Eyes Wide Open: an interactive learning method for the design of rule-based systems

Cérès Carton, Aurélie Lemaitre, Bertrand B. Coüasnon

International Journal on Document Analysis and Recognition manuscript No.
(will be inserted by the editor)

Eyes Wide Open: an interactive learning method for the design of rule-based systems

Cérès Carton, Aurélie Lemaitre and Bertrand Coüasnon

[1] IRISA-INSA, Campus de Beaulieu, Avenue du Général Leclerc, Rennes
[2] IRISA-Université de Rennes 2, Campus de Beaulieu, Avenue du Général Leclerc, Rennes

Received: date / Revised version: date

Abstract. We present in this paper a new general method, the Eyes Wide Open method (EWO) for the design of rule-based document recognition systems. Our contribution is to introduce a learning procedure, through machine learning techniques, in interaction with the user to design the recognition system. Therefore, and unlike many approaches that are manually designed, ours can easily adapt to a new type of documents while taking advantage of the expressiveness of rule-based systems and their ability to convey the hierarchical structure of a document. The EWO method is independent of any existing recognition system. An automatic analysis of an annotated corpus, guided by the user, is made to help the adaption of the recognition system to a new kind of document. The user will then bring sense to the automatically extracted information. In this paper, we validate EWO by producing two rule-based systems: one for the Maurdor international competition, on a heterogeneous corpus of documents, containing handwritten and printed documents, written in different languages and another one for the RIMES competition corpus, a homogeneous corpus of French handwritten business letters. On the RIMES corpus, our method allows an assisted design of a grammatical description that gives better results than all the previously proposed statistical systems.

1 Introduction

Document image analysis is the process of automatically extracting useful information from page images. This process can be performed using different classes of algorithms. Among these algorithms, we can distinguish two main classes: the *statistical* layout analysis and the *syntactical* layout analysis.

On the one hand, the statistical layout analysis methods are able to deal with noise and uncertainty, often present in document analysis [22]. However, they lack the ability to convey the hierarchical structure of a document. These methods can deal with local variations but their inference of the global structure is quite limited and the obtained inferred structure may not be interpretable. On the other hand, the syntactic methods describe the global structure of a document in grammatical rules, which are interpretable by a human. They have a strong expressiveness power, which allows them to convey the hierarchical structure of a document. They allow the introduction of knowledge in the grammatical description of the document but they do not benefit from an automatic learning process based on the data, such as the one present in statistical methods.

In the state of the art, document recognition systems are dedicated to documents that are getting *more and more complex* and for which the hierarchical organization of a document is of importance. It is why we think that rule-based methods are well adapted for document layout analysis. Unfortunately, the fact that no automatic learning process exists in syntactic methods makes them long and difficult to develop. When a new type of document needs to be recognized, the whole process of adaption of the recognition system must be redone (figure 1). It is a complex and time-consuming task as it is done manually in three steps. Step 1 (figure 1-(1)): a small sampling of documents is extracted from the whole learning data set and the user manually expresses knowledge based on this sampling and his a priori knowledge. Step 2 (figure 1-(2)): a document recognition system is produced with the knowledge manually extracted by the user and applied on the documents. Step 3 (figure 1-(3)): an evaluation of the produced system is done which makes a trial and error approach to improve the set of rules designed by the user. The three steps are then repeated which is also time-consuming. Moreover, at each step of the trial and error approach, the user doesn't know if he has detected all the possible cases present in the set of documents or not.

Furthermore, data set size is constantly growing and the documents in a same data set can be *heterogeneous*. Data sets can be composed of different types of documents. Documents can also be of the same type but present a variety of layouts, for example for invoices or pay slips. This heterogeneity increases the difficulty to create a good recognition system, as it is more and more

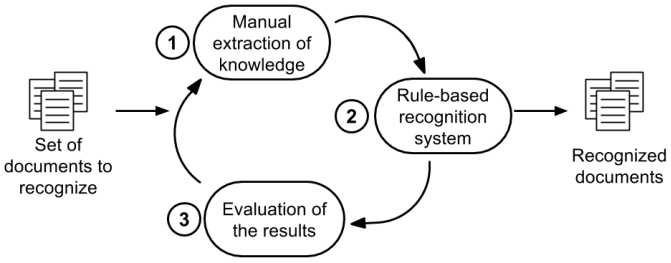


Fig. 1. Current design of a rule-based recognition system, based on manual extraction of knowledge combined with a trial and error approach

difficult to have a small sampling representative of the documents to analyze. It increases the difficulty of detecting all the cases present in a corpus of documents and then to be able to describe the appropriate document recognition system.

Therefore, we propose a new method, Eyes Wide Open (EWO), to introduce a learning process for the design of rule based systems. EWO can replace the manual extraction of knowledge. This method will help the user to start with a basic logical description of the documents and end with a complete grammatical description. With EWO, the logical structure of the document is completed and optimized while the physical structure is learned. This is possible due to three aspects of the method:

- *Automatic and exhaustive* analysis of an annotated data set,
- *Interaction* with the grammar writer,
- *Evaluation of the pertinence* of the built grammar.

Using EWO, we can have the advantages of the rule-based systems, especially their expressiveness, without one of their major drawback which is the time needed to adapt the system to a new type of document.

This paper is organized as follows. Section 2 discusses previous work on document recognition systems, but also on data clustering. Section 3 presents an overview of EWO method. Then, we describe how the rules inference is made progressively with the logical structure inference (section 4) and the physical structure inference (sections 5 and 6). Finally, we demonstrate in section 8 that EWO has been successfully used on more than 2,000 documents of two international competitions data sets.

2 Related work

2.1 Document recognition systems

We present here the advantages and drawbacks of both statistical and syntactical methods for document image structure analysis.

2.1.1 Statistical methods Statistical methods for document structure analysis are based on different formalisms, such as 2D Markov Random Fields [18] or Conditional Random Fields [24]. A few contributions using classical machine-learning tool can also be found in the literature. For example, Rangoni [26] uses a dynamic perceptive neural network.

Lemaitre [18] proposes a method based on the Markov Random Field, with a pixel-level labeling. Different labels can then coexist in the same region which is not theoretically possible as the global consistency of the block labeling is not assured. The authors underline that errors are due to the lack of global information in the analysis (at block level for example).

