Intelligent Document Processing

Floriana Esposito, Stefano Ferilli, Teresa M.A. Basile and Nicola Di Mauro
Department of Computer Science – University of Bari
{esposito, ferilli, basile, ndm}@di.uniba.it

Abstract

Digital repositories raise the need for an effective and efficient retrieval of the stored material. In this paper we propose the intensive application of intelligent techniques to the steps of document layout analysis, document image classification and understanding on digital documents. Specifically, the complex interrelation existing among layout components, that are fundamental to assign them the proper semantic role, suggest the exploitation of first-order representations in some learning steps. Results obtained in a prototypical system for scientific conference management prove that the proposed approach can be beneficial both for the layout recognition and for the selection of interesting components of the document, from which extracting the text for categorizing the document according to its topic.

1. Introduction

In the last years, the spread of computers and the Internet have caused a significant amount of documents to be available in electronic form. Collecting them in digital repositories raised problems that go beyond simple acquisition issues, and cause the need for organizing and classifying them in order to improve the effectiveness and efficiency of the retrieval procedure. The success of such a process is tightly related to the ability of understanding the semantics of the document components and content. Since the obvious solution of manually creating and maintaining an updated index is clearly infeasible, due to the huge amount of data under consideration, there is a strong interest in methods that can provide solutions for automatically acquiring such a knowledge. In this paper we propose the intensive application of intelligent techniques, and focus on the steps from document acquisition to the extraction of significant text, to be exploited for later categorization and information retrieval purposes. A preliminary Layout Analysis step is needed to identify the blocks that make up a document and to detect relations among them, resulting in the so-called *layout structure*. The next document processing step concerns the association of the proper logical role to each such component, resulting in so-called *logical structure*. This can enable a multiplicity of applications, including hierarchical browsing, structural hyperlinking, logical component-based retrieval and style translation. Document analysis is traditionally concerned with scanned images [8]. In this context Machine Learning techniques have been profitable applied as given evidence by WISDOM++ [4]. It is a knowledge-based system characterized by an extensive use of Machine Learning (ML) methods applied to infer automatically the rules for performing the various processing steps. Indeed, one of its distinguishing features is the use of a rule base to support some tasks performed in the various steps. The rule base is automatically built from a set of training documents using different inductive ML methods, which make the system highly adaptive. Here, differently from the past techniques that work on scanned image documents [8,6,7,4], we focus on electronic documents in PostScript (PS) or Portable Document Format (PDF), that represent the current *de facto* standard for document interchange and hence cover the greatest part of the available material. Our layout analysis process on documents in electronic format, sketched in Figure 1, starts with the application of a pre-processing module, called WINE, that rewrites basic PostScript operators to turn their drawing instructions into objects. It takes as input a PDF/PS document and produces (by an intermediate vector

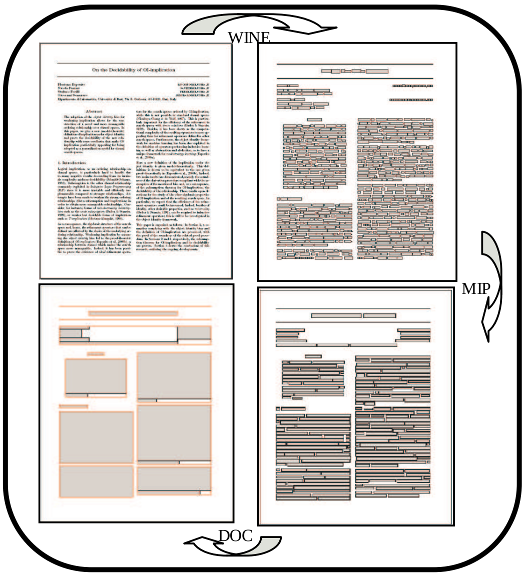


Figure 1 Document Analysis Layout

format) the initial document's *XML basic representation*, that describes it as a set of pages made up of *basic blocks*. Such a representation is then exploited by an algorithm, named DOC, that collects semantically related basic blocks into groups by identifying *frames* that surround them based on whitespace and background structure analysis. Specifically, DOC is a variant of Breuel's algorithm [1], that finds iteratively the maximal white rectangles in a page. The modification operated by DOC to such an algorithm consists in the identification of a stop criterion to end the process before finding insignificant white spaces such as inter-word or inter-line ones. Such a criterion was empirically established as the moment in which the area of the new white rectangle retrieved represents a percentage of the total white area in the document (computed by subtracting the sum of all the areas of the basic blocks from the whole area of the document) less than a given threshold based on the analysis of the significance of the contribution that the newly found white area gives to the overall result.

