OCD Dolores - Recovering Logical Structures for Dummies

Jean-Luc Bloechle[1,2], Maurizio Rigamonti[1,2], Rolf Ingold[1]
1. DIVA Group, Department of Computer Sciences
University of Fribourg, Switzerland
{firstname.lastname}@unifr.ch
2. Sugarcube Information Technology sàrl
{firstname.lastname}@sugarcube.ch

Abstract — **This paper presents OCD Dolores, an environment that aims at recovering the logical structures from documents by interactively inferring their models. Dolores is based on OCD, an XML canonical document format used to represent structured electronic content efficiently. The relevance of our restructuring system is assessed through a deep evaluation of Dolores' logical labeling capacities.**

Keywords - document structures, logical structure, document model, learning system, interactive learning, Dolores, XED, OCD;

I. INTRODUCTION

In 2011, document logical restructuring environments are desired more than ever, however very few systems emerged and existing ones are not generic at all and/or suffer from strong inertia, i.e., it is very difficult and time consuming to adapt new document classes. For instance, researches about recovering logical structures from electronic documents all target at customized restructuring tasks for very specific sets of documents [1, 2, 3, 4].

A generic and efficient system should provide functionalities such as interactive learning environments and assisted acquisition of document models. However, in current approaches, document models are either edited manually, a laborious task that has to be done by an expert user, or compiled from large sets of ground-truthed data, that have to be manually acquired during a demanding labeling phase. Note that compiled models tend to be static since they are tightly coupled to their ground-truthed data and, consequently, any model update requires much effort.

To put it in a nutshell, the creation of document models is one of the most time consuming tasks when researchers and end-users want to recover structures from documents. There is a lack of 1) flexible tools with user-friendly interfaces and visual feedbacks of users' actions and, 2) recognition improving with iterative use and models being refined incrementally. As a consequence, our researches led to the development of a complete and innovative document logical restructuring system called Dolores.

This paper is organized around five main sections. Section II first emphasizes the philosophy that guided us during the development of Dolores. Section III presents the interface of Dolores, its dynamic and interactive labeling environment. Section IV dissects the learning system integrated in Dolores. Section V gives a deep evaluation of our learning environment. And finally, Section VI briefly concludes by underlying the achievements of Dolores and foreseeing some ideas for potential improvements and extensions

II. DOLORES' PHILOSOPHY

Human beings are able to cope with new situations by using their knowledge; they learn through experiences and improve their own knowledge incrementally. In document analysis, new situations are also inevitable because there are an infinite variety of documents and, therefore, it is unconceivable to develop a universal system able to recognize any document structure a priori. Based on this assumption, a strategy similar to the "human way of learning" should be applied to the recognition of document structures. Accordingly, an incremental learning mechanism based on an interactive system (used to gain experience) seems the right path to follow.

In this sense, Dolores subscribes to the CIDRE - Cooperative and Interactive Document Reverse Engineering - [5] philosophy which precisely advocates the idea that an analysis system does not work in a fully automatic way, but cooperates with the user. Thus, Dolores provides an intuitive human-computer interaction: the system first proposes a solution, second, the human performs some correction, and third, the system analyzes the correction and adapts its model consequently, in order to propose a better solution.

On the one hand, interactions increase the efficiency of the system incrementally by learning from users' feedbacks, and by dynamically adapting itself to the document characteristics. On the other hand, users take full benefit from the visualization, by rapidly retrieving and correcting analysis errors.

The CIDRE philosophy was successfully applied in 2(CREM) [6], where the users can supervise the analysis of logical structures using the xmillum graphical user interface [7]. Dolores precisely takes its roots in 2(CREM) and replaces its rules-based built-in system with a dynamic learning approach.

III. THE INTERACTIVE LEARNING ENVIRONMENT

OCD Dolores – OCD DOcument LOgical REStructuring – is a high-level restructuring environment able to recover the logical structures from any textual document such as newspapers, scientific papers, journals, magazines, e-books, etc. OCD Dolores is based upon the OCD [8] file format (whereas previous versions used XCDF [9]), an XML representation able to efficiently store the content, layout and

physical structures extracted and recovered from electronic documents (thanks to XED [8]).