Montreuil [24] presents a method where a hierarchical combination of Conditional Random Fields is made. The physical layout segmentation is done with three levels of CRF and the final step consists in segmenting and labeling the blocks with a CRF model. The authors point out that the integration of textual information allows the good results obtained, as blocks with different labels can have the same structural features. They analyze errors in their model due to error accumulation at the different levels of segmentation. They underline that the main advantage of their method is the *use of a training procedure*, which takes into account the variability of the documents to analyze. This shows the crucial importance to introduce a training phase to obtain an efficient system, easily adaptable to a new kind of documents.

Statistical methods allow the integration of noise and uncertainty, which are often present in the documents to analyze. However, the statistical methods usually are *not able to convey the hierarchical structure of the documents* which is needed to deal with complex documents such as tables, mathematical expressions or charts. They can deal with local variations but their inference of the global structure is limited. For example, the spatial inter-dependencies that can be modeled with a CRF are limited to small portions of the space. Shetty [27] models spatial and logical inter-dependencies between neighboring patches, where a patch is approximately the size of a word. However, to correctly convey the hierarchical structure, an ideal method must also be able to model spatial inter-dependencies between elements like paragraphs, titles, images, etc.

We think that there are advantages using statistical methods, in particular their ability to adapt to a new type of documents through a learning process of the data set. However, they lack the ability to convey the global hierarchy of a document. Moreover, it is difficult to integrate user knowledge in the system and it is often not interpretable by a human user. These aspects are in particular treated by the syntactical methods.

2.1.2 Syntactical methods To correctly segment and recognize the documents, we need a representation of the knowledge, a model, to interpret the input data. The model contains all the information to transform a physical structure into a logical one. Rule-based systems have been proposed to do so as [11], [19]. However, they are poorly flexible and can become rather arbitrary. To allow a more precise description of the relation between elements, these methods may incorporate notions based on formal grammars.

Grammatical systems allow a more precise semantic description for the relations between elements. Using grammatical approaches, the image is segmented in primitives and the user builds a rule tree that describes how to compose these primitives. The build rule tree allows a natural expression of recursive structure such as hierarchical ones. Conway [3] proposed a bottom-up parsing method based on a page grammar. Krishnamoorthy [15] proposed a document logical structure recognition method that recursively applies grammar to horizontal and vertical projection profiles of the page. Coüasnon [4] proposed a generic grammatical framework, DMOS-P. The DMOS-P parser uses a 2D-grammar and performs a top-down analysis with backtrack.

Grammatical systems, like rule-based systems, are *understandable by a human* and allow the *integration of user knowledge*. Furthermore, we can increase the flexibility of these methods by *integrating statistical information* [23], obtained for example from recognizers, during syntactic layout analysis. Stochastic grammar on 1D-grammars have also been proposed in the literature [30] [21]. We do not develop this aspect as it does not prevent from designing the grammar which is the issue we are interested in here.

One of the major drawbacks of the grammatical approaches is that they do not allow an automatic learning of the document recognition system. The adaptation to a new type of documents is then costly. Our goal is to create a single framework which can be *rapidly applied to new domains*, with a high confidence that the resulting system will be efficient and reliable. This is in contrast to a number of previous systems, where retargeting requires a long process to describe the logical structure and to hand tune many parameters. The current methods lack a learning process to make easier the design of a new system.

2.1.3 Learning grammars The challenges of grammatical approaches include computational complexity, grammar designs, feature selection, and parameter estimation. Few methods have been proposed to automatically design grammatical descriptions.

Shilman [28] presents a method to learn non-generative grammatical models for document analysis. They focus their effort on feature selection and parameter estimation. The user needs to specify the page grammar and to provide a set of correctly labeled pages. This method allows automating a part of the grammar writing, making it easier and faster. It is a first interesting step in the building of a machine learning step for grammar systems. However, the user is not helped for the grammar design which is, in our opinion, not a trivial task, especially since the data set are more and more complex and heterogeneous.

Grammatical inference has been studied in numerous fields [5], however our task presents characteristics that make our problem difficult to solve. Grammar induction method have been mostly develop for one dimensional grammar whereas we need to deal with bidimensional grammars to be able to efficiently describe document structures. Moreover, grammar induction methods are not robust to noisy data. An efficient document recognition system must be able to deal with noisy images. Finally, it is difficult in most grammar induction methods to combine the grammar induction with other techniques or prior knowledge. All these difficulties inherent to the document structure recognition problem make it too difficult to be solved completely automatically. We then propose to introduce an interaction with the user.

To obtain a generic method for the analysis of documents, we decide to use syntactic method as it presents the ability to convey the hierarchy of the documents that statistical methods cannot handle. Furthermore, it is possible to integrate statistical information as it has successfully been shown by Maroneze [23]. To make it easier to adapt to a new type of documents, we propose to assist the grammar description by a training phase that will give an exhaustive view on the data set and therefore allow a good understanding of the data. To do so, we will use in particular *data clustering techniques*.

2.2 Data clustering

Automatic extraction of knowledge is a great challenge. The amount and the diversity of the data make it more and more difficult. One of the major techniques of the extraction of knowledge is data clustering. The goal of data clustering is to discover the natural grouping(s) of a set of objects. Our goal is to gain insight into data, detect anomalies, and identify salient features. To discover the natural grouping we need to define a similarity measure between observations, which is not easy to specify without any prior knowledge about cluster shapes.

An ideal cluster can be defined as a set of points that is *compact*, the objects in a same cluster are similar, and *isolated*, the objects of a cluster are completely different from the objects of the other clusters. In reality, a cluster is a subjective entity, and which significance and interpretation requires domain knowledge [6]. It is possible for a human to detect clusters in two and possibly three dimensions but we need automatic algorithms for higher dimensions, as for not well separated data. Additionally, quantitative evaluation of the quality of clustering results is difficult due to the subjective notion of clustering.

A large number of clustering algorithms exist. Jain and al. [12] divide the algorithms into two groups: *hierarchical* and *partitional*. Hierarchical clustering algorithms recursively find nested clusters while partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. No single algorithm is able to identify all sorts of cluster shapes and structures that are encountered in practice. Each algorithm defines a representation of the data which is closely tied with the purpose of grouping. The representation must be chosen in function of the data and the goal of the user. However, in our case, we must be able to fit with various configurations without changing the clustering algorithm.

Among the various clustering methods, the K-Means algorithm, which minimizes the squared-error criteria, is one of the simplest algorithms. It is computationally efficient and does not require specifying many parameters. This algorithm has been well studied [13] and a lot of extension have been proposed, for example use of Mahalanobis distance [20], of L1 distance, or introduction of the fuzzy set theory to obtain non exclusive partitions [25]. Its major limitation, however, is the inability to identify clusters with arbitrary shapes. Furthermore, one of the input parameter is the number of clusters that we do not know a priori.

