Document Image Understanding for Digital Library Access

F. Esposito, S. Ferilli, N. Di Mauro, T.M.A. Basile

Dipartimento di Informatica
Università degli Studi di Bari

Abstract

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 obvious solution of manually creating and maintaining an updated index is clearly infeasible, due to the huge amount of data under consideration, thus 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 from document acquisition to the extraction of significant text, to be exploited for later categorization and information retrieval purposes. Experiments proving the viability of the proposed approach in the real-world sample application domain of automatic Scientific Conference Management are also reported.

Keywords

[I.2.1] Applications and Expert Systems, [I.2.6] Learning, [I.7] Document and Text Processing

1. Introduction

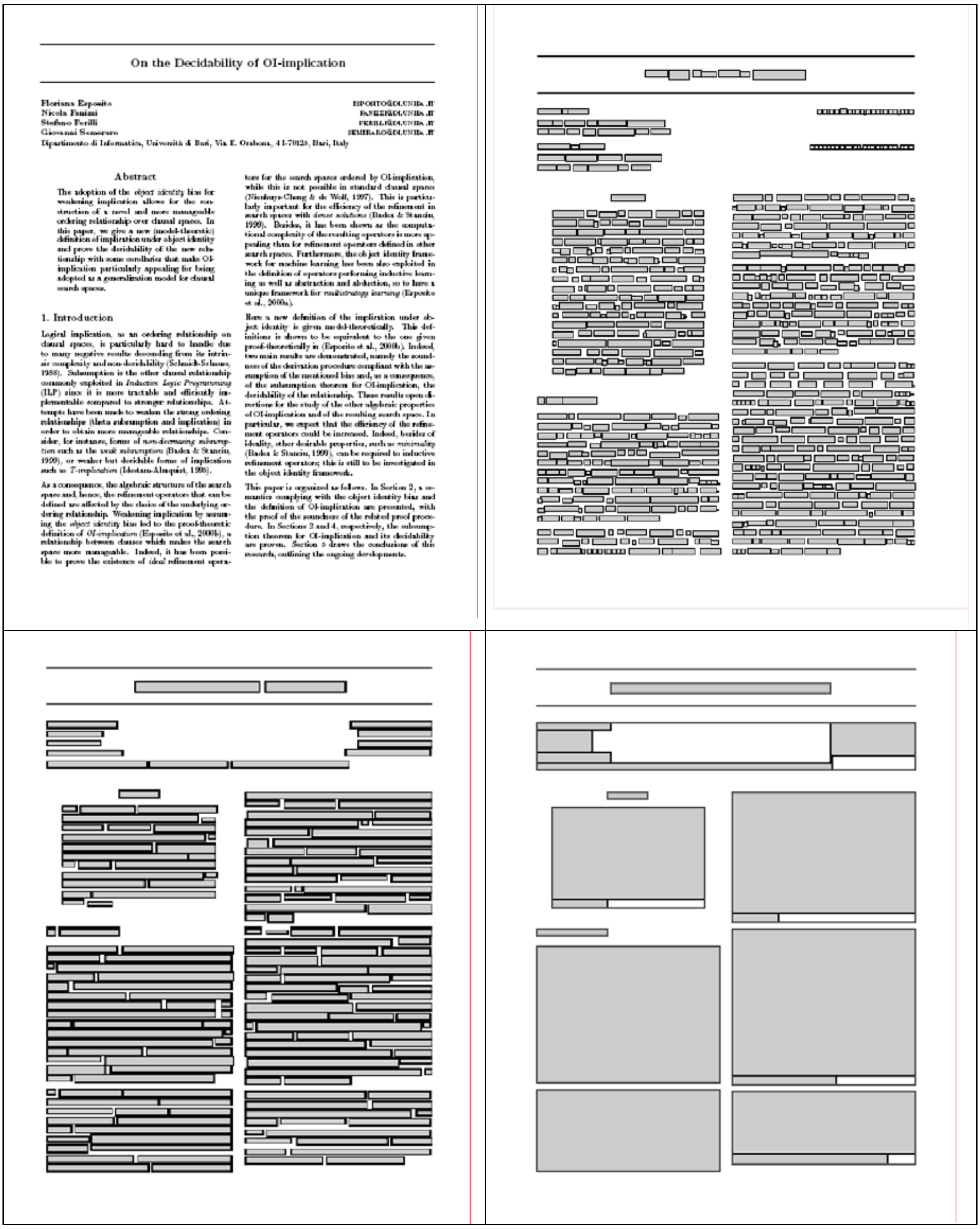
Knowledge distribution and preservation in time has always been a fundamental issue in the history of cultural development. Since knowledge can be conveyed by many different kinds of documents, document management assumed a key role in this perspective. A significant support to such a concern has been provided thanks to the power and flexibility offered by automatic techniques developed by computer science. Since several centuries up to a few decades ago, the only support available for dumping and spreading information was paper, which caused nearly the totality of our legacy material to be in the form of printed paper documents, often in few copies stored in archives and libraries. This clearly constitutes a serious obstacle to wide access and distribution of the information content they bear, which caused the need for document processing techniques aimed at ensuring preservation and access of the available material.