OCD Dolores integrates an innovative learning system supported by a GUI allowing document models to be inferred from interaction. User actions provide knowledge which is instantly injected in the model and reflected through the GUI in order to speed-up and simplify further user actions. In practical terms, the interface offers powerful labeling facilities by assisting the user thanks to visual hints; i.e., the user interactively improves the document model by checking color labels assigned to OCD text blocks and by correcting potential recognition mistakes. In order to keep Dolores intuitive and user friendly, all the learning system complexity and processing is "left behind the curtain".

Figure 1 shows the interface of Dolores, which highlights OCD text block labels with color shades. Labels are dynamically assigned by the recognition system to the text blocks. The user can assign or create new labels by right-clicking on a text block (popup menu). Labels confirmed by the user (a simple click on them) have thin surrounding stroked rectangles meaning that their corresponding text blocks are integrated in the model training set.

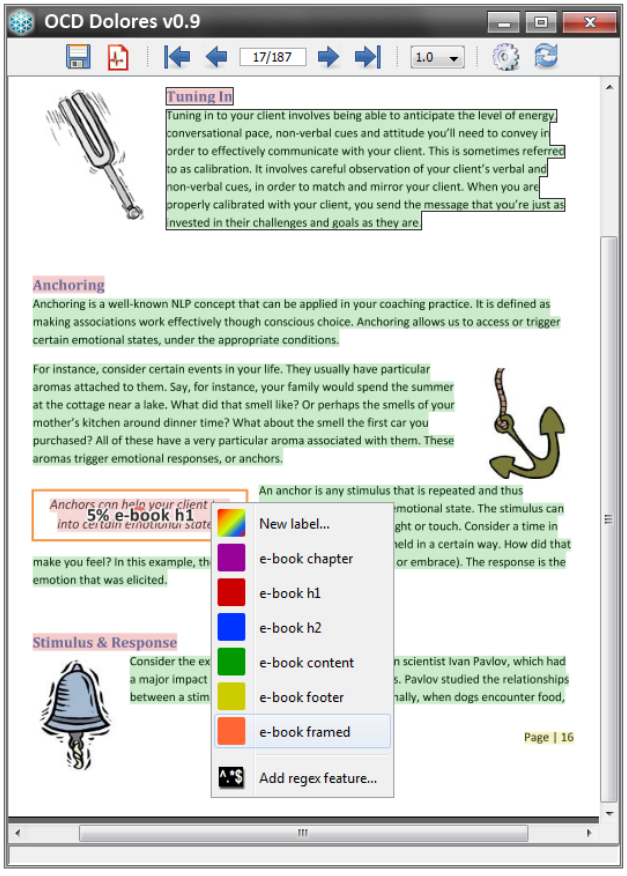

Figure 1. GUI of OCD Dolores in learning mode

Each time a label is confirmed by the user, the document model is updated "on the fly" and the interface is refreshed accordingly (by assigning label modification, if any, to text blocks). Such dynamic labeling of text blocks allows the user to visually estimate the accuracy of the model by observing the assigned colors. Additionally, a confidence value, corresponding to the degree of certainty given to a label, is shown as a percentage rate when the mouse cursor hovers a text block (cf. Figure 1). Poor confidence values are highlighted by simply displaying them on top of their text blocks. Benefiting from these visual hints, the user can decide to improve the document model either by labeling some misclassified text blocks (with the contextual popup menu), or by validating a text block having a low confidence rate (simply by clicking on it). A user can also confirm a page or even a complete document with a single click once all the page/document labels are correctly assigned by the model (thus feeding the model instantly).

Once the recognition error is deemed to be significantly low, the user can apply the inferred model on untrained documents to recognize their logical structures. In case of recognition errors the document model can still be improved by re-labeling some misclassified text blocks, thanks to the incremental learning facilities of Dolores.

IV. Model and Learning System

The features and functionalities of our interface implied to take some crucial considerations into account when developing our backing learning system. Both learning and classification processes have to be performed very efficiently because OCD Dolores must ensure an interactive response time, i.e., the processing time after each action must be reasonably short. In order to fulfill such needs, we elaborated our own neural network. Our document model is hence inferred by the set of confirmed labeled text block samples feeding a neural network.

The extracted set of features from the text blocks is therefore a key issue of our system, which is precisely the subject of the next subsection, while the neural network configuration is described in the second subsection.

A. Extracted Features

The artificial neural network integrated in Dolores defines a statistical document model which requires a set of training samples. Thus, we defined a set of about 60 features used to extract our samples from the ground-truthed data (i.e., confirmed labeled text blocks):

Morphological features: text block min-x, min-y, max-x, max-y, center-x, center-y, width, height, width/height, text rotation, etc.