Automatically determining the number of clusters K has been one of the most difficult problems in data clustering. Usually, clustering algorithms are run with different values of K; the best value of K is then chosen based on a predefined criterion. As example of criterion we can cite gap statistics [31], silhouette statistics [14] or Akaike information criterion and Bayesian information criterion [29]. Despite of the range of existing criteria, it is not easy to decide which value of K leads to the most meaningful clusters.

While hundreds of clustering algorithms exist, it is difficult to find a single clustering algorithm that can handle all types of cluster shapes and sizes or even decide which algorithm would be the best for a particular data set [7]. When there is a good match between the model and the data, good partitions are obtained. Since the structure of the data is not known a priori, in general we need to try diverse algorithms to determine an appropriate one for the clustering task.

One recent development of clustering has been the ensemble methods [16]. The basic idea is that by taking *multiple looks* at the same data, we can generate multiple partitions (*clustering ensemble*) of the same data. By combining the resulting partitions, it is possible to obtain a good data partitioning even when the clusters are not compact and well separated. The evidence accumulation step that combines the information provided by the different partitions can be viewed as learning the similarity measure among the data points. A complete method was presented by Fred and Jain as the Evidence Accumulation Clustering (EAC) [8]. For the clustering ensemble generation, they suggest that any clustering algorithm can be used. They conducted the experiments using K-means with random initialization and random selection for K. Kuncheva and Todorova [16] define the best ensemble methods use K-means for the individual clusters. The combined clusters are obtained using a single or average linkage method. The maximum lifetime criterion is used to determine the number of clusters.

2.3 Philosophy of our approach

By introducing a training process in rule-based recognition systems, we can obtain a system that can deal with complex hierarchy and be understandable by a human. The training process provides an extensive overview on the data allowed by the use of the EAC clustering combined with an interaction with the user.

We want the EWO method to be generic and to easily adapt to any new data set. It is difficult to determine the best clustering algorithm to fit the data as we have no a priori. We therefore decide to use the EAC clustering that adapts well to various data sets as its clusters can have an arbitrary shape. Furthermore, the number of clusters is automatically determined with EAC. Thus, we do not need any parameters.

3 EWO method: overview

The Eyes Wide Open method has been designed to extract knowledge from an annotated corpus of documents in order to build a grammatical description in cooperation with the user. The EWO method infers both the logical and the physical structures of the documents. Our method is independent of any existing syntactical recognition system, and therefore can be used in cooperation with any system. The user gives as an input a set of annotated documents representative of the documents that need to be recognized. Each element of a document that need to be recognized must be at least annotated by its label and the position of its bounding box.

Figure 2 presents the overall functioning of our method to introduce a learning process in the design of a grammatical description. The user builds a very first basic logical description of the documents to recognize: he specifies the types of elements that need to be recognized (generally matching the labels present in the ground truth). This simple initial description of the logical structure will then be extended and completed with EWO, in an interactive learning process with the user, to describe *how* the documents will be recognized.

The *logical structure* is completed through an automatic discovering of the logical structure variants of the grammatical description (cf. section 4). Each logical structure variant is then combined with the *physical structure* of the documents which is learned through the inference of the position operator (cf. section 5) and the inference of the physical properties for each rule (cf. section 6). Then the logical and physical descriptions are confronted to study the possible confusions and limit them (cf. section 7). Each block of EWO is described in the following sections of this paper. To illustrate it, we now introduce an example based on the RIMES data set that is used in the following sections.

Example: We take the example of a user who wants to recognize French handwritten business letters (figure 5). To do so, he defines a first basic logical structure of the letters (figure 3): a letter is a document composed of eight different types of elements (that can be present or absent in each letter). As usual, in grammatical systems, the user must manually define the content of each of the eight rules, corresponding to each element: sender details, date and place... The novelty of the EWO method is to provide an assistance for the generation of the grammatical description of the eight rules. In this article we will present how each rule can be automatically completed with the logical variants and the physical structure in order to be able to segment and recognize the elements.

4 Logical structure inference

To complete and extend the first very basic description made by the user (figure 3), the EWO method enables to infer the logical structure. Indeed, each rule can be composed of one or more versions. If different general cases exist in the documents, then different versions of

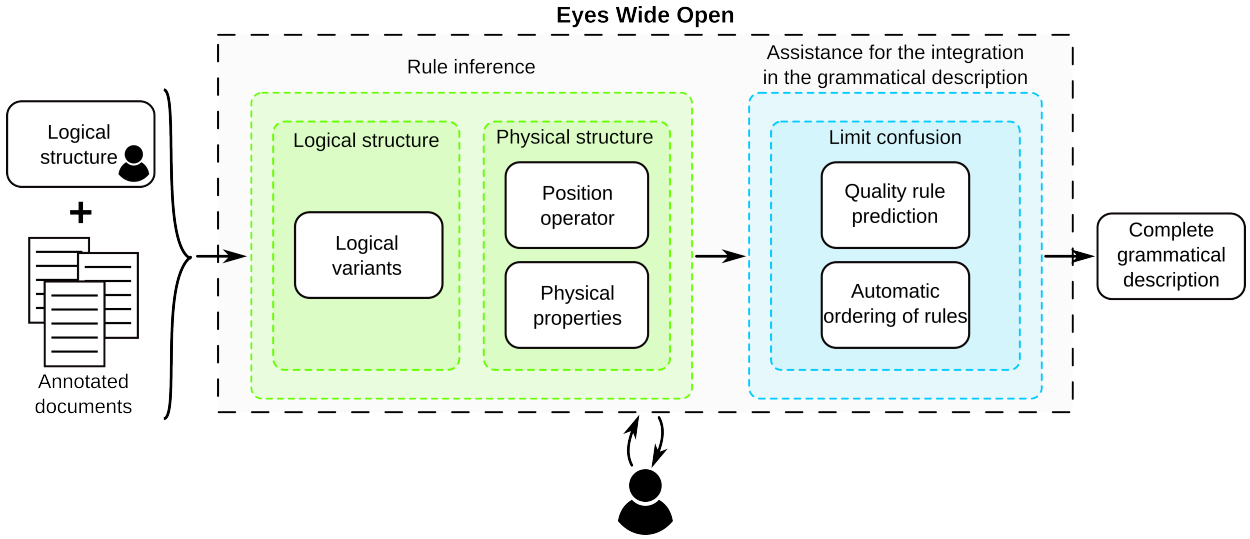


Fig. 2. Overview of EWO method: it enables the design of a grammatical description with (1) a rule inference module that infers a logical and a physical structure of the rules, and (2) a module that provides assistance for the integration in the grammatical description in order to limit the confusion during the analysis.

```
letter ::=
   senderDetails &&
   recipientDetails &&
   datePlace &&
   object &&
   opening &&
   textSection &&
   signature &&
   PS.
```

Fig. 3. First logical description of a french handwritten letter

the rule must be described. It is the first level of hierarchy of the grammar.

