Machine Learning methods for automatically processing historical documents: from paper acquisition to XML transformation

F.Esposito, D.Malerba, G.Semeraro, S.Ferilli, O. Altamura, T.M.A.Basile, M.Berardi, M.Ceci, N.Di Mauro
Dipartimento di Informatica – Università di Bari, Via E. Orabona, 4 - 70125 Bari
tel +39-080-5442140 fax +39-080-5443196
{esposito, malerba, semeraro, ferilli, altamura, basile, berardi, ceci, nicodimauro}@di.uniba.it

Abstract

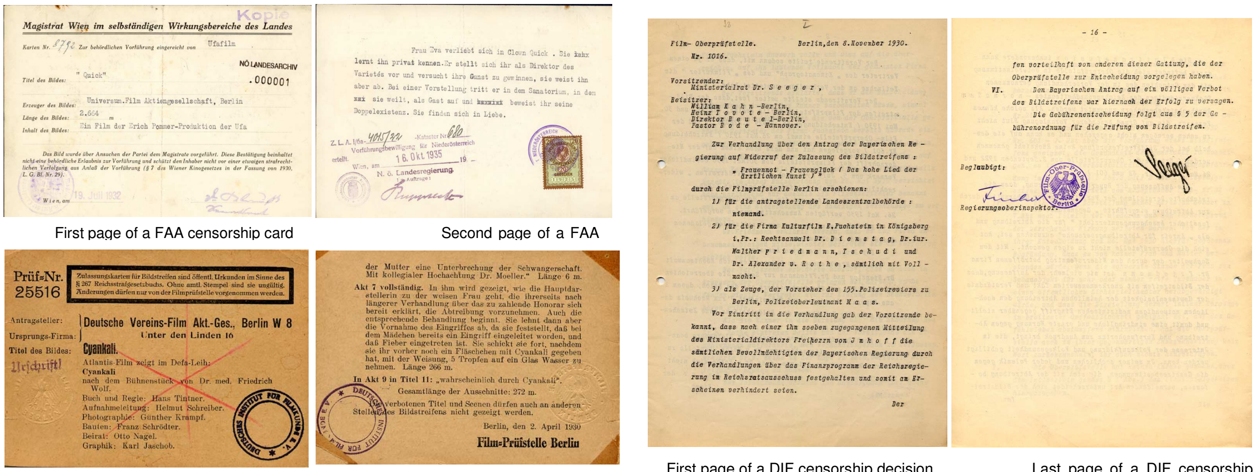
One of the aims of the EU project COLLATE is to design and implement a Web-based collaboratory for archives, scientists and end-users working with digitized cultural material. Since the originals of such a material are often unique and scattered in various archives, severe problems arise for their wide fruition. A solution would be to develop intelligent document processing tools that automatically transform printed documents into a web-accessible form such as XML. Here, we propose the use of a document processing system, WISDOM++, which uses heavily machine learning techniques in order to perform such a task, and report promising results obtained in preliminary experiments.

1. Introduction

The preservation of cultural heritage is mainly based on the possibility of saving, storing, accessing and interpretating cultural objects such as documents, texts, paintings and works of arts. Such objects are often unique, very valuable, fragile, irreplaceable and locally preserved in scientific collections at museums, in archives, or in urban and historic areas. Archives, museums and other cultural institutions also manage large collections of documents in the form of photos, expertise's papers, records, scientific studies and analyses. Both the objects themselves as well as the supplementary documentation are often accessible only through physical contact. Duplicates in the form of text (e.g., critical editions), or images (facsimiles, photographs) on paper are extremely expensive in terms of manpower, know-how and printing costs, and often cannot be justified for a small scientific audience. An answer to these problems might lie in the creation of digital libraries, enhanced by the concept of an annotation collaboratory, that are able to bundle documents, interpretation knowledge, work processes and an expert network in a flexible working environment.

The intrinsic nature of the document processing procedures poses several constraints that require tailored solutions. Intelligent Systems are becoming over the years valuable working instruments for researchers involved in humanistic sciences. The new challenge is to develop tools that can facilitate the fruition and investigation of the cultural heritage, to be used by non-experts or communities of researchers both for their personal work and for collaborative purposes. Technologically, the World Wide Web can serve as a standard communication platform for such communities as well as a gateway for document-centered digital library applications. Yet, while the Web may solve the problem of the diffusion and access of this material in its digital form, new automated tools are needed to allow a more intelligent processing and a personalized utilization of this knowledge.

