing an element labeled "Title". Such templates generally correspond to the first page of chapter, and must occur in the INEX ground-truth. We evaluate the set of the selected pages against the INEX ground-truth (Table 7). The initial step (no correction) provides a good precision (93.8%), but a low recall (55.6). After the correction step, precision is still high (89.5, -4.11) and recall gains 10% up 65.6%. This evaluation could be considered as indicative of the effect of the error correction step for other templates: it should improve recall without penalizing too much precision.

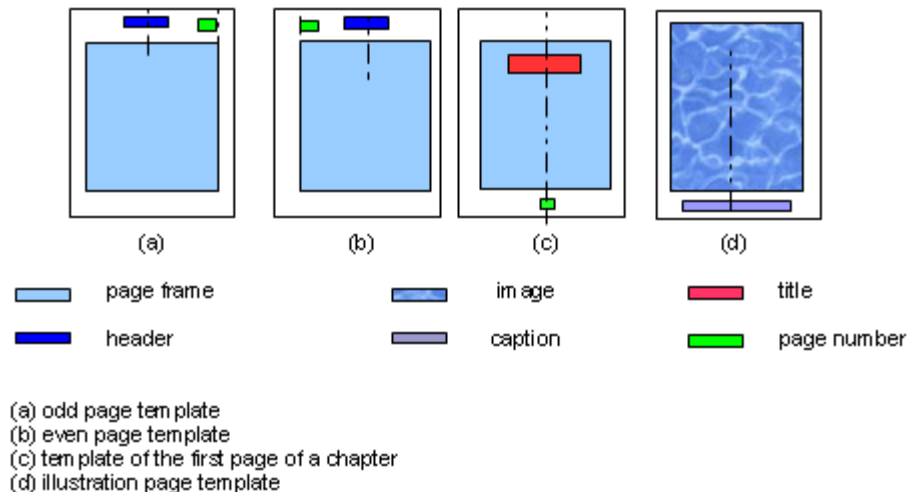


Figure 2: The different page templates generated from a document sample.

Table 7: Effect of the correction step for page templates (INEX)

Evaluation for the initial set of "Title" templates		Evaluation for "Title" templates with correction step	
Precision	Recall	Precision	Recall
93.6	55.6	89.5(-4.1)	65.6 (+10.0)

Most of the errors captured by the correction step are due to labeling errors. In the case of the "title" templates, pages were initially not recognized by the template due to unlabeled titles missed by the ToC entries detector.

We also found some errors due to software bugs, especially in the fuzzy matching algorithms which generate false positives. This is a general drawback of the methods: the more complicated models are, the more difficult applying them becomes. It is currently difficult to estimate the impact of such bugs before correcting them.

To conclude with this illustration, we would like to emphasize the interest of an object such as page template regarding error correction and quality assurance. These templates are built with several logically labeled elements and reflect the general layout of a document. They offer a very good context, combining layout and structural information (presence of other labeled elements), in order to correct elements present in a template.

4. DISCUSSION

What can be concluded regarding these experiments? First, structural redundancy present in a document is a very natural way to deal with noise. Our choice on how to use this redundancy among the many different possibilities is through the notion of *model*, by formalizing documents objects with different kinds of information.

Second, a two-stage approach, generating a partial labeling (with non noisy elements), then inferring a model of these objects and using it to identify and correct noisy elements is an efficient and generic solution. Efficient as experiments partially show, and generic, since it can be applied to many logical analysis components: Even though we show results for only two of them, we successfully conduct tests with others.

An interesting remark regarding quality assurance for large collection is that the ability for a method to provide indicators related to quality. The use of models can here provide a useful indicator along two lines: by estimating the coverage of models and by estimating the number of new elements found by using them. Can we estimate the quality of previous processing with such an observation? Taking as examples the page number detection, a key point would be to relate the OCR accuracy for page numbers to OCR accuracy for the running text of the document. Remember that all guessed elements do not correspond to errors: paginating is in no way an obligation, and in some cases no page number is printed on the page, as for some chapter title pages, and front- and back-matter parts. Acknowledging these cases and considering a guessed element as a zoning or OCR error, then OCR accuracy for page numbers over these collections varies from 94% (PDF collection) to 62%. On 180 pages of one book of the GB collection, [19] estimates word accuracy at 66% (word error rate at 34%). [15] evaluating a 19th century newspapers collection gives a *number group accuracy* around 64%. This is then in line with our estimation for the GB collection.