Finding this hierarchy is then crucial for the grammar writing but it is not an easy task. To do so, the user must have a *global overview* of the documents. When the user does a manual extraction of knowledge, only a small sampling of the documents is analyzed. It is not sufficient to be able to find the first level of the hierarchy. It is why our method offers an exhaustive analysis of all the documents of the training data set.

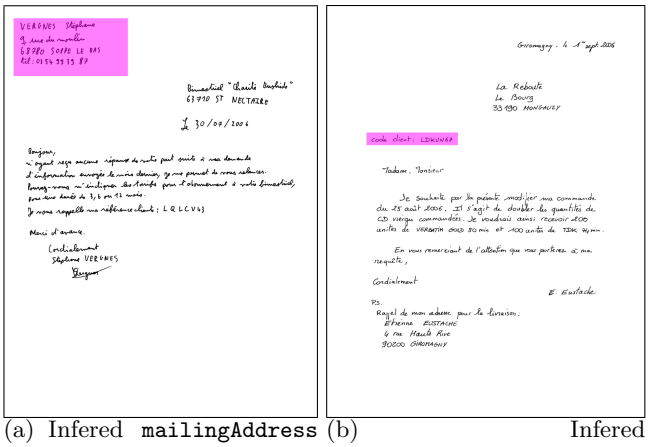
To infer the logical structure, an unsupervised clustering of the data is made in EWO, as we want to find existing groups that are not directly annotated in the ground truth. To do so, we use the EAC Clustering presented in section 2.2 [8]. As an input, we need a labeled data, corresponding to the elements that must be recognized. If the ground-truth is not available, we must build it. In the examples of this paper, the ground-truth was provided with the data-sets of the competition, so we could use it without any preprocessing. The ground truth contains the bounding boxes of the segmented text blocks. Each bounding box is labelled with the logical function of the element (`senderDetails`, `dateAndPlace`...). The clustering is made on the physical properties of the boxes that have the same label. The EAC clustering adapts well to any new corpus of documents and automatically detects the number of clusters to find. Then, the best parameters for the clustering algorithm can be automatically detected. Each detected cluster can be used to define a different version of the grammatical rule by the grammar writer through an interactive step. The grammar writer visualizes some representative elements of the cluster to validate its relevance and then name it to bring semantics to the automatically inferred logical structure.

To create the clustering ensemble, we use several times the K-means algorithm with a random initialization for K at each step on all the training data set. The K-means algorithm has been proved efficient for the clustering ensemble and it has the advantage to be computationally efficient. For the determination of the combined clustering, we use the average link method as it is more robust. In general, it produces better results especially in situations of touching clusters.

Example: In the example of the handwritten business letters, the first searched element is `senderDetails`. The EWO method automatically detects two different clusters, using the height and the width of the bounding boxes of the `senderDetails` labeled in the ground-truth (figure 4). The user visualizes few representative examples of the two clusters and labels them to bring sense to the data. Here, the user labels cluster 1 as `mailingAddress` and cluster 2 as `clientIdentifier`. The inference of the rule is continued with the physical inference.

The specific case of the outliers

In our analysis, we provide an exhaustive view on the data, including the outliers. Extreme values can come either from an atypical value or from a ground truth error. We then decide not to include them in the general analysis, as they have in general a strong impact on the statistical indicators and on the shape of the clusters.



(a) Infered `mailingAddress` logical variant

(b) Infered `clientIdentifier` logical variant

Fig. 4. Example of logical structure inference for `senderDetails` elements in ground truth.

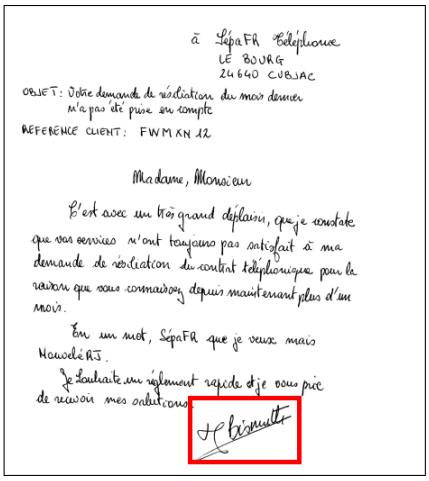


Fig. 5. Automatic outlier detection: example of an outlier annotated as `textSection` instead of `signature` due to a ground truth error

However, they are important information for the grammar writer and the EWO method detects and displays them to him.

We propose to detect the outliers with a very classical method, recommended for example in this handbook of statistical analysis [10] : considering that all points that do not belong to $[mean - t \times SD; mean + t \times SD]$ are outliers, where SD is the standard deviation of the observed distribution. As also recommended in [10], we use very classical values for t, $t = 3$ if the number of observation is greater than 80, otherwise $t = 2.5$. This technique is simple to use and well known not to detect all the outliers. We then limit the risk to delete interesting values.

As a good side effect of the outliers detection, Eyes Wide Open is able to detect a wide range of annotation errors in the annotated ground truth. Annotated ground truth can be made partly or completely manually. As all processes, errors may happen. It is very interesting to be able to detect these errors which will improve the learning and the rules but will also allow a correct evaluation of the results. Figure 5 presents an example of an automatically detected outlier. It is a ground truth error: the signature was labeled as `textSection`, instead of `signature`.

5 Physical structure learning: position operator

For the description of a bi-dimensional structure of documents, we need to define some position operators that express the zone of the image that is analyzed. A given element can be found in one or more zones of the page. Each positioning variant will lead to a position operator.