2. An exploitation scenario

Let us now present the functionality of the proposed system by means of a typical exploitation scenario. An Author connects to the Internet and opens the submission page, where (after registering, or after logging in if already registered) he can browse his hard disk and submit a paper by choosing the corresponding file in one of the accepted formats. The paper is received undergoes the following processing steps. The layout analysis algorithm is applied, in order to single out its layout components. Then, it is translated into a first-order logic description and classified by a proper module according to the theory learned so far for the acceptable submission layout standards (e.g., full paper, poster, demo). Depending on the identified class, a further step exploits the same description to locate and label the layout components of interest for that class (e.g., title, author, abstract and references in a full paper). The text that makes up each of such components is read, stored and used to automatically file the submission record (e.g., by filling its title, authors and abstract fields). If the system is unable to carry out any of these steps, such an event is notified to the Conference administrators, that can manually fix the problem and let the system complete its task. Such manual corrections are logged and used by the incremental learning component to refine the available classification/labeling theories in order to improve their performance on future submissions. Nevertheless, this is done off-line, and the updated theory replaces the old one only after the learning step is successfully completed: this allows further submissions to take place in the meantime, and makes the refinement step transparent to the Authors. Alternatively, the fixes can be logged and exploited all at once to refine the theory when its performance falls below a given threshold.

In this paper, we focus on the layout-related processing, up to text extraction. The next step, which is currently under investigation, should concern categorization of the paper content according to the text read. This would allow to match the paper topic against the reviewers' expertise, in order to find the best associations for the final assignment. Specifically, we plan to exploit the text contained in the title, abstract and bibliographic references, since we assume they concentrate the subject and research field the paper is concerned with, respectively. If a "Related Work"

Section is present, the corresponding citations can be identified and ignored, since they probably refer more to the historical background of the reported research than to its actual methods, techniques and findings. This also requires a pre-processing step that extracts from each reference the meaningful content (and ignores, for instance, page numbers, place and editors).

3. Exploitation of intelligent techniques

The first problem that arises when exploiting the DOC algorithm in real-world domains is due to the large number of basic blocks discovered by WINE that often correspond to fragments of words. A first aggregation based on their *overlapping* or *adjacency* usually yields composite blocks surrounding whole words. The number of blocks after this step is still large, thus a further aggregation (e.g., of words into lines) is needed (see Figure 1). Since grouping techniques based on the mean distance between blocks proved unable to correctly handle the case of multi-column documents, we turned to the application of ML approaches in order to automatically infer *rewriting rules* that could suggest how to set some parameters in order to group together rectangles (words) to obtain lines. Specifically, such a learning task was cast to a Multiple Instance Problem (MIP) and solved exploiting the algorithm proposed in [2]. Indeed, each elementary block is described by means of a feature-vector made up of parameters interpreted according to the representation in Figure 2.