In the project COLLATE (Collaboratory for annotation, indexing and retrieval of digitized historical archive material IST-1999-20882) one of the aims is to design and implement a Web-based collaboratory for people and insitutions working with digitized cultural material. The documents concern European films of the early 20th century that are not accessible in digitized formats. The need is to acquire and manage all the documents referring to a unique subject in order to reconstruct an entity in the knowledge base available on the Web. Documents are of different nature, often on partially damaged supports, with different standard and ancient typing characters. A straightforward application of OCR technology produces poor results because of the variability of the layout structure of printed documents. A more advanced solution would be to develop intelligent tools that automatically transform a large variety of printed documents into a web-accessible form. This requires a solution to several image processing problems, such as the separation of textual and graphical components in a document image (*document analysis*), the recognition of the kind of document (*document classification*), the identification of semantically relevant components of the page layout



Kopie

Magistrat Wien im selbständigen Wirkungsbereiche des Landes

Karten Nr. 8792 Zur behördlichen Vorführung eingereicht von Ufafilm

NÖ LANDESARCHIV
.000001

Titel des Bildes: " Quick"

Erzeuger des Bildes: Universum.Film Aktiengesellschaft, Berlin

Länge des Bildes: 2.664 m

Inhalt des Bildes: Ein Film der Erich Pommer-Produktion der Ufa

Das Bild wurde über Ansuchen der Partei dem Magistrate vorgeführt. Diese Bestätigung beinhaltet nicht eine behördliche Erlaubnis zur Vorführung und schützt den Inhaber nicht vor einer etwaigen strafrechtlichen Verfolgung aus Anlaß der Vorführung (§ 7 des Wiener Kinogesetzes in der Fassung von 1930, L. G. Bl. Nr. 29).

Wien, am 19. Juli 1932

First page of a FAA censorship card

Frau Eva verliebt sich in Clown Quick . Sie ~~lehx~~ lernt ihn privat kennen. Er stellt sich ihr als Direktor des Varietés vor und versucht ihre Gunst zu gewinnen, sie weist ihn aber ab. Bei einer Vorstellung tritt er in dem Sanatorium, in dem ~~xxx~~ sie weilt, als Gast auf und ~~xxxxxxx~~ beweist ihr seine Doppelexistenz. Sie finden sich in Liebe.

Z. L. A. I/6a-4015/22 -Kataster Nr. 680
Vorführungsbewilligung für Niederösterreich
erteilt.
Wien, am 16. Okt. 1935 19 –
N. ö. Landesregierung.
Im Auftrage:

NIEDERÖSTERREICH
ÖSTERREICHISCHE STEMPELMARKE 20 GROSCHEN 1925

Second page of a FAA

Prüf=Nr.
25516

Zulassungskarten für Bildstreifen sind öffentl. Urkunden im Sinne des § 267 Reichsstrafgesetzbuchs. Ohne amtl. Stempel sind sie ungültig. Änderungen dürfen nur von der Filmprüfstelle vorgenommen werden.

Antragsteller: Deutsche Vereins-Film Akt.-Ges., Berlin W 8
Ursprungs-Firma: Unter den Linden 16

Titel des Bildes: Cyankali.

Urschrift!

Atlantis-Film zeigt im Defa-Leih:
Cyankali
nach dem Bühnenstück von Dr. med. Friedrich Wolf.
Buch und Regie: Hans Tintner.
Aufnahmeleitung: Helmut Schreiber.
Photographie: Günther Krampf.
Bauten: Franz Schrödter.
Beirat: Otto Nagel.
Graphik: Karl Jaschob.

DEUTSCHES INSTITUT FÜR FILMKUNDE E.V.

First page of a DIF censorship card

der Mutter eine Unterbrechung der Schwangerschaft. Mit kollegialer Hochachtung Dr. Moeller." Länge 6 m.