An obvious solution has been digitization, that fulfils the preservation requirements but is in itself not sufficient to solve the whole problem of content-based access. More precisely, the question is, given collections of digitized documents, how it is possible to discover among them useful knowledge to be used as meta-information to support their retrieval and management. This poses several sub-problems, involving issues that range from the analysis of the document layout, to the identification of the document type and of the role played by its components, up to the exploitation of techniques for document content indexing.

More recently, the situation has significantly changed, although paper has not lost its predominance as a means for conveying information. Specifically, in the last decades, the great diffusion of computers on one side, and of the Internet on the other, caused a migration or duplication of many documents from the paper support to the digital one. This greatly facilitates copy, distribution and accessibility of the documents themselves, but also poses new challenges to the computer science researchers, and raises a number of problems for their effective handling, which in turn require the development of different approaches and techniques that are able to deal with their peculiarities and characteristics. Indeed, the perspective has radically changed as well: documents are no more simply digitized images of an original paper counterpart, but are composed and produced directly in electronic form, of which the paper printout is just a facility that improves physical handling and readability. Moreover, a huge amount of documents in digital format are spread throughout the World Wide Web in the most diverse websites, and a whole research area focused on principles and techniques for setting up and managing document collections in the form of Digital Libraries has started and suddenly developed.

This work presents the current version of the prototype DOMINUS (DOcument Management INtelligent Universal System), a system for automated electronic documents processing characterized by the intensive exploitation of intelligent techniques in each step of the process from document acquisition to document indexing for categorization and information retrieval purposes. Since the system is general and flexible, it can be embedded as a document management engine into many different domain-specific applications. Here, we focus on the Conference Management domain, and show how DOMINUS can usefully support some of the more critical and knowledge-intensive tasks involved by the organization of a scientific conference, such as the assignment of the submitted papers to suitable reviewers.

The paper is organized as follows: after describing the techniques exploited by the various document processing steps performed by DOMINUS, a sample application to the paper-reviewer assignment in an international conference is presented. Performance of the system, and of its embedded Machine Learning techniques, in the various steps suggest it can be a powerful tool for automatic Digital Libraries management.



On the Decidability of OI-implication

Floriana Esposito ESPOSITO@DI.UNIBA.IT
Nicola Fanizzi FANIZZI@DI.UNIBA.IT
Stefano Ferilli FERILLI@DI.UNIBA.IT
Giovanni Semeraro SEMERARO@DI.UNIBA.IT
Dipartimento di Informatica, Università di Bari, Via E. Orabona, 4 I-70125, Bari, Italy

Abstract

The adoption of the *object identity* bias for weakening implication allows for the construction of a novel and more manageable ordering relationship over clausal spaces. In this paper, we give a new (model-theoretic) definition of implication under object identity and prove the decidability of the new relationship with some corollaries that make OI-implication particularly appealing for being adopted as a generalization model for clausal search spaces.

1. Introduction

Logical implication, as an ordering relationship on clausal spaces, is particularly hard to handle due to many negative results descending from its intrinsic complexity and non-decidability (Schmidt-Schauss, 1988). Subsumption is the other clausal relationship commonly exploited in *Inductive Logic Programming* (ILP) since it is more tractable and efficiently implementable compared to stronger relationships. Attempts have been made to weaken the strong ordering relationships (theta subsumption and implication) in order to obtain more manageable relationships. Consider, for instance, forms of *non-decreasing subsumption* such as the *weak subsumption* (Badea & Stanciu, 1999), or weaker but decidable forms of implication such as *T-implication* (Idestam-Almquist, 1993).

As a consequence, the algebraic structure of the search space and, hence, the refinement operators that can be defined are affected by the choice of the underlying ordering relationship. Weakening implication by assuming the *object identity* bias led to the proof-theoretic definition of *OI-implication* (Esposito et al., 2000b), a relationship between clauses which makes the search space more manageable. Indeed, it has been possible to prove the existence of *ideal* refinement operators for the search spaces ordered by OI-implication, while this is not possible in standard clausal spaces (Nienhuys-Cheng & de Wolf, 1997). This is particularly important for the efficiency of the refinement in search spaces with *dense solutions* (Badea & Stanciu, 1999). Besides, it has been shown as the computational complexity of the resulting operators is more appealing than for refinement operators defined in other search spaces. Furthermore, the object identity framework for machine learning has been also exploited in the definition of operators performing inductive learning as well as abstraction and abduction, so to have a unique framework for *multistrategy learning* (Esposito et al., 2000a).

Here a new definition of the implication under object identity is given model-theoretically. This definitions is shown to be equivalent to the one given proof-theoretically in (Esposito et al., 2000b). Indeed, two main results are demonstrated, namely the soundness of the derivation procedure compliant with the assumption of the mentioned bias and, as a consequence, of the subsumption theorem for OI-implication, the decidability of the relationship. These results open directions for the study of the other algebraic properties of OI-implication and of the resulting search space. In particular, we expect that the efficiency of the refinement operators could be increased. Indeed, besides of ideality, other desirable properties, such as *minimality* (Badea & Stanciu, 1999), can be required to inductive refinement operators; this is still to be investigated in the object identity framework.