We use the representation of a position operator (figure 6) as a rectangle defined by two points: $A(X_a, Y_a)$ and $B(X_b, Y_b)$. We add a third point $V(X_v, Y_v)$ called the point of view. The point of view gives an orientation to the analysis: inside of the zone of the position operator, the elements are analyzed from the closest to

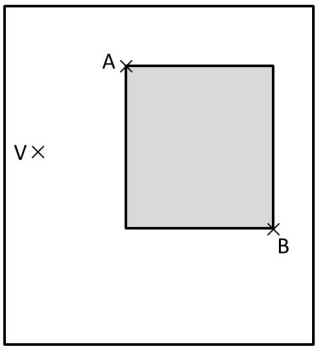
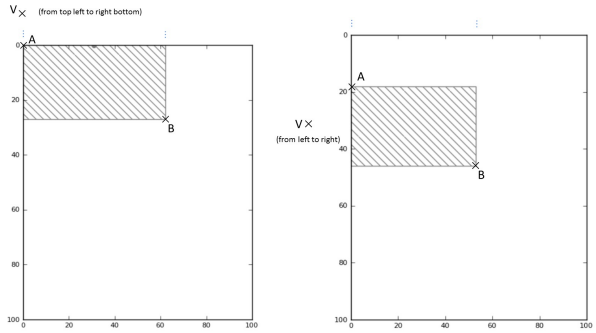


Fig. 6. Position operator: points A and B define the research zone in the image. V is the point of view, here on the left so the elements will be analyzed from left to right

the furthest from the point of view. The point of view is chosen to minimize the confusion, i.e. the risk to mix up the searched element with an other one, so that we first analyze the elements that have the best chance to be the one we are looking for.

We developed a learning system, LearnPos, presented in [2], for the automatic inference of the position operators. With LearnPos, two types of positioning can be considered: absolute positioning and relative positioning. The absolute positioning gives the position of an element *in the page*. For example, we can express that the signature is at the bottom of the page in a letter. The relative positioning gives the position of an element *in function of another one*. For example, the text section is located under the opening. The choice of absolute or relative positioning is determined by the grammar writer. The learning process of position operators is:

1. The user explicitly asks for the positioning *Pos* of a component. He only specifies the type of *Pos*: absolute or relative
2. The system detects the different groups on the histogram of bounding boxes coordinates and infers as



(a) Pos_1: Mailing addresses (b) Pos_2: Client identifiers

```
mailingAddressZone :-
   Xa = 0,
   Ya = 0,
   Xb = 62% image width,
   Yb = 27% image height,
   Xv = 0,
   Yv = -100.
```

```
clientIdentifierZone :-
   Xa = 0,
   Ya = 18% image height,
   Xb = 55% image width,
   Yb = 47% image height,
   Xv = -100,
   Yv = 32% image height.
```

(c) Associated inferred code : coordinates of A,B,V

Fig. 7. Automatic inference of position operators for `senderDetails` elements. Two groups of positions are automatically detected, Pos_1 and Pos_2. The semantic name of the positions is provided by the user.

many position operators as necessary $(Pos_1, ..., Pos_n)$.

3. For each Pos_i, the system automatically computes the zone coordinates A_i, B_i and the best point of view V_i.
4. The generated $(Pos_1, ..., Pos_n)$ are presented to the user who validates each position Pos_i.

The order of the position operators and the point of view are optimized to minimize the confusion. The point of view is inferred from a set of nine predetermined position (center, from top to bottom, from left to right, etc.). Our method tests each of the nine possible points of view and selects the one that minimizes the confusion. With EWO the position operators are now automatically learned instead of being manually defined, which avoids to manually define six values for each position operator. Further details on the inference of the position operator are described in [2].

*Example:*We compute the absolute position operator of the `senderDetails`. By automatic inference, the LearnPos system detects that there are two variants of the position Pos_1 and Pos_2. They are presented to the user (figure 7(a) and 7(b)) who brings its knowledge by naming the zones "mailing address zone" and "client identifier zone". He then obtains the associated parameters of A(X_a, Y_a), B(X_b, Y_b) and V(X_v, Y_v) that were automatically computed (figure 7(c)).

```
senderMailingAddress ::=
   AT(upperLeftCorner) &&
   0% <= height <= 32% page height &&
   0% <= width <= 50% page width.
```

Fig. 8. Sender mailing address rule: physical properties

6 Physical structure learning: physical properties

6.1 Estimation of the physical properties

In a grammatical description, the physical properties of the elements that need to be recognized must be described to be able to correctly segment the document image. Numerous parameters on the physical properties of the elements need to be manually determined which is long and difficult. For example, for the sender details, we can use the following properties: text block width, text block height, specific vocabulary, etc.

The properties that we are looking for are the own specific properties of the rule we want to describe. We are not in the case of a classifier where we need to find discriminating features. It is why EWO uses descriptive statistics for the available variables of the ground truth for each rule version to learn the physical properties. The overall view on the documents and the division of a rule in homogeneous and meaningful version allows the relevance of the inferred properties. In particular, we use the interquartile range descriptive statistics considering as the "normal" values the values in the interval: $[Q1 - 1.5 \times (Q3 - Q1); Q3 + 1.5 \times (Q3 - Q1)]$. The advantage of this criterion is that we make no assumption on the data distribution.

Example: If we take the example of the sender mailing address, thanks to EWO we obtain automatically inferred properties presented in figure 8.

6.2 Automatic ground truth enrichment

Using this approach, we observed segmentation problems. The level of details of the annotated ground truth is coarse: we generally only have the bounding box of the elements. That means that during the analysis in EWO all the segmentation problems are hidden as we directly use the segmented elements to learn the properties.

With the grammar rule described in the section 6.1, the example presented in figure 9 was segmented and recognized as a `senderMailingAddress`. For a human user, it is obvious that the last line is not a part of the same text block than the four first lines. The terminals of the grammatical description in the letter grammar are the *lines*, whereas the ground truth is made of the *bounding boxes* of the logical elements (the text blocks). Thus, we do not have in the ground-truth the required information to write the adapted rules to accurately segment the elements.

To improve the segmentation, we need to obtain the properties from the actual leaf elements of the grammar.