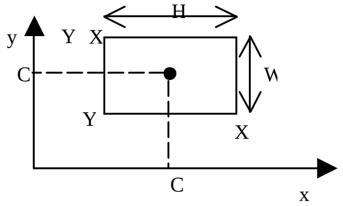


Figure 2 Block Features

Starting with this basic description of the blocks, the example description, from which rewriting rules have to be learned, is built considering the block itself and its **C**lose **N**eighbor blocks, where the information about the *x* and *y* positions for both the block considered and its neighbors, are the original ones and the parameters concerning the height and width of the blocks are replaced by two new parameters representing the distance between the block and each of its neighbors. Fixed the block O_n, we find all the instantiations for the close neighbors CNO_{nk} of the considered block O_n. This example (set of instances) will be labeled by an expert as positive for the target concept "*the two blocks can be merged*" if the blocks O_n and CNO_{nk} are adjacent and belong to the same line in the original document, as negative otherwise. In such a representation, a block has at least one close neighbor block and at most eight, i.e. positioned on top/bottom, in the left/right side and in the corners, hence, each example represents a bag of instances to which can be applied the *Iterated-Discrim* algorithm [2] in order to discover the relevant features and their values in this way obtaining rules made up of numerical constraints allowing to automatically set parameters to group together lines and frames. In this way, the *XML line-level description* of the document, that represents the actual input to DOC, is obtained. At the end of this step it could be possible that some blocks are not correctly recognized, i.e. white areas are considered black areas and *vice versa*. In such a case it is necessary a phase of layout correction that is automatically performed in DOC by embedded rules automatically learned for this task. Indeed, we firstly collect the manual corrections performed on some documents and represent them by means of a first-order description language representing both the situations before and after the manual correction and then we exploit first-order logic learning on this training set in order to identify correction rules. Once the layout structure has been identified, the semantic role must be associated to the significant components. Since the logical structure is obviously different according to the kind of document, two steps are in charge of identifying such a structure. The first aims at associating the document to a class that expresses its type (e.g., scientific/newspaper article, etc.). The second step aims at identifying the significant layout components for that class and at associating to each of them a tag that expresses its role (e.g., title, author, abstract, etc.). Again, we propose the application of first-order logic learning in these steps. In particular, the need for expressing relations among layout components requires the use of symbolic first-order techniques, while the continuous flow of new material, that is typical in digital document collections, calls for *incremental* abilities that can revise a faulty knowledge previously acquired. Thus, we exploited INTHELEX [5] to learn rules for the layout correction and embedded such learned rules, with a predictive accuracy of 98%, in DOC that automatically applies this information on new documents. Furthermore INTHELEX was embedded in the document processing system as the learning component for automatically labeling the documents along with their significant components, in order to help the automatic extraction of the interesting text to improve the organization of documents collection for a more efficient storage and retrieval process.

In order to exploit the learning system, a first-order logic representation of the document suitable for it must be provided. Dealing with multi-page documents, the document description have to contain page information (e.g., the page number; the total number of pages in a document, etc.). Furthermore, the set of rectangles in which each single page is divided are described by means of their type (text, graphic, line) and their horizontal and vertical position in the document. The algebraic relations are exploited to express the inclusion between pages-frames and between blocks and frames. The other relation described between rectangles is the spatial one. It is applied to all the blocks belonging to the same frame and to all the adjacent frames. Fixed a rectangle r one can ideally divide the plan containing it in 25 parts according to the plan partitions reported in [9]. Then, the relation between r and the other rectangles is described in terms of the occupied plans from them with respect to r. Such representation of the plans allows in turns to introduce in the descriptions the topological relations (*closeness, intersection* and *overlapping*) between rectangles [3], that are deduced by the spatial relationships by means of deduction and abstraction capabilities of the learning system.

4. Experiments

The experiments on document image classification and understanding were carried out on a dataset made up of 122 documents coming from online repositories and formatted according to the Springer-Verlag Lecture Notes (*LNCS)* style, the *Elsevier* journals style, the International Conference on Machine Learning (*ICML)* and the European Conference on Artificial Intelligence *(ECAI)* proceedings style. Each document description obtained by the pre-processing phase, was considered a positive example for the class it belongs to, and as a negative one for all the other classes to be learned. The system performance was evaluated according to a 10-fold cross validation methodology along with the execution time and predictive accuracy of the learned theories. The system was provided with a background knowledge expressing topological relations. Due to the different layout components that could be interesting for a specific class but not for others, the identification step of the layout components must be preceded by a classification process in order to infer rules able to recognize the correct class the document belongs to.

The experimental results averaged on the 10-fold of such task shows the lowest accuracy for *LNCS* (93.1%), w.r.t. the 98% of *Elsevier* and *ICML* and the 97% of *ECAI*, that could be expected due to the less strict check for conformance to the layout standard by the editor. The worst runtime for the *ECAI* class (3109.9 sec.), w.r.t 118.34 sec. of *Elsevier*, 51.86 sec. of *ICML* and 118.34 of *ECAI*, can be explained by the fact that this is actually a generalization of *ICML* (from which it differs because of the absence of two horizontal lines above and below the title), which makes hard the revision task. Anyway, the high predictive accuracy for this class should ensure that few revisions will be needed.

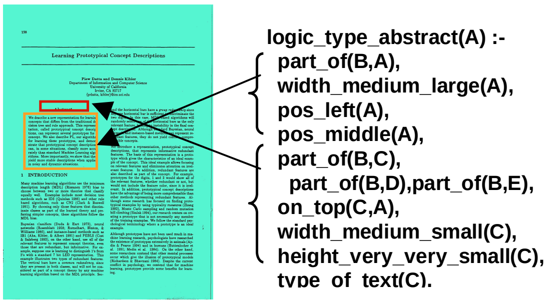
A second experiment aimed at learning rules able to identify the layout components was performed on the *title*, *authors*, *abstract* layout components of the documents belonging to the *LNCS* and *ICML*. These two classes were chosen since they represent two distinct kinds of layout (a single-column the former, a double-column the latter). Table 1 shows the averaged results on the 10 folds. As it is possible to note, also in this experiment the system showed good performance. Figure 3 reports a sample document for the ICML class of documents.