Structural features: font-size, font-style, interline, justification, number-of-tokens, number-of-lines, tokens-per-line, etc.

Syntactical features: percentage of caps, letters, numbers, symbols, spaces, etc.

Relational features: font sizes of adjacent blocks (N, E, S, W), widths of adjacent blocks, distance to adjacent images, widths of adjacent images, etc.

Dynamic features: they are dynamically generated during the labeling process when Dolores encounters some new text block attributes: font-name, font-size, font-color,

and background-color. The number of features therefore increases during the model acquisition.

Beside the aforementioned set of features, OCD Dolores offers the user to define customized features of two different types: region and regex features.

A **region feature** is a rectangular area interactively defined by the user. The returned value is the overlapping percentage between the bounding box of a text block and the rectangular region.

Similarly, a user can create a **regular expression feature** (or regex) by editing a text field using the Perl regex syntax. The value returned by a regex feature is the number of regular expression matches found in a text block.

B. Sugarcube Multilayer Perceptron

The document model of Dolores integrates what we call a "sugarcube multilayer perceptron" or SMLP. Actually, standard MLPs have some undesirable properties: they need manual configuration and fine tuning to perform efficiently. Moreover, they are black boxes: while the pure logic of an MLP is clearly defined and understood, it is not possible to grasp the meaning underlying the network decisions, i.e., the embedded knowledge of an ANN is far from explicit. We therefore developed a neural network solving such issues. The aim of an SMLP is hence twofold: first, facilitating the neural network configuration by removing any manual settings and, second, unveiling the black box of MLPs by providing an interactive visual feedback highlighting the relevance of input features relative to output classes.

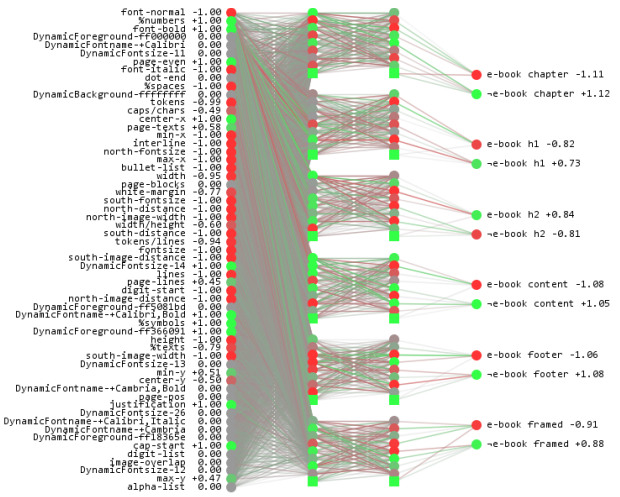

Figure 2. Topology of our SMLP with 6 output classes

Note that the following considerations result from two years of experimentations we did during the development phase of OCD Dolores. Things are exposed as is without deep explanation since the SMLP is the subject of an upcoming paper.

The SMLP topology is presented in Figure 2: one input layer fully connected to a first hidden layer, followed by partially connected hidden and output layers. The exclusive set of hidden neurons allocated to each output neuron allows an SMLP to behave as a set of distinct MLPs, each one of them dedicated to single label recognition. We found that such topology enables output neurons to develop their own decision boundaries in total freedom whereas frequent classes tend to overcome the whole network with standard MLPs.

SMLPs need very few learning cycles to achieve their training phase, a real advantage when working with responsive environments. Moreover, the proposed network topology is able to efficiently deal with very few training samples, which is also a key issue in an interactive environment where the training data is progressively integrated, i.e., the training set is very small at beginning.

Our SMLP implementation roughly follows the Duda-Hart "Pattern Recognition" book guidelines (normalization, weights initialization, learning momentum, weight decay, activation function, backpropagation training). However, parameters such as learning rate, number of hidden neurons and number of training cycles are dynamically set by our system (which is not the subject of this paper). The neural network thus performs in complete autonomy, hiding any learning intricacies from the user. Still, note that an SMLP tries to avoid overfitting by minimizing the number of learning cycles needed to efficiently recognize the ground-truthed data.