This paper is organized as follows. In Section 2, a semantics complying with the object identity bias and the definition of OI-implication are presented, with the proof of the soundness of the related proof procedure. In Sections 3 and 4, respectively, the subsumption theorem for OI-implication and its decidability are proven. Section 5 draws the conclusions of this research, outlining the ongoing developments.

Figure 1: Line and final layout analysis representations of a document

2. Document Analysis

The availability of large, heterogeneous repositories of electronic documents is increasing rapidly, and the need for flexible, sophisticated document manipulation tools is growing correspondingly. These tools can usefully exploit the logical structure, a hierarchy of visually observable organizational components of a document, such as paragraphs, lists, sections, etc. Knowledge of this structure can enable a multiplicity of applications, including hierarchical browsing, structural hyperlinking, logical component-based retrieval and style translation. Identifying such a structure is in charge of a layout-based process called Document Analysis.

Document analysis is traditionally concerned with scanned images [9]. Conversely, this paper deals with electronic documents, available in large quantity on the Internet. Indeed, electronic documents have many advantages over paper ones, including compact and lossless storage, easy maintenance, efficient retrieval and fast transmission. Some of their major advantages are that they may have an explicit structure, that can take the form of a hierarchy of physical components (columns, paragraphs, textlines, words, tables, figures, etc.), or of a hierarchy of logical components (titles, authors, affiliations, abstracts, etc.), or both. Since this structural information can be very useful for indexing and retrieving the information contained in the document, physical layout and logical structure analysis of document images is a crucial stage in a document processing system. Hence, our goal is to automatically discover the logical structure of a digital document. Specifically, the presented approach is concerned with discovering a full logical hierarchy in digital documents in PostScript (PS) or Portable Document Format (PDF), based primarily on layout information. Indeed, PS and PDF are the current *de-facto* standards for electronic document representation and interchange.

In the layout of a document objects are spatially organized in *frames*, defined as collections of related objects completely surrounded by white space. Indeed, the document structure can be seen as a tree made up of a set of elementary blocks that can be merged in order to compose words; in turn, sets of words are grouped into lines and, finally, sets of lines can be merged to build frames, that together make up the whole document. The proposed approach performs the bottom-up construction of such a tree from the basic blocks up to the lines level and then proceeds top-down using another algorithm to discover the frames.

2.1. Basic PostScript Analysis

DOMINUS uses a pre-processing algorithm, named WINE (Wrapper for the Interpretation of Non-uniform Electronic document formats), that takes as input a PS or PDF document (such as the PDF version of this paper) and performs a syntactic transformation, producing a document in vector format, described in XML. PS is a simple interpretative programming language with powerful graphical capabilities, designed to describe the appearance of text, graphical shapes, and sampled images on printed or displayed pages. PDF is an evolution of PS rapidly gaining acceptance as a promising file format for electronic documents. Like PS, it is an open standard, enabling integrated solutions from a broad range of vendors, which is the reason why we focused on these two languages. WINE uses Ghostscript (available at `http://www.cs.wisc.edu/~ghost/`), an interpreter that is able to convert PS/PDF files to many raster formats, view them on displays, and print them on printers that don't have built-in PS capability. A similar algorithm is pstoedit (available at `http://www.pstoedit.net`). In particular, WINE rewrites PS operators to turn the instructions into objects. For example, the PS instruction to display a text becomes an object describing a text with attributes for the geometry (location on the page) and appearance (font, color, etc.). This way, a document is transformed into its corresponding basic representation, that describes the original digital document as a set of pages, each of which is composed of basic blocks.

2.2. Reduction of Basic Blocks

A problem that arises in real-world domains is due to the extremely 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

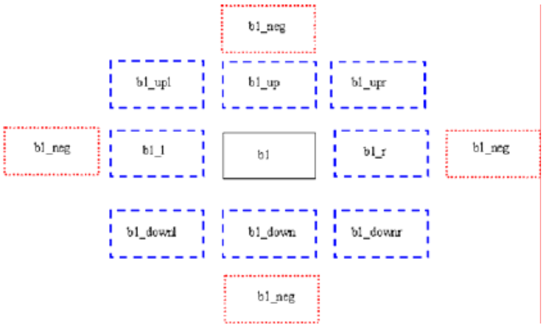


Figure 2: Close Neighbors for b1

a further aggregation 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, ML approaches were considered in order to automatically infer rewriting rules that could suggest proper settings for the parameters needed by algorithms that group together words to obtain lines. Specifically, such a learning task was cast to a Multiple Instance Problem (MIP) and, thus, solved by means of the Iterated-Discrim algorithm proposed in [5]. Each example (set of instances) consists of a block and its Close Neighbor blocks (as shown in Figure 2), plus the distance between the block and each of its neighbors, and is labelled by an expert as positive for the target concept "the two blocks can be merged" iff two blocks are adjacent (vertically or horizontally) and belong to the same line in the original document, or as negative otherwise. Then, the Iterated-Discrim algorithm discovers rules made up of numerical constraints explaining how to set some values in order to group together lines and frames. This yields the line-level description of the document, that represents the input to the next steps of layout analysis.