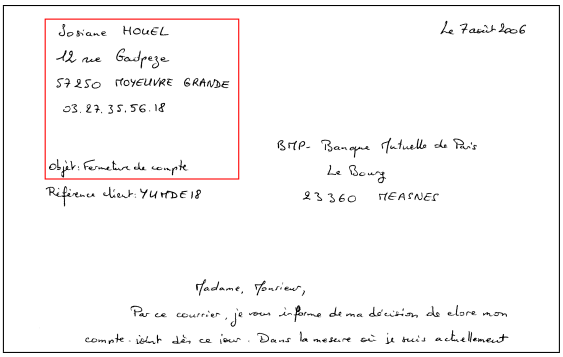


Fig. 9. Example of badly recognized sender mailing address due to a bad text block segmentation

To do so, we propose to *enrich the ground truth* by integrating complementary computed information to the manually annotated ground truth. For example, those complementary elements can be automatically produced by a recognition system that is able to produce the leaves of the analysis. Any complementary information can be integrated to the annotated ground truth, as long as its position in the document is known. Each complementary element is assigned to an annotated zone of the ground truth using their respective positions. We can then obtain more precise properties than the one contained in the annotated ground truth. This new knowledge has a cost: its uncertainty, as these data have not been obtained by a ground truth but automatically and have not been validated.

The properties that can be inferred from the ground truth enrichment are various as they only depend on the information precision given by the user. When the user wants to learn the physical properties, he has to indicate which type of elements must be used. In EWO, the user can ask questions that will lead to the automatic inference of a parameter. The user can for example use the "text line" information to learn what are the acceptable variations on the line spacing in a sender details block.

In general, the user has the ability to understand why the segmentation fails but it is difficult for him to estimate a threshold between acceptable and non acceptable values. If the threshold is too tolerant, a lot of segmentation errors will remain. On the opposite, if the threshold is too strict, a lot of elements will not be completely recognized. This balance here is not fixed after multiple trial and error cycles but automatically by using the real values of the learning data set. With these parameters obtained thanks to the questions of the user, the segmentation can be improved while reducing the time needed to write the grammar.

Example: To improve the segmentation of the sender mailing address, we add automatically inferred properties on the text line using the ground truth enrichment (figure 10).

The complete rule was inferred by successive steps: logical inference, position operator inference and physical properties learning before and after the ground truth

```
senderMailingAddress ::=
  AT(top left) &&
  2  to 5 lines &&
  maximum variation for line spacing= 115px &&
  maximum variation for line alignement= 226px &&
  0% <= height <= 32% page height &&
  0% <= width <= 50% page width.
```

Fig. 10. Sender mailing address enriched rule

enrichment. For that purpose, the user does not need to fix any free parameter. Indeed, the parameters of the method are restricted to the EAC clustering algorithm where they are chosen randomly and the different partitions combined to automatically determine the final partition (see section 4).

7 Assistance for the integration in the grammatical description

When designing a grammar, the grammar writer has searched the own properties of each type of element. This description can lead to confusion with other elements of the documents that have similar properties. With the classical manual approach of the grammar writing, the trial and error approach is used to find the confusion which means that the user has to:

- use the recognition system ;
- use a metric, if available, to evaluate the obtained result and then find the confusions.

These two steps are time consuming. However, to obtain a fully functional grammar, the user has no other choice than reduce the confusion. To do so, the user may modify each rule or determine a convenient order for the rules.

7.1 Prediction of the quality of the rules

We try to decrease the time needed to adjust the rules. In EWO, we integrate a system to simulate the behavior of the grammatical description outside of the recognition system to speed up the design of the system. To do so, the rules are approximated as requests. These requests are then applied on the annotated ground truth to obtain a simulation of the result of the grammar. We can then spot the confusion between elements. The rules can be changed by the user to minimize the confusion without using the recognition system.

With this method, the time needed to obtain these results is significantly decreased as EWO provides the results of a request in few seconds, compare to the duration of computing the system on for example 1000 pages at each request. Moreover, the user can easily adapt the grammar rules depending on its objectives: precision, recall or both.

7.2 Automatic ordering of the rules

An other way to reduce confusion is to find an appropriate ranking of the rules. When an element is detected

in the recognition system and a label produced, the element is consumed. It will not be available anymore in the analysis except if the analysis backtracks to explore another solution. It is why we must detect first the elements where we have a good precision to minimize confusion. The ordering of the rules is of crucial importance for the global performance of the grammar.

In EWO, when the user is satisfied with the rules he has designed, he can ask for an automatic ordering of the rules. To do this automatic ordering of the rules, we propose the following algorithm:

1. Apply each rule to the available elements
2. Compute precision
3. Select rule with the best precision
4. Consume the corresponding elements
5. Go back to step 1

This automatic ordering of the rules allows the user to automatically optimize the logical structure of his grammatical description by minimizing the possible confusions. This is possible thanks to the rules approximated as requests as presented in section 7.1.

8 Experimental validation

To evaluate the efficiency of our approach, we used the EWO method for the design of grammatical descriptions for two data sets used for open international competitions, the RIMES data set and the Maurdor data set. The RIMES data set is a homogeneous data set of French handwritten business letters whereas the Maurdor data set is a heterogeneous data set of complex documents.

8.1 Existing DMOS method

Eyes Wide Open is used to extract knowledge in a corpus of annotated documents for the design of a rule-based document recognition system. Here, we used the DMOS (Description and MOdification of the Segmentation) method [4] which is a grammatical method for structured document recognition. The DMOS method does not propose a machine learning step to create a new document description. The grammar design and the parameter estimation are both done manually.

This grammatical method is used to illustrate the efficiency of EWO. As it has been presented in section 3 EWO is independent in its implementation of the DMOS method and can be used with another rule-based recognition system. It can also be used for a specific hand-coded recognition system.

8.2 RIMES: an example of a homogeneous data set

The RIMES international competition [9] established a publicly available database containing pages of 5,605 of handwritten letters and faxes. These documents can be used for different tasks related to document recognition: layout analysis, writer identification, handwritten text recognition, etc. Images are 300 dpi gray scale scanned pages. All the images have been manually annotated with a ground truth.

The document structure recognition task in RIMES consists in the identification of up to eight different zones in each image and their assignment to one of the following labels: *sender details* (return address), *recipient details* (inside address), *date/place*, *subject*, *opening*, *message body*, *signature* and *attachment/postscript*. The data set for this task is composed of 1250 French letters. To compare the results obtained by our recognition system to the ground truth, we use the metric proposed by the RIMES committee. It consists in comparing the labels of each black pixel in both hypothesis and ground truth. This metric corresponds to the pixel error rate defined by the sum of all documents image pixels.

8.3 Position operator evaluation

We remind here the results obtained in [2], validating the efficiency of the automatic inference of the position operators. A grammatical description for the RIMES evaluation campaign already existed, this grammar was presented by Lemaitre [17]. In this existing grammar, position operator parameters were manually defined. We generated position operators with Eyes Wide Open for this grammar and introduced our inferred position operators in the existing deterministic grammar. The objective was to check that these automatically defined position operators are correct.