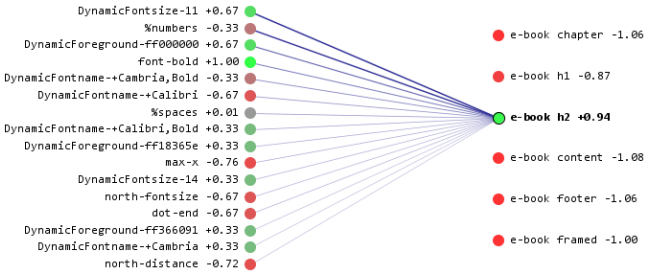

Figure 3. Most discriminating features relative to the h2 label

Last, but not least, the knowledge integrated in an SMLP is available through an interface allowing users to appreciate the discriminating power of each input feature by back propagating link weights relatively to a selected output neuron (see Figure 3).

V. EVALUATION

This section assesses the usability and efficiency of OCD Dolores through the evaluation of a set of concrete experiments. Unfortunately, standard evaluation protocols are of no use in our case, since existing offline learning systems evaluate their recognition performance thanks to batch processes performing cross-validation on groundtruthed data. Dolores can't apply such an evaluation protocol because it is based on an interactive learning system.

A specific protocol has therefore been defined to evaluate our interactive system. For this purpose, we use a set of five documents per evaluation; the first three documents are used to infer the document model, whereas the two last ones are

only used to appreciate its performance. The evaluation protocol adopts the following procedure: the first document is entirely integrated in the model through the interactive labeling; the recognition rate is computed after each user action and reported on a bar chart showing the evolution of the model relatively to the number of labeled objects, i.e., user actions. This basic model is then applied to the second document, the recognition rate is again reported, however, only the misclassified text blocks are labeled by the user and integrated in the document model, i.e., the document model is incremented. The same process is applied to the third document. Finally, the fourth and fifth documents are only used to observe the performance of the document model by looking at the classification matrices and recognition rates.

We evaluated OCD Dolores over five different document classes, going from very basic to quite complex layouts: Swiss TV Schedules, E-books, Wikipedia Articles, Scientific Papers, and Newspapers. For concision purpose, we only detail the E-books and Newspaper evaluations because of their significance (note that the 3 remaining evaluations also give very satisfying and comparable results).

A. *Inferring an E-book Model*

We downloaded five PDF e-books from planetpdf.com, each of them composed of parts and chapters : "Nostromo: A Tale of the Seaboard", "Crime and Punishment", "Madame Bovary", "Sons and Lovers", and "Anna Karenina".

Following our evaluation protocol, we labeled the first e-book; Figure 4 shows the evolution of the recognition rate relatively to the number of labeled objects. Note that a labeled object is the action of labeling a text block, or integrating a page/document (used when all the text blocks of a page/document are correctly recognized). One can observe that 13 labeling were enough to achieve a full recognition of the first e-book. The second e-book was then classified thanks to the previously inferred document model and resulted in a recognition rate of 99,98%. The classification matrix of Figure 5 shows that there was only one misclassified text block.

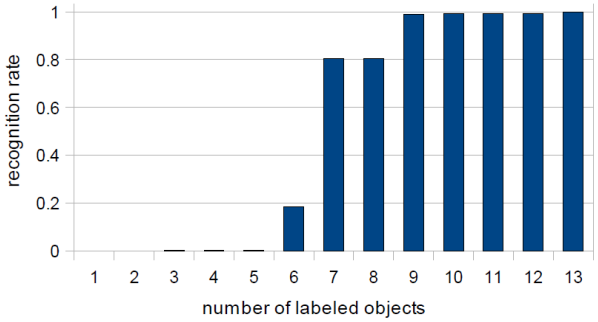

Figure 4. "Nostromo: A Tale of the Seaboard", all text blocks are recognized after 13 labeling actions.

Using the incremental learning facilities of Dolores, we simply labeled the misclassified text block to increment the existing document model. This updated model was then successfully applied on "Madame Bovary" since we got a recognition rate of 100%. We therefore left the document model unchanged and, again, applied it successfully on "Sons and Lovers" and "Anna Karenina" (recognition rate of 100% in both cases).

These e-book documents have a simple logical structure and a basic layout; moreover, they have low intra-class variability which explains the perfect recognition on untrained documents.