Akt 7 vollständig. In ihm wird gezeigt, wie die Hauptdarstellerin zu der weisen Frau geht, die ihrerseits nach längerer Verhandlung über das zu zahlende Honorar sich bereit erklärt, die Abtreibung vorzunehmen. Auch die entsprechende Behandlung beginnt. Sie lehnt dann aber die Vornahme des Eingriffes ab, da sie feststellt, daß bei dem Mädchen bereits ein Eingriff eingeleitet worden, und daß Fieber eingetreten ist. Sie schickt sie fort, nachdem sie ihr vorher noch ein Fläschchen mit Cyankali gegeben hat, mit der Weisung, 5 Tropfen auf ein Glas Wasser zu nehmen. Länge 266 m.

In Akt 9 in Titel 11: „wahrscheinlich durch Cyankali".

Gesamtlänge der Ausschnitte: 272 m.

Die verbotenen Titel und Scenen dürfen auch an anderen Stellen des Bildstreifens nicht gezeigt werden.

Berlin, den 2. April 1930

Film=Prüfstelle Berlin

DEUTSCHES INSTITUT FÜR FILMKUNDE E.V.

Last page of a DIF censorship

38 I

Film- Oberprüfstelle. Berlin, den 8. November 1930.
Nr. 1016.

Vorsitzender:
Ministerialrat Dr. Seeger,

Beisitzer:
William Kahn -Berlin,
Heinz Tovote - Berlin,
Direktor Beutel-Berlin,
Pastor Bode - Hannover.

Zur Verhandlung über den Antrag der Bayerischen Regierung auf Widerruf der Zulassung des Bildstreifens :

„ Frauennot - Frauenglück (Das hohe Lied der ärztlichen Kunst) "

durch die Filmprüfstelle Berlin erschienen:

1) für die antragstellende Landeszentralbehörde : niemand.

2) für die Firma Kulturfilm E. Puchstein in Königsberg i. Pr.: Rechtsanwalt Dr. Dienstag, Dr. iur. Walther Friedmann, Tschudi und Dr. Alexander v. Rothe, sämtlich mit Vollmacht.

3) als Zeuge, der Vorsteher des 155. Polizeireviers zu Berlin, Polizeioberleutnant Maas.

Vor Eintritt in die Verhandlung gab der Vorsitzende bekannt, dass nach einer ihm soeben zugegangenen Mitteilung des Ministerialdirektors Freiherrn von Imhoff die sämtlichen Bevollmächtigten der Bayerischen Regierung durch die Verhandlungen über das Finanzprogramm der Reichsregierung im Reichsratsausschuss festgehalten und somit am Erscheinen verhindert seien.

Der

First page of a DIF censorship decision

- 16 -

fen vorteilhaft von anderen dieser Gattung, die der Oberprüfstelle zur Entscheidung vorgelegen haben.

VI. Dem Bayerischen Antrag auf ein völliges Verbot des Bildstreifens war hiernach der Erfolg zu versagen.

Die Gebührenentscheidung folgt aus § 5 der Gebührenordnung für die Prüfung von Bildstreifen.

Beglaubigt:

Film-Ober-Prüfstelle Berlin

Regierungsoberinspektor.

Last page of a DIF censorship

Figure 1. Examples of documents to be processed

(*document understanding*), the extraction of sequences of characters from portions of the document image (OCR), and the transformation of the page into HTML/XML format. A large amount of knowledge is required to this purpose [1], [4]: While in the literature a great effort is made to hand-code such a knowledge [15], we propose the massive application of inductive learning techniques throughout all the steps of document processing.

2. Wisdom ++

WISDOM (Windows Interface System for Document Management) is the system used to process the documents [6]. 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 machine learning methods, which make the system highly *adaptive*.

Document images processed in the COLLATE project (see Figure 1) are provided by three national film archives: Deutsches Filminstitut (DIF), Filmarchiv Austria (FAA) and Národní Filmový Archiv (NFA). Generally, documents are multi-page, each page being an RGB 24bit color image in TIFF format whose size can reach up to 50 MB. Since WISDOM++ can manage 300dpi black-and-white images of at most A4-format documents, a preliminary conversion is necessary. In this application only some pages of each document are interesting for document classification and understanding. Specifically, only the first, second and last page have been processed.