The learning data set is composed of the same 300 images for the two grammatical descriptions. As it can be seen in table 1, using EWO to learn the position operators we decreased the number of manual parameters by 40% while slightly increasing the performances.

	Existing grammar [17]	Our proposal using EWO [2]
Nb of manual parameters	102	66
Nb of automatic parameters	0	48
Global error rate	11.34	9.78

Table 1. EWO decreases the number of manually defined position operators in a grammar while decreasing the global error rate (RIMES competition database - 100 images).

In the previous grammar propose by Lemaitre [17], some position operators were manually and intuitively defined by the user, and it was not possible to learn them with EWO. In order to overcome this limitation, we propose in the next experiment to build a complete grammar with EWO, and therefore infer knowledge.

8.4 Defining a complete grammar with EWO

A complete grammatical description was created using EWO for handwritten letter recognition. We used 900

letters for the learning database, 250 letters for the validation set and 100 letters for the test set, corresponding to the test set of the competition. We can then compare our system with the other published systems. As we presented in section 3, the first simple logical description defined one rule for each of the eight types of zone, ordered using the human reading order of the documents. From this elementary logical description, a complete grammatical description was designed. EWO has been used to determine the rules through the logical and physical structures inference. Fig. 11 shows the final ordered grammar produced with EWO.

```
letter ::=
   opening &&
   textSection &&
   signature &&
   senderMailingAddress &&
   recipientDetails &&
   ps &&
   datePlace &&
   object &&
   clientIdentifier.
```

Fig. 11. Final ordered grammar for handwritten letter recognition

In this grammar defined with EWO, 34 different rules and variants were inferred. For the physical structure inference, we inferred 14 position operators. We also inferred 60 parameters for the physical properties inference. The results presented in table 2 show the error rates of our system and four other systems:

- A DMOS system (Description and Modification of Segmentation) with a a purely syntactical approach [17] (Sys1)
- A DMOS system with an approach combining content recognition and structure analysis [23] (Sys2)
- A Markovian Random Field (MRF) approach using structural and spatial features, completed by a post processing [18] (Sys3)
- A CRF based approach combining a CRF model to split the document the image and another CRF to assign a block label to each line [24] (Sys4)

DMOS [17] (Sys1)	DMOS stochastic [23] (Sys2)	MRF [18] (Sys3)	CRF [24] (Sys4)	Our system EWO
8.97	5.53	8.53	6.33	5.82

Table 2. Error rates obtained on the RIMES competition database test set (100 documents) for the layout analysis task

Table 2 shows that our model gives comparable results to the ones obtained by the best statistical system (Sys4) and to the best syntactical system manually learned (Sys1). This can be explained by the fact that it is a structural approach that was designed through a statistical approach that gave an extensive knowledge on the database. Our models used a handwritten word recognizer for the opening detection, like [23] approach, as some blocks with different labels can have the same structural properties. Table 3 confirms the good performances of our method on the learning (900 documents) and validation (250 documents) data sets.

Data set	Learning	Validation	Test
Size	900	250	100
Error rate	5.7	5.3	5.8

Table 3. Error rates obtained on the three sets of the RIMES competition database for our system with EWO

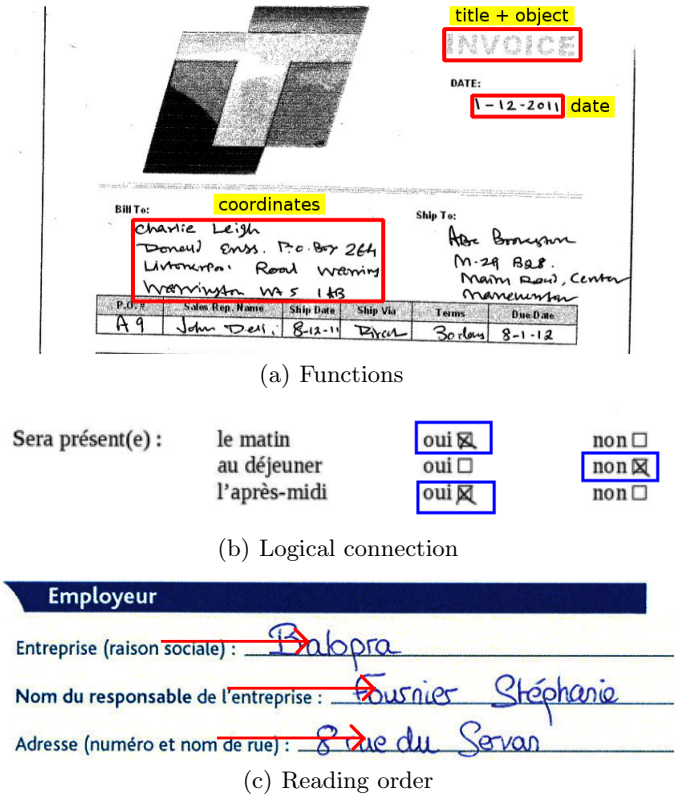
8.5 Maurdor: an example of a heterogeneous data set

Eyes Wide Open has also been validated on the MAURDOR competition [1] during the evaluation campaign. One of the specifities of the MAURDOR evaluation campaign is that it relies on a very heterogeneous corpus of documents. The whole corpus for MAURDOR 2013 comprises a total of 8,129 documents in English, French and Arabic. It various categories of documents: blank or completed forms, printed and manually annotated business documents or correspondence, private and handwritten correspondence sometimes with printed letterheads, and oher documents (newspaper articles, blueprints, etc.).

The systems are evaluated on a 1,000-document corpus following the training corpus proportions. With EWO, we built a grammatical description to participate to the Maurdor international competition on the task 5, dedicated to the extraction of the logical structure. This task consists in the determination of three types of information: (1) functions of each text area. A text area can have 1 to 4 functions; (2) logical connections between semantic areas (for instance, the connection between a check box and the text area which is associated); (3) reading order for the various areas (for instance, a column sequence in a press article).

The metric used for the MAURDOR task 5 ranges -100 to 100. It is positive if the system adds more information than errors. Two different systems participated to this task. The other participant proposed a *discriminative description* of the elements, based on SVM. We proposed a syntactic recognition system designed using Eyes Wide Open. We proposed a *description of the own specific properties* of the elements. Table 4 presents the result obtained during the Maurdor campaign by these two systems.