Label	Predicted x Actual Label								Total
title	1	0	0	0	0	0	0	0	1
author	0	1	0	0	1	0	0	0	2
advice	0	0	1	0	0	0	0	0	1
header	0	0	0	966	0	0	0	0	966
content	0	0	0	0	4071	0	0	0	4071
footer	0	0	0	0	0	966	0	0	966
part	0	0	0	0	0	0	8	0	8
chapter	0	0	0	0	0	0	0	41	41

Figure 5. Classification matrix of "Crime and punishment" using the model generated by "Nostromo: A Tale of the Seaboard", only one text block is misclassified, i.e., "author" is classified as "content".

B. *Inferring a Newspaper Model*

The newspaper class has a rich layout as well as a deep logical structures; it is therefore a case study class for a representative evaluation. We downloaded five consecutive PDF versions of the Swiss newspaper "La Liberté", from the 5th to the 9th of January 2009. "La Liberté" is composed of many different parts including advertisements, funerals, and stock exchange pages. Thus we decided to extract a predefined subset of pages from each newspaper consisting of both the International and National parts (i.e., about five pages for each publication).

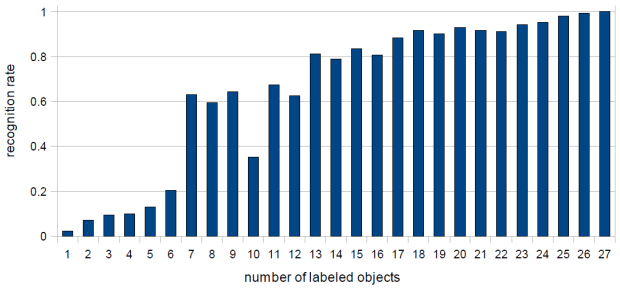

Figure 6. Recognition rate of "La Liberté" from the 5th january 2009 relatively to the number of labeling actions.

Figure 6 shows that the first newspaper publication was completely recognized after 27 labeling actions. Note that the action number 10 results in a sudden recognition drop. This is due to the fact that a new text block label was created, and, consequently, the system was faced with one more class to recognize (resulting in an increased source of recognition errors). This first document model version was then applied on the second newspaper in order to appreciate its accuracy. A recognition rate of 90.8% was achieved; Figure 7 shows the corresponding classification matrix.

Part of misclassified text blocks were hence integrated in an updated version of the model, i.e., incremented with the second newspaper. This model was then applied on the third newspaper; the recognition rate increased up to 96.0%. Once again, the document model was incremented with misclassified text blocks in order to obtain a better newspaper model. Finally, the resulting model was applied on the two remaining newspapers of our set. Recognition rates of 100% and 98.8% were obtained, respectively.

Label	Predicted x Actual Label													Total
page-nb	5	0	0	0	0	0	0	0	0	0	0	0	0	5
header	0	10	0	0	0	0	0	0	0	0	0	0	0	10
date	0	0	5	0	0	0	0	0	0	0	0	0	0	5
column	0	0	0	2	1	0	0	0	0	0	0	3	0	6
highlight	0	0	0	0	18	0	0	0	0	0	0	0	0	18
title	0	0	0	0	1	23	1	0	0	0	0	1	4	30
content	0	0	0	0	2	2	144	0	0	0	0	6	0	154
caption	0	0	0	0	0	0	0	7	0	0	0	0	0	7
subtitle	0	0	0	0	0	0	0	0	6	0	0	0	0	6
author	0	0	0	0	0	0	0	0	0	8	0	1	0	9
intercontent	0	0	0	0	0	0	0	0	0	0	1	0	0	1
advertisement	0	0	0	0	0	0	0	0	0	0	0	9	0	9
intertitle	0	0	0	0	3	0	0	0	0	0	0	0	10	13

Figure 7. Classification matrix of the second newspaper publication using the model generated by the first one gives a recognition of 90.8%.

We observed that the last newspaper had three misclassified text blocks: one “advertisement” text block was classified as “author” and two “caption” were classified as “page-nb”. These errors are easily interpretable; indeed, advertisements have irregular layouts using lots of various font sizes and styles, which may induce the document model in error. Furthermore, the two misclassified captions were quite unusual since only composed of single numbers, a property that is precisely used as a discriminant feature for the “page-nb” label class (see Figure 8).