Table 1 System Performance for Understanding

LNCS	Time	Accur	*ICML*	Time	Accur
Title	33.47	95.93		51.67	97.87
Author	47.88	95.39		29.07	97.12
Abstract	133.7	93.06		111.13	97.51

Furthermore, it is showed one of the human readable rules discovered by the system to identify the layout components of such a class and the exact recognizing and mapping of the layout blocks referred to in the rules on the sample document. For instance, the rule describing the label *abstract* says that a “block A is an abstract of a document B belonging to *ICML* collection if its width is medium-large and it is placed on the left (w.r.t. the horizontal position) middle (w.r.t. the vertical position) part of the document”. This definition also refers to other three objects, C, D and E, one of which, C, has smaller size than the block A and is placed above A. The domain expert recognized block C as that containing the title (word) “Abstract” in a paper.

After these definitions are been learned, the system proceed to automatically extract the text in the electronic document from those blocks that are interesting for the user purpose, e.g. the text from the title and the abstract as reported in Figure 4.



```
logic_type_abstract(A) :-
  part_of(B,A),
  width_medium_large(A),
  pos_left(A),
  pos_middle(A),
  part_of(B,C),
    part_of(B,D),part_of(B,E),
  on_top(C,A),
  width_medium_small(C),
  height_very_very_small(C),
  tvpe of text(C).
```

Figure 3 Learned rules mapping on a document

5. Conclusions

Document management is critical for the distribution and preservation of knowledge. This paper proposes the intensive application of ML techniques to the steps of layout analysis, document image classification and understanding on documents in electronic format. The complex interrelation existing among layout components suggest the exploitation of first-order representations both in layout recognition and in automatically classifying the documents and their layout components, according to their semantics, from which extracting the text for categorizing the document subject. A prototypical system for scientific conference management implementing these ideas shows the benefit of the proposed approaches.

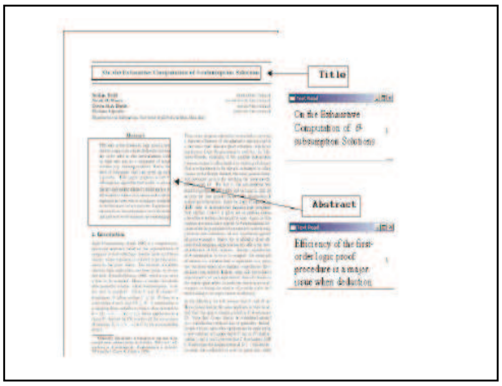

Title
On the Exhaustive Computation of θ-subsumption Solutions
Abstract
Efficiency of the first-order logic proof procedure is a major issue when deducting

Figure 4 Text Extraction from a document

References

[1] T.M. Breuel. Two geometric algorithms for layout analysis. In Workshop on Document Analysis Systems, 2002.

[2] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1-2):31-71, 1997.

[3] M. Egenhofer. Reasoning about binary topological relations. Proc. of SSD, Vol. 525 of LNCS, pp. 143-160. 1991.

[4] F. Esposito, D. Malerba, and F.A. Lisi. Machine learning for intelligent processing of printed documents. Journal of Intelligent Information Systems, 14(2/3):175-198, 2000.

[5] F. Esposito, S. Ferilli, N. Fanizzi, T.M.A. Basile, and N. Di Mauro. Incremental Multistrategy Learning for Document Processing. AAI Journal, 17(8/9):859-883, 2003.

[6] G. Ford, G.R. Thoma. Ground truth data for document image analysis Proc. of SDIUT'03, pp. 199-205, 2003.

[7] J Kim, D.X Le, G.R. Thoma. Automated labeling algorithms for biomedical document images Proc. of WMSCI’03, Vol. V, pp. 352-57, 2003.

[8]. G. Nagy. Twenty years of document image analysis. In IEEE TPAMI, 22(1):38-62, 2000.

[9] D. Papadias and Y. Theodoridis. Spatial relations, minimum bounding rectangles, and spatial data structures. IJGIS, 11(2):111-138, 1997.