Document preprocessing consists in the evaluation of the skew angle, the rotation of the document, as well as the computation of a spread factor. The evaluation of the skew angle is essential, since the method used for the subsequent step of document segmentation is generally ineffective when applied to skewed documents. Once the skew angle has been estimated the document image can be corrected by means of an inverse rotation operator. The *spread factor* of the document image is used to define some parameters of the segmentation algorithm. At the end of the preprocessing phase, the resolution of the document image is reduced to 75 dpi (about 70 KB for an A4-sized page), which is a reasonable trade-off between accuracy and speed of the segmentation process and also filters out noisy black specks on a white background.

If the primary goal of the document analysis process is interpretation of text data, graphic data appearing in the digitized document must be first separated from the text so that subsequent processing stages may operate exclusively on the textual information. This is obtained in two steps: image segmentation and block classification. WISDOM++ segments the reduced document image into rectangular blocks by means of an efficient variant of the Run Length Smoothing Algorithm (RLSA) [18]. In order to facilitate subsequent processing steps, each block must be classified according to the type of content: text, horizontal line, vertical line, picture (i.e., halftone images) and graphics (e.g., line drawings). Such a classification is performed by means of a decision tree automatically built from a set of training examples (blocks) of the five classes whose performance has been reported in [2]. The result of the segmentation process is a list of classified blocks, corresponding to printed areas in the page image. Each block is described by its top left-hand and bottom right-hand corners' coordinates, and its type.

The number of blocks is generally less than a hundred; however, this representation is still too detailed for learning document classification and understanding rules. The perceptual organization process that aims at detecting structures among blocks is called the *layout analysis*. The result is a hierarchy of abstract representations of the

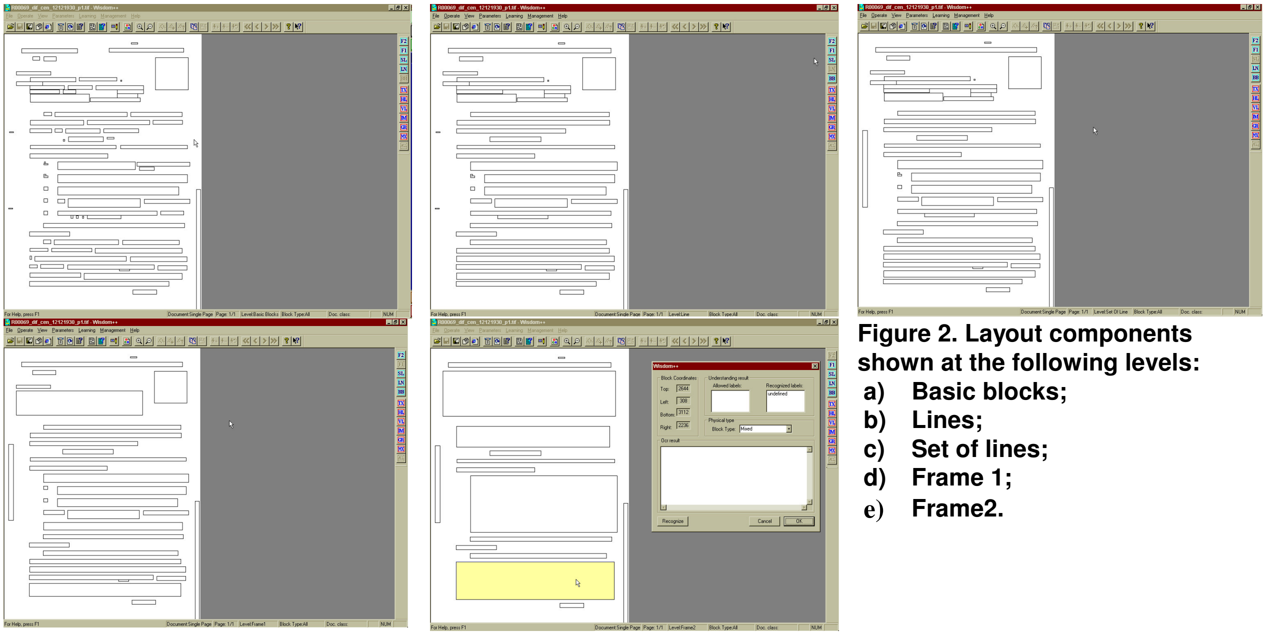


Figure 2. Layout components shown at the following levels:
a) Basic blocks;
b) Lines;
c) Set of lines;
d) Frame 1;
e) Frame2.