2.3. Geometric Layout Analysis

After transforming the original document into its line-level representation, DOMINUS applies DOC (Document Organization Composer), a variant of the algorithm reported in [2] for retrieving the whitespace and background structure of documents in terms of rectangular covers, to obtain the layout structure (see Figure 1 again) of the original document through a process that iteratively identifies and removes white rectangles by decreasing area. After identifying these background pieces inside the document, DOC computes their complement, thus obtaining the desired output. When computing the complement, two levels of description are generated: the former refers to single blocks filled with the same kind of content, the latter consists of rectangular frames that may be made up of many blocks of the former type. Thus, the overall description of the document includes both kinds of objects, plus information on which frames include which blocks and on the actual spatial relations between frames and between blocks in the same frame (e.g., above, touches, etc.). This allows to maintain both levels of abstraction independently.

2.4. Stop Criterion

Since DOC finds iteratively the maximal white rectangles up to a complete coverage of the document background, a stop criterion is needed to end this process at a useful grain-size (i.e., before normal inter-word or inter-line spaces are extracted). 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. The empirical study was performed applying the algorithm in full on a set of documents of three different categories, and it took into account the values of three variables in each step of the algorithm (whose evolution was tracked and plotted, as reported in Figure 3): Number of new white rectangles, normalized between 0 and 1 with respect to the maximum (black line), Percentage of the last white area retrieved with respect to the total white area of the page (bold line), Percentage of the white area retrieved with respect to the total white area of the page (dashed line). At such a point, highlighted in the figure with a black rhomboidal shape, all the useful white spaces in the document, have been identified.

3. Embedding Learning in Document Processing

Once the document layout structure has been identified, a number of tasks must be carried out on each document, and the huge amount of documents to be handled suggested the use of a concept learning system to

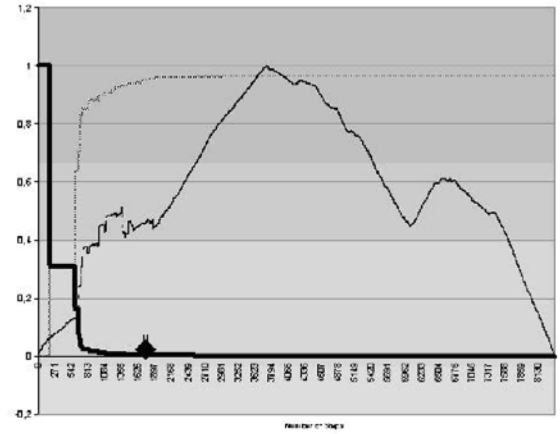


Figure 3: Stop Criterion Analysis

infer rules for such a task. Specifically, 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. For these reasons, DOMINUS exploits the learning system INTHELEX [7] to infer rules for automatically labelling the documents along with their significant components, aimed at supporting the automatic extraction of the interesting text and the organization of the document collection for a more efficient storage and retrieval process. INTHELEX is an incremental Inductive Logic Programming [8] system able to induce conceptual (first-order logic) descriptions from positive and negative examples. It incorporates inductive refinement operators to restore the correctness of the theory, deductive operators for recognizing known concepts that are implicit in the examples descriptions, abstraction operators for shifting the representation language and abduction operators to hypothesize unseen information.

3.1. Representation Language

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 must be enriched with page information. Each block/frame in the layout is described by means of its type (text, graphic, line) and horizontal/vertical position in the document. Also the inclusion relations between page and frame and between block and frame are expressed. The spatial relations, based on the partition of the plan with respect to a rectangle in 25 parts as reported in Figure 4 [12], are described both between blocks belonging to the same frame and between adjacent frames. Lastly, the topological relations between rectangles [6], including closeness, intersection and overlapping, are deduced from the spatial relationships by means of deduction and abstraction capabilities of the system, and included during the learning process.

As already pointed out, the multi-strategy capabilities of the learning system were exploited, which required proper information to be provided to it. For instance, the following fragment of background knowledge allowed it to exploit the above description language to infer topological relations (by means of its deductive capabilities):

```
over_alignment(B1,B2):-
      occupy_plan_9(B1,B2),
      not(occupy_plan_4(B1,B2)).
left_alignment(B1,B2) :-
      occupy_plan_17(B1,B2),
      not(occupy_plan_16(B1,B2)).
touch(B1,B2) :-
      occupy_plan_17(B1,B2),
      not(occupy_plan_13(B1,B2)).
```

On the other hand, the following is an extract of abstraction theory that allows the system to discretize numeric values of size and position (by exploiting its abstraction operators) for rectangles width discretization:

```
width_very_small(X):-
       rectangle_width(X,Y), Y >= 0,Y =< 0.023.
width_small(X):-
       rectangle_width(X,Y), Y > 0.023,Y =< 0.047.
width_medium_small(X):-
       rectangle_width(X,Y), Y > 0.047,Y =< 0.125.
width_medium(X):-
       rectangle_width(X,Y), Y > 0.125,Y =< 0.203.
```