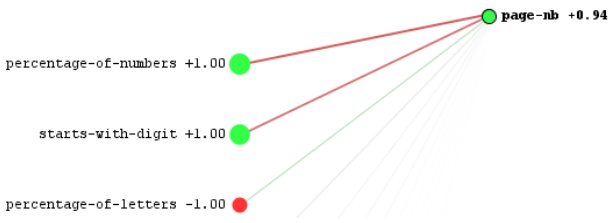

Figure 8. The most discriminant feature relatively to the page-nb class is the “percentage-of-numbers” followed by “starts-with-digit”.

Document Class Recognition

Until now, we applied document models presupposing that we knew the class of a given document. An exciting issue would now be to identify the model of a document given a predefined set of models. We did this by applying iteratively each model to a document; the one maximizing the recognition outputs of the SMLP weighted by the output class probabilities (inferred from the training set) being elected.

Hence, we applied our five inferred models on our 10 untrained documents. Figure 9 show that the model confidence is always the highest when corresponding to its document class. This experience is crucial because it proves that the class of a document can be identified given a set of predefined document models. Such results assesses the reality of fully automatic restructuring systems using predefined document models.

	TV Schedule	E-book	Wikipedia Article	Scientific Paper	Newspaper
TSI2 2005/09/21	**0.86**	0.18	0.26	0.23	0.06
TSR1 2005/09/21	**0.94**	0.15	0.30	0.23	0.06
Lawrence Sons and Lovers	0.41	**0.98**	0.11	0.28	0.07
Anna Karenina	0.41	**0.98**	0.11	0.31	0.09
Charles Breton	0.38	0.36	**0.96**	0.34	0.24
Albert Camus	0.39	0.30	**0.95**	0.32	0.23
The Organization and ...	0.41	0.53	0.52	**0.92**	0.45
Computing with Proteins	0.37	0.58	0.53	**0.93**	0.43
La Liberté 2009/01/08	0.36	0.37	0.54	0.47	**0.89**
La Liberté 2009/01/09	0.37	0.39	0.56	0.47	**0.86**

Figure 9. Document model recognition on untrained documents using 5 predefined models (TV Schedule, E-book, etc.).

VI. CONCLUSION

This paper presented OCD Dolores, an innovative system facilitating and speeding up the creation of document models thanks to an interactive and dynamic learning environment. Presented evaluations precisely emphasized the efficiency and relevance of our system; they clearly showed us to carry on this way. Two tools based on OCD Dolores will soon see the light: the first one able to reinject recovered structures into PDF files and the second one allowing PDF documents to be reflowed as electronic books (EPUB 3.0).

REFERENCES

[1] W. S. Lovegrove and D. F. Brailsford. “Document analysis of pdf files: methods, results and implications.” Electronic Publishing - Origination, Dissemination and Design, 8(3):207-220, 1995.

[2] A. Anjewierden and S. Kabel. “Automatic indexing of documents with ontologies.” In 13th Belgian/Dutch Conference on Artificial Intelligence (BNAIC 2001), pp. 23-30, Amsterdam, Holland, 2001.

[3] F. Rahman and H. Alam. “Conversion of pdf documents into html: a case study of document image analysis.” In Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers 2003, pp. 87-91, USA, 2003.

[4] H. Déjean and J.-L. Meunier. “A system for converting pdf documents into structured xml format.” In IAPR International Workshop on Document Analysis System (DAS'06), pp. 129-140, Nelson, New Zealand, 2006.

[5] F. Bapst, R. Brugger, A. Zramdini and Rolf Ingold. “Integrated mult-agent architecture for assisted recognition.” In Document Analysis Systems (DAS) pp. 172-188, Malver, Pennsylvania, October 1996.

[6] L. Robadey. “2(CREM) : Une méthode de reconnaissance structurelle de documents complexes baée sur des patterns bidimensionnels. PhD thesis, PhD Thesis, IIUFUniversité de Fribourg, Switzerland, 2001.

[7] O. Hitz, L. Robadey, and R. Ingold. “An architecture for editing document recognition results using xml.” In 4th IAPR International

[8] J.-L. Bloechle, D. Lalanne and R. Ingold. “OCD: An Optimized and Canonical Document Format.” In 10th International Conference on Document Analysis and Recognition (ICDAR’09), pp. 236-240, Barcelona, Spain, October 2009

[9] J.-L. Bloechle, M. Rigamonti, K. Hadjar, D. Lalanne and R. Ingold. “XCDF: A Canonical and Structured Document Format.” Document Analysis Systems (DAS’06), pp. 141-152, Nelson, New Zealand, 2006