Other logical elements can be covered by such an approach. We are now investigating the structure (illustration, caption), which is prone to zoning error in our corpora. We have started this experiment, but the error level is for a large part of our corpora so huge, that building a model is challenging, and the components we used are to be tuned to be able to deal with such very noisy documents.

Last but not least, the traditional chicken or the egg dilemma: among the many Document Analysis components, how to

organized them. Many works use a simple organization: a pipeline. Adding a feedback mechanism at the component level (as we show for the page number component), can improve the quality of each pipeline step, but does not change the general shape of a system (pipeline). But we see in the notion of model a promising breakthrough for combining different logical structures, as illustrated by page templates: such models integrate several document objects and offer a good context to perform correction and validation for their document objects. Another advantage is that these models can be automatically inferred from a document, which ensures a last property required for mass digitization: **versatility**.

5. ACKNOWLEDGMENTS

This work is supported by the Large Scale Integrating Project SHAMAN, co-funded under the EU 7th Framework Programme (<http://shaman-ip.eu/shaman/>). We would also like to thank the reviewers whose comments have helped us improve this article.

6. REFERENCES

- [1] Henry Baird, Towards Versatile Document Analysis Systems, Proc., 7th IAPR Document Analysis Workshop (DAS'06), Nelson, New Zealand, February 12-15, 2006.
- [2] Steven M. Beitzel, Eric C. Jensen, David A. Grossman, A Survey of Retrieval Strategies for OCR Text Collections, Proceedings 2003 Symposium on Document Image Understanding Technology, 2003
- [3] Hervé Déjean, Jean-Luc Meunier: Structuring documents according to their table of contents. ACM Symposium on Document Engineering 2005: 2-9
- [4] Hervé Déjean, Jean-Luc Meunier: Logical document conversion: combining functional and formal knowledge. ACM Symposium on Document Engineering 2007: 135-143
- [5] Hervé Déjean, Jean-Luc Meunier, Versatile page number analysis, Document Recognition and Retrieval XV. Edited by Yanikoglu, Berrin A.; Berkner, Kathrin. Proceedings of the SPIE, Volume 6815, pp. 68150K-68150K-9 (2008).
- [6] Hervé Déjean, Unsupervised page Template inference, submitted to DRR 2011.
- [7] MPACT, OCR in Mass digitisation, conference proceedings, April 2009,
- [8] Gabriella Kazai, Antoine Doucet, Marijn Koolen, and Monica Landoni, Overview of the INEX 2009 Book Track, INEX Workshop pre-proceedings, pp. 120-129, 2009.
- [9] Vladimir Kluzner, Asaf Tzadok, Apostolos Antonacopoulos, Word-based Adaptive OCR for historical books, 10th international conference on document analysis and recognition, 2009, pp. 501-505.
- [10] Xiaofan lin, Removal of extraneous text from electronic documents, US patent application 20040139384, 2004
- [11] Xiaofan Lin, Yan Xiong: Detection and analysis of table of contents based on content association. IJDAR 8(2-3): 132-143, 2006.
- [12] Xiaofan Lin, Quality Assurance in High Volume Document Digitization: A survey, Hewlett-Packard Laboratories report, 2006
- [13] Daniel Lopresti, Optical Character Recognition Errors and Their effects on Natural Language Processing, And 2008.
- [14] Faisal Shafait, Geometric Layout Analysis of scanned documents, PhD thesis, University of Kaiserslautern, 2008.
- [15] Simon Tanner, Trevor Muñoz, Pich Hemy Ros, D-Lib Magazine July/August 2009 Volume 15 Number 7/8 Measuring Mass Text Digitization Quality and Usefulness Lessons Learned from Assessing the OCR Accuracy of the British Library's 19th Century Online Newspaper Archive
- [16] Tesseract, open source OCR engine, May 2008. <http://code.google.com/p/tesseract-ocr/>
- [17] Sherif Yacoub, Automated Quality Assurance for document Understanding systems, IEEE Software, 20(3), pp. 76-82, May-June 2003
- [18] Luc Vincent. Google Book Search: Document understanding on a massive scale. In Proceedings, IAPR 9th Int'l Conf. on Document Analysis and Recognition (ICDAR'07), Curitiba, BRAZIL, August 2007.
- [19] Pingping Xiu and Henry S. Baird, Analysis of whole-book recognition, DAS 2010.