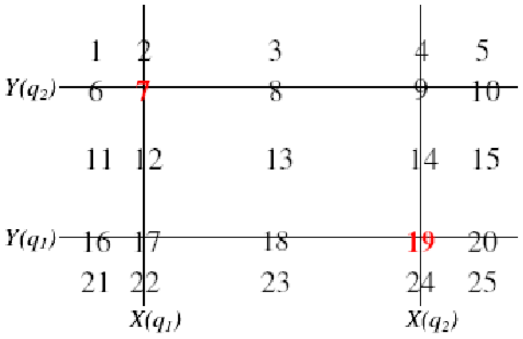


Figure 4: Representation Plans according to [12]

Classification	***SVLN***	***ECAI***	***ICML***	***Elsevier***	***Average***
Accuracy (%)	94	98	96	98	96,53
Runtime (sec)	204	100	69	54	107

	Accuracy (%)	***Runtime (sec)***	***Accuracy (%)***	***Runtime (sec)***
Understanding	**SVLN**		**ICML**	
Title	95,93	33	97,87	52
Author	95,39	48	97,12	29
Abstract	93,06	134	97,51	111
References	95,24	51	97,54	98

Table 1: INTHELEX Performance on the Classification and Understanding tasks

3.2. Learning Rules

Due to the fixed stop threshold, it could be the case that after the layout analysis step some white areas are not retrieved while some others that are retrieved are meaningless. In such a case a phase of layout correction would be desirable. 944 manual corrections of the former case and 953 of the latter, performed on 36 documents, were collected and described, representing the situation before and after the correction. Then, INTHELEX was exploited on this training set to identify correction rules, whose average accuracy according to a 10-fold cross-validation technique was 97.7%. Thus, they were embedded in DOC, in order to automatically perform the layout correction on new documents. Once the final layout structure has been identified, the semantic role must be associated to the significant components, in order to support the automatic extraction of the interesting text and the organization of the document collection for a more efficient storage and retrieval process. Since the logical structure is obviously different depending on the kind of document, two steps are in charge of identifying such a structure. The former aims at associating the document to a class that expresses its type (e.g., scientific/newspaper article, commercial letter, etc.). The latter 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. in a scientific paper).

The process aimed at learning rules to identify the class of a document and the role of its layout components was run on a dataset made up of 108 documents coming from online repositories and formatted according to the following styles: Springer-Verlag Lecture Notes in Computer Science (*LNCS*, 31 documents), *Elsevier* journals (32), Proceedings of the International Conference on Machine Learning (*ICML*, 20) and of the European Conference on Artificial Intelligence (*ECAI*, 25). Each document description obtained was considered a positive example for the class it belongs to, and as a negative example for all the other classes to be learned.

As already pointed out, a first learning process for the classification step was run, resulting in good system performance, according to a 10-fold cross validation methodology, in both runtime and predictive accuracy as reported in Table 1. The lowest accuracy refers to LNCS, which 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 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, respectively), which makes hard the revision task for the learning system. Anyway, the high predictive accuracy for this class should ensure that few revisions will be needed

A second phase aimed at learning rules able to identify the significant layout components was performed on the following layout components: *title*, *authors*, *abstract* and *references* of the LNCS and ICML documents. These two classes were chosen since they represent two distinct kinds of layout (a single-column the former, a double-column the latter). In Table 1 the averaged results on the 10 folds along with the execution time and predictive accuracy are reported. As it is possible to note, also in this experiment the system has shown good performance.

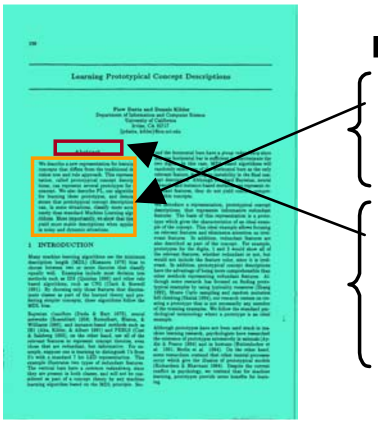
Figure 5 reports some of the rules discovered by the system to identify the layout components of the *ICML* set of documents. It is noteworthy that these rules have a high degree of human readability. For instance, the rule describing the label *abstract* says that a block *A* is an abstract of a document *B* belonging to the *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 block *A* and is placed above *A*. The domain expert recognized block *C* as that containing the title (word) "Abstract" in a paper. Indeed, as shown in Figure 5, it is possible to exactly recognize and map on a sample document the layout blocks referred to in the rules.