As it can be observed, the other participant system has better results than our system for the *type score*. Indeed, in the case of the Maurdor task 5, the determination of the functions of each text area is a classification task. The elements are already segmented, their transcription is known and we need to affect a label to each of these elements. It explains why a discriminative system obtains better results than ours. However in a



(a) Functions

(b) Logical connection

(c) Reading order

Fig. 12. Examples for each of the three types of information to annotate in MAURDOR

System	Type	Order	Group
Other participant	69	28	61
Our system	55	45	60

Table 4. Results obtained for the MAURDOR competition task 5 on test data set (1000 documents). Score in [-100;100], the higher is better.

more complete task, integrating the segmentation step, our method should be more efficient.

However, on the determination of the order and the membership to a group, we obtain similar to better results than the result obtained by the other participant. For these tasks, some knowledge on the structure of the elements is necessary to correctly describe the documents. The manual design of the grammatical description would have been a hard work due to the heterogeneity in the corpus. EWO allows us to design easily a system that obtained correct results for the competition.

9 Conclusion

In this paper, we presented Eyes Wide Open (EWO), a method for the inference of rule-based recognition system. This method introduces a learning process in the design of syntactical methods, in interaction with the user. Thanks to this method, we can benefit from the advantages of the syntactical methods (expressiveness, human understandable, possible introduction of user knowledge) without their main drawback which is the time needed to adapt the system to a new type of document.

The rules are built progressively with a logical and physical structures inference. This inference is made by the EAC clustering that is automatically done without any free parameter. Then, the interaction with the user brings semantic in the automatically inferred structures.

We used our method to produce grammatical descriptions on more than 2,000 documents of two different international competition data sets. EWO has been proved to be a generic method, usable on really different type of documents to produce almost automatically an adapted grammar, and then much faster. Moreover, the results obtained on the RIMES data set, with an error of 5.82%, are comparable to the best results obtained by syntactical methods manually designed [23] (5,53%) and to the best results obtained with a statistical method [24] (6,33%).

References

1. S. Brunessaux, P. Giroux, B. Grilheres, M. Manta, M. Bodin, K. Choukri, O. Galibert, and J. Kahn. The maurdor project: Improving automatic processing of digital documents. In *Document Analysis Systems (DAS)*, pages 349–354, April 2014.
2. C. Carton, A. Lemaitre, and B. Coüasnon. Learnpos: a new tool for interactive learning positioning. In *Document Recognition and Retrieval DRR XXI*, 2014.
3. A. Conway. Page grammars and page parsing. A syntactic approach to document layout recognition. In *ICDAR*, pages 761–764, 1993.
4. B. Coüasnon. Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. *IJDAR*, 8(2):111–122, 2006.
5. C. de la Higuera. A bibliographical study of grammatical inference. *Pattern Recogn.*, 38(9):1332–1348, 2005.
6. Vladimir Estivill-Castro. Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, June 2002.
7. C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.
8. A L N Fred and AK. Jain. Data clustering using evidence accumulation. In *ICPR*, volume 4, pages 276–280 vol.4, 2002.
9. E. Grosicki, M. Carree, J.-M. Brodin, and E. Geoffrois. Results of the rimes evaluation campaign for handwritten mail processing. In *ICDAR*, pages 941–945, 2009.
10. J.F. Hair, W.C. Black, B.J. Babin, and R.E. Anderson. *Multivariate Data Analysis, Seventh Edition.* Pearson Education, Inc, 2010. chapter 2.
11. Y. Ishitani. Logical structure analysis of document images based on emergent computation. In *ICDAR*, pages 189–192, 1999.
12. A K Jain, M N Murty, and P. J. Flynn. Data clustering: A review, 1999.
13. Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.*, 31(8):651–666, June 2010.
14. Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley-Interscience, 9th edition, 1990.

15. M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Trans. PAMI*, 15(7):737–747, Jul 1993.
16. L.I Kuncheva, S.T. Hadjitodorov, and L.P. Todorova. Experimental comparison of cluster ensemble methods. In *ICIF*, pages 1–7, 2006.
17. A. Lemaitre, J Camillerapp, and B. Coüasnon. A generic method for structure recognition of handwritten mail documents. In *Document Recognition and Retrieval DRR XV*, San Jose, États-Unis, 2008.
18. M. Lemaitre, E. Grosicki, E. Geoffrois, and F. Preteux. Preliminary experiments in layout analysis of handwritten letters based on textural and spatial information and a 2d markovian approach. In *ICDAR*, volume 2, pages 1023–1027, 2007.
19. C. Lin, Y. Niwa, and S. Narita. Logical structure analysis of book document images using contents information. In *ICDAR*, volume 2, pages 1048–1054 vol.2, Aug 1997.
20. Jianchang Mao and AK. Jain. A self-organizing network for hyperellipsoidal clustering (hec). *Neural Networks, IEEE Transactions on*, 7(1):16–29, Jan 1996.
21. S. Mao and T. Kanungo. Stochastic language models for automatic acquisition of lexicons from printed bilingual dictionaries. 2001.
22. S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms: A literature survey. 2003.
23. André O. Maroneze, Bertrand Coüasnon, and Aurélie Lemaitre. Introduction of statistical information in a syntactic analyzer for document image recognition. In *DRR*, pages 1–10, 2011.
24. F. Montreuil, S. Nicolas, E. Grosicki, and L. Heutte. A new hierarchical handwritten document layout extraction based on conditional random field modeling. In *ICFHR*, pages 31–36, Nov 2010.
25. N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy c-means model. *Trans. Fuz Sys.*, 3(3):370–379, August 1995.
26. Y. Rangoni, A. Belaid, and S. Vajda. Labelling logical structures of document images using a dynamic perceptive neural network. *IJDAR*, 15(1):45–55, 2012.
27. S. Shetty, H. Srinivasan, and S. Srihari. Segmentation and labeling of documents using conditional random fields. In *DRR*, 2007.
28. M. Shilman, P. Liang, and P. Viola. Learning nongenerative grammatical models for document analysis. In *ICVV*, volume 2, pages 962–969 Vol. 2, Oct 2005.
29. C. A. Sugar and G. M. James. Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association*, 98:750–763, 2003.
30. Y. Tateisi and N. Itoh. Using stochastic syntactic analysis for extracting a logical structure from a document image. In *ICPR*, volume 2, pages 391–394, 1994.
31. R. Tibshirani, W. Guenther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 2001.