document image, the *layout structure*. The leaves of the layout tree are the blocks, while the root represents the whole document. A page may group together several layout components, called *frames*, which are rectangular areas of interest in the document page image. The layout analysis is done in two steps:

1. A *global analysis* in order to determine possible areas containing paragraphs, sections, figures and tables. This step is based on an iterative process, in which the vertical and horizontal histograms of text blocks are alternatively analyzed in order to detect columns and sections/paragraphs, respectively.
2. A *local analysis* to group together blocks which possibly fall within the same area. Perceptual criteria considered in this step are: *proximity* (e.g. adjacent components belonging to the same column/area are equally spaced), *continuity* (e.g. overlapping components) and *similarity* (e.g. components of the same type, with an almost equal height).

Pairs of layout components that satisfy some of these criteria may be grouped together. Each layout component is associated with one of the following types: text, horizontal line, vertical line, picture, graphic and mixed. When the constituent blocks of a component are homogeneous, the same type is inherited by the layout component; otherwise, the associated type is set to *mixed.*

The layout structure extracted by WISDOM++ is a hierarchy with six levels: basic blocks, lines, set of lines, frame1, frame2, and pages (see Figure 2). If the user is not satisfied with the result of the layout analysis he can act directly on the results of the segmentation process by deleting some blocks or he can modify the result of the global analysis by performing three different operations:

a) Horizontal splitting: a column/section is cut horizontally.
b) Vertical splitting: a column/section is cut vertically.
c) Grouping: two sections/columns are merged.

After each operation, WISDOM++ recomputes the result of the local analysis process, so that the user can immediately perceive the final effect of the requested corrections and can decide whether to confirm or reject it. Once the user has completed the correction process, WISDOM++ has the description of when and how the user has modified the result of the global analysis, and generates corresponding training observations in order to automatically learn rules for the automated correction of the layout analysis by means of one of the learning systems WISDOM++ incorporates [13].

The splitting operations can be described by means of a binary function, *split(X,S)*, where *X* represents the column/section to be split, *S* is an ordinal number representing the step of the correction process, and the range of the split function is the set {*horizontal*, *vertical*, *no_split*}. The grouping operation can be described by a ternary predicate *group(A,B,S)*, where *A* and *B* are the two grouped sections (columns) and *S* is the ordinal number representing the step of the correction process.

3. Document classification

The logical components of the document, such as title, authors, can be identified after having detected the layout structure. They can be arranged in a hierarchical structure, which is called *logical structure*, resulting by a division of the content of a document into increasingly smaller parts. The leaves of the logical structure are the basic logical components, such as authors of a paper. The heading of an article, encompassing the title and the author, is therefore a composite logical component. The root of the logical structure is the document class. The discovery of the logical structure of a document can be cast as the problem of associating some layout components with a correspondent logical component. In WISDOM++ it consists in the association of a page with a document class (*document classification*) and of second frames with basic logical components (*document understanding*).

Classification is performed by matching the layout structure of the first page against *models of classes of documents* that are able to capture the invariant properties of the images/layout structures of documents belonging to the same class. They are expressed in a first-order logic language, so that the classification problem can be reformulated as a matching test between two logic formulae: one that describes a model and another that represents the image/layout properties of the first page. In such a first-order logic language unary function symbols (*attributes*) describe properties of a layout component, while binary predicate and function symbols (*relations*) express spatial relationships between layout components. Table 1 reports a list of the attributes/relations used.

Table 1 . Attributes/Relations used to describe both the models and the documents.

Attribute/relation name (Extracted from)	**Definition**
image_length(doc) (Image)	Integer domain (1..5000)
image_width(doc)(Image)	Integer domain (1..4000)
width(block) (Page layout)	Integer domain (1..640)
height(block) (Page layout)	Integer domain (1..890)
x_pos_centre(block) (Page layout)	Integer domain (1..640)
y_pos_centre(block) (Page layout)	Integer domain (1..875)
type_of(block) (Page layout)	Nominal domain: text, hor_line, image, ver_line, graphic, mixed
part_of(page,block) (Page layout)	Boolean domain: true if page contains block
on_top(b1,b2) (Page layout)	Boolean domain: true if block b1 is above b2
to_right(b1,b2) (Page layout)	Boolean domain: true if block b2 is to the right of b1
alignment(block1,block2) (Page layout)	Nominal domain: only_left_col, only_right_col, only_middle_col, both_columns, both_rows only_upper_row, only_lower_row, only_middle_row