4. Document Indexing and Information Retrieval

A problem of most existing word-based retrieval systems consists of their ineffectiveness in finding interesting documents when the users do not use the same words by which the information they seek has been indexed. This is due to a number of tricky features that are typical of natural language. One of the most common concerns the fact that there are many ways to express a given concept (synonymy), and hence the terms in a user's query might not match those of a document even if it could be very interesting for him. Another one is that many words have multiple meanings (polysemy), so that terms in a user's query will literally match terms in documents that are not semantically interesting to the user. The Latent Semantic Indexing (LSI) technique [3] tries to overcome the weaknesses of term-matching based retrieval by treating the unreliability of observed term-document association data as a statistical problem. Indeed, LSI assumes that there exists some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to the retrieval phase and that can be estimated by means of statistical techniques.

LSI relies on a mathematical technique called Singular-Value Decomposition (SVD). Starting from a (large and usually sparse) matrix of term-document association data, the SVD allows to build and arrange a *semantic* space, where terms and documents that are closely associated are placed near each other, in such a way to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that do not actually appear in a document may still end up close to it, if this is consistent with the major association patterns in the data. Position in the space thus serves as a new kind of semantic indexing, and retrieval proceeds by using the terms in a query to identify a point in the space, and returning to the user documents in its neighbourhood. It is possible to specify a reduction parameter that intuitively represents the number of different concepts to be taken into account, among which distributing the available terms and documents.

The large amount of items that a document management system has to deal with, and the continuous flow of new documents that could be added to the initial database, require an *incremental* methodology to update the initial LSI matrix. Indeed, applying from scratch at each update the LSI method, taking into account both the old (already analysed) and the new documents, would become computationally inefficient. Two techniques have been developed in the literature to update (i.e., add new terms and/or documents to) an existing LSI generated database: Folding In [1] and SVD updating [11]. The former is a much simpler alternative that uses the existing SVD to represent new information but yields poor-quality updated matrices, since the information contained in the new documents/terms is not exploited by the updated semantic space. The latter represents a trade-off between the former and the recomputation from scratch.



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),
type_of_text(C).

```
logic_type_author(A) :-
      part_of(B,A),
      type_of_text(A),
      pos_upper(A), part_of(B,C),
      height_very_very_small(C),
      type_of_text(C),
      pos_upper(C), part_of(B,D),
      on_top(D,A), on_top(E,D),
      pos_center(E), pos_upper(E),
      width_very_large(D),
      height_smallest(D),
      pos_center(D), pos_upper(D).
```

```
logic_type_title(A) :-
      part_of(B,A),
      type_of_text(A),
      pos_center(A), pos_upper(A),
      part_of(B,C),
      height_very_very_small(C),
      type_of_text(C),
      on_top(C,D), on_top(A,E),
      part_of(B,E),
      width_very_large(E),
      height_smallest(E),
      pos_center(E), pos_upper(E).
```

Figure 5: Learned definitions for semantic components of ICML class

5. A Sample Application Scenario: Scientific Conference Management

Organizing scientific conferences is a complex and multi-faceted activity that often requires the use of a web-based management system to make some tasks a little easier to carry out, such as the job of reviewing papers. Some of the features typically provided by these packages are: submission of abstracts and papers by Authors; submission of reviews by the *Program Committee Members* (PCMs); download of papers by the *Program Committee* (PC); handling of reviewers preferences and bidding; web-based assignment of papers to PCMs for review; review progress tracking; web-based PC meeting; notification of acceptance/rejection; sending e-mails for notifications. One of the hardest and most time-consuming tasks in Scientific Conferences organization is the process of assigning reviewers to submitted papers. Due to the many constraints to be fulfilled, carrying out manually such a task is very tedious and difficult, and does not guarantee to result in the best solution.

This section shows how this complex real-world domain can take advantage of the proposed document management system as concerns both the indexing and retrieval of the documents and their associated topics. Specifically, it describes an expert component developed to be embedded in scientific Conference Management Systems (CMS), named GRAPE [4], that automatically assigns reviewers to papers submitted to a conference, additionally assessing the quality of the results in terms of profitability and efficiency, based on DOMINUS output.

A small pattern language has been defined in the literature, that captures successful practice in several conference review processes [10]. In this work we follow two patterns, indicating that papers should be matched, and assigned for evaluation, to reviewers who are competent in the specific paper topics (*ExpertsReviewPapers*), and to reviewers who declared to be willing to review those papers in the bidding phase (*ChampionsReviewPapers*). As to the former, in the current practice, before the submission phase starts, the PCC usually sets up a list of research topics of interest for the conference. Then, all reviewers are asked to specify which of them correspond to their main areas of expertise, while, during the submission process, authors are asked to explicitly state which conference topics apply to their papers. Such an information provides a first guideline for associating reviewers to papers. As to the latter pattern, after the submission phase ends, reviewers may perform the so-called bidding, i.e., after receiving a list of titles, authors and abstracts of all submitted papers, they may indicate which papers they would like or feel competent to review, and which ones they do not want to review (either because they do not feel competent, or because they have a conflict of interest). Further information to match papers and reviewers can be deduced from the papers. For example, related work by some reviewer explicitly mentioned in the paper might suggest that reviewer could be appropriate for that paper; conversely, if the reviewer is a co-author or a colleague of the paper authors, then a conflict of interest can be figured out.

One possible source of problems, in the above procedure, lies in the topics selected by the authors being sometimes misleading with respect to the real topic of the paper. For this reason, in order to make the assignment more objective, the explicit indication of the paper topics by their authors is replaced by an automatic inference of such an information by DOMINUS. Moreover, while the bidding preferences approach is usually preferred over the topics matching one, we give priority to the latter. Indeed, the topics selected by a reviewer should refer to his background expertise, whereas specific preferences about papers could be due to matter of taste or to other vague questions (e.g., the reviewer would like to review a paper just for curiosity; the abstract is imprecise or misleading, etc.). We believe that if a paper preferred by a reviewer does not match his topics of expertise, this should be considered as a warning.

5.1. GRAPE

GRAPE (Global Review Assignment Processing Engine) is an expert system, written in CLIPS, for solving the reviewers assignment problem, that takes advantage of both the papers content (topics) and the reviewers preferences (biddings). It could be used exploiting the papers topics only, or both the paper topics and the reviewers' biddings.

Let P = $\{p_1, ..., p_n\}$ denote the set of *n* papers submitted to the conference C, regarding *t* topics (*conference topics*, TC), and R=$\{r_1, ..., r_m\}$ the set of *m* reviewers. The goal is to assign the papers to reviewers, such that the following *basic constraints* are fulfilled:

1) each paper is assigned to exactly *k* reviewers (usually, *k* is set to 3 or 4);
2) each reviewer should have roughly the same number of papers to review (the mean number of reviews *per* reviewer is equal to *nk/m*);
3) papers should be reviewed by domain experts;
4) reviewers should revise articles based on their expertise and preferences.

As regards constraint 2, GRAPE can take as input additional constraints *MaxReviewsPerReviewer(r,h)*, indicating that the reviewer *r* can reviews at most *h* paper, that override the general principle and must be taken into account for calculating the mean number of reviews per reviewer.

No of Assigned Topics	No of Documents	No of Assigned Topics	No of Documents
1	31	**6**	27
2	44	**7**	13
3	51	**8**	11
4	43	**9**	5
5	32		

Table 2: Statistics Documents - Assigned Topics

Two measures were defined to guide the system during the search of the best solutions: the *reviewer's gratification* and the *article's coverage*. The former represents the gratification degree of a reviewer, calculated on the basis of the papers assigned to him. It is based on the *confidence degree* between the reviewer and the assigned articles (the confidence degree between a paper p_i concerning topics Tp_i and the reviewer r_j expert in topics Tr_j is defined as the number of topics in common) and on the number of assigned papers that were actually bid by the reviewer. The article's coverage represents the coverage degree of an article after the assignments. It is based on the *confidence degree* between the article and the reviewers it was assigned to (the same as before), and the *expertise degree* of the assigned reviewers (represented by the number of topics in which they are expert, and computed for a reviewer r_j as Tr_j / TC). GRAPE tries to maximize both measures during the assignment process, in order to fulfil the basic constraints 3 and 4. To reach this goal a fundamental requirement is that each reviewer must provide at least one topic of preference, otherwise the article coverage degree would be always null.

The assignment process is carried out into two phases. In the former, the system progressively assigns reviewers to papers with the lowest number of candidate reviewers. At the same time, the system *prefers* assigning papers to reviewers with few assignments. In this way, it avoids to have reviewers with zero or few assigned papers. Hence, this phase can be viewed as a search for review assignments by keeping low the average number of reviews *per* reviewer and maximizing the coverage degree of the papers. In the latter phase, the remaining assignments are chosen by considering first the confidence levels and then the expertise level of the reviewers. In particular, given a paper p_i which has not been assigned *k* reviewers yet, GRAPE tries to assign it to a reviewer r_j with a high confidence level between r_j and p_i. In case it is not possible, it assigns a reviewer with a high level of expertise.

The assignments resulting from the base process are presented to each reviewer, that receives the list A of the *h* assigned papers, followed by the list A' of the remaining ones, in order to actually issue his bidding. When all the reviewers have bid their papers, GRAPE searches for a new solution that takes into account these biddings as well, in addition to the information about expertise. In particular, it tries to change previous assignments in order to maximize both article's coverage and reviewer's gratification. By taking the article's coverage high, GRAPE tries to assign the same number of papers bid with the same class to each reviewer. Then, the solution is presented to the reviewers as the final one.

The main advantage of GRAPE relies in the fact that it is a rule-based system. Hence, it is very easy to add new rules in order to change/improve its behaviour, and it is possible to describe background knowledge, such as further constraints or conflicts, in a natural way. For example, one could insert a rule expressing the preference to assign a reviewer to the articles in which he is cited.

5.2. Evaluation

The system was evaluated on real-world dataset built by using data from different conferences [4]. Here we present an experiment carried out on a dataset composed by the 264 papers submitted to the 18th Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE 2005), whose Call for Papers identified 34 topics of interest. Firstly, the layout of each paper was automatically analysed in order to recognize the significant components. In particular, the abstract and title were considered the most representative of the document subject, and hence the corresponding text was read and the words contained therein were stemmed according to the technique proposed by Porter [13], resulting in a total of 2832 word stems. This allowed to apply the LSI in order to index the whole set of documents. The parameters for such a technique were set in such a way that all the conference topics were covered as different concepts.

The experiment consisted in performing 34 queries, each corresponding to one conference topic, on the database previously indexed. The results showed that 88 documents per query had to be considered, in order to include the whole set of documents. However, returning just 30 documents per query, 257 out of 264 documents (97,3%) are already assigned to at least one topic, which is an acceptable trade-off (the remaining 7 documents can be easily assigned by hand). Thus, 30 documents was considered a good parameter, and exploited to count the distribution of the topics between the documents, as reported in Table 3. Interestingly, more than half of the documents

(54,7%) concern between 2 and 4 topics, which could be expected both for the current interest of the researchers in mixing together different research areas and for the nature of the topics, that are not completely disjoint (some are specializations of others).

In order to have an insight on the quality of the results, in the following we present some interesting figures concerning the assignments suggested by GRAPE for the 264 papers of the IEA/AIE 2005 Conference, where the requirement was to assign each paper to 2 of the 60 Reviewers (i.e., k=2 reviews *per* paper were required). In solving the problem, the system was able to correctly assign 2 reviewers to each paper in 79.89 seconds. GRAPE was able to assign papers to reviewers by considering the topics only (it never assigned a paper by expertise). In particular, it assigned 10 papers to 38 reviewers, 9 to 4 reviewers, 8 to 6 reviewers, 7 to 1 reviewer, 6 to 9 reviewers, 5 to 1 reviewer, and 2 to 1 reviewer, by considering some *MaxReviewsPerReviewer* constraints for some reviewers that explicitly requested to revise few papers.

6. Conclusion

The huge amount of documents available in electronic form and the flourishing of digital repositories raises problems concerning document organization and classification, aimed at effectiveness and efficiency of their successive retrieval, that cannot be faced by manual techniques. This paper proposed the intensive application of intelligent techniques as a support to all phases of automated document processing, from document acquisition to document indexing. Experiments in the real-world sample application domain of automatic Scientific Conference Management, and in particular on the hard task of paper-reviewer assignment, prove the viability of the proposed approach.

Different future work directions are planned for the proposed system. First of all the document management system will be improved to handle different and more difficult kinds of documents. Second, the conference management system will be extended to cover other knowledge-intensive tasks currently in charge of the organizers, such as final presentations partition and scheduling according to the paper subject. Last, in a more general perspective, the proposed techniques will be applied to the problem of matching the documents in a digital library to the interests of the library users. The use of ontologies for improving matching effectiveness will be investigated as well.

References

[1] M.W. Berry, S.T. Dumais and G.W. O'Brien, “Using linear algebra for intelligent information retrieval,” Society for Industrial and Applied Mathematics Rev., vol. 37, no. 4, 1995, pp. 573-595.

[2] T.M. Breuel, “Two geometric algorithms for layout analysis,” Proc. Workshop on Document Analysis Systems, 2002.

[3] S. Deerwester et al., “Indexing by Latent Semantic Analysis,” Journal of the American Society of Information Science, vol. 41, no. 6, 1990, pp. 391-407.

[4] N. Di Mauro, T.M.A. Basile and S. Ferilli, “GRAPE: An Expert Review Assignment Component for Scientific Conference Management Systems,” Innovations in Applied Artificial Intelligence , F. Esposito and M. Ali, eds., LNCS 3533, Springer, 2005, pp. 789-798.

[5] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez, “Solving the multiple instance problem with axis-parallel rectangles,” Artificial Intelligence, vol. 89, no. 1/2, 1997, pp. 31-71.

[6] M. Egenhofer, “Reasoning about binary topological relations,” Proc. 2nd Symposium on Large Spatial Databases, LNCS 525, Springer, 1991, pp. 143-160.

[7] F. Esposito et al., “Incremental multistrategy learning for document processing,” Applied Artificial Intelligence: An International Journal, vol. 17, no. 8/9, 2003, pp. 859-883.

[8] S. Muggleton and L. De Raedt, “Inductive logic programming: Theory and methods,” Journal of Logic Programming, 19/20, 1994, pp. 629-679.

[9] G. Nagy, “Twenty years of document image analysis in PAMI,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 2000, pp. 38-62.

[10] O. Nierstrasz, “Identify the Champion,” N. Harrison, B. Foote and H. Rohnert, eds., Pattern Languages of Programm Design, 2000, pp. 539-556.

[11] G.W. O'Brien, “Information Management Tools for Updating an SVD-Encoded Indexing Scheme,” Master's Thesis, The University of Knoxville, Tennessee, 1994.

[12] D. Papadias and Y. Theodoridis, “Spatial relations, minimum bounding rectangles, and spatial data structures,” Int’l Journal of Geographical Information Science, vol. 11, no. 2, 1997, pp. 111-138.

[13] M.F. Porter, “An Algorithm For Suffix Stripping,” Program, vol. 14, no. 3, 1980, pp. 130-137.