The rules can be learned from a set of training examples of documents classified by the user. Two different strategies can be applied to learning tasks: batch and incremental. The system embedded in WISDOM++ for batch learning is ATRE [11]. The learning problem can be formulated as:

Given: a set of concepts $C_1, C_2, \dots, C_r$ to be learned, a set of observations O described in a language L_O, a background knowledge BK described in a language L_{BK}, a language of hypotheses L_H, a generalization model over the space of hypotheses and a user's preference criterion

Find: a (recursive) logical theory T for the concepts $C_1, C_2, \dots, C_r$, such that T is complete and consistent with respect to O and satisfies the user's preference criterion.

As to the representation language, the basic component is the *literal* in the two forms: $\boldsymbol{f(t_1, \dots ,t_n)=Value}$ where f is an n-ary function symbol, i.e. a relation or attribute, t_i's are constant terms, and *Value* is one of the possible values of f s domain o$\boldsymbol{f(t_1, \dots, t_n) \in Range}$ (***set literal***)**,** where f is function symbol called *descriptor*, t_i' s are terms, and *Range* is a closed interval of possible values taken by f.

The *language of observations* L_O allows an efficient and comprehensible *object-centered representation* of observations that are represented by ground multiple-head clauses made up of a conjunction of simple literals in the head. Indeed, in order to reduce the computational complexity of the problem, the description of the document is restricted to the properties of the frame2 layout components alone. A partial description of the page layout of a document follows:

**image_lenght(1)=3468, image_width(1)=2418,
part_of(1,2)=true, part_of(1,3)=true, …, part_of(1,25)=true,
width(2)=15, width(3)=20, …, width(25)=429,
height(2)=239, height(3)=4, …, height(25)=24,
type_of(2)=text, type_of(3)=text, …, type_of(25)=text,
x_pos_centre(2)=20, …, x_pos_centre(25)=334,
y_pos_centre(2)=420, …, y_pos_centre(25)=558,
on_top(3,9)=true, on_top(9,8)=true, …, on_top(19,20)=true,
to_right(2,7)=true, to_right(3,9)=true, …, to_right(2,4)=true,
alignment(3,13)=only_right_col,
alignment(7,8)=only_upper_row, ...**

where the constant *1* denotes the whole page, and *2–25* the layout components at the frame2 level.

Examples can be considered as *positive* or *negative*.

The *language of hypotheses*, L_H is that of *linked*, *range-restricted* definite clauses [5] with simple set literals in the body and one simple literal in the head.

The *language of background knowledge* L_{BK} has the same constraints as L_H: this knowledge allows to reduce the search space of the hypotheses with different biases.

ATRE's theories can be transformed into Datalog programs [3] with built-in predicates: a simple literal $f(t_1,\dots,t_n)=Value$ can be transformed into an $(n+1)$-ary predicate $f(t_1,\dots,t_n,Value)$, while a set literal $f(t_1, \dots, t_n) \in Range$, where *Range* is an interval $[a\ ..\ b]$, can be transformed into $f(t_1, \ldots, t_n, Z)$, $Z \geq a$, $Z \leq b$. In such a way, it is possible to extend notions and properties of first-order logic to ATRE definite clauses.

Regardless of the chosen representation language, a key role of the induction process is the search through a space of hypotheses. A *generalization model* provides a basis for organizing this search space, since it establishes when a hypothesis covers a positive/negative example and when an inductive hypothesis is more general/specific than another. The generalization model adopted in ATRE is a variant of Plotkin' s *relative generalization* [16], named *generalized implication* [11].

The learning algorithm in ATRE belongs to the family of *separate-and-conquer* algorithms [14]: It is based on the strategy of learning one clause at a time (*conquer step*), removing the covered examples (*separate step*) and iterating the process on the remaining examples.

Some preliminary experimental results on the task of learning rules for classification of COLLATE documents are reported. The dataset consisted of 89 documents: 18 of class *FAA censorship card (faa_cen)*, 10 of *A4-size DIF censorship decision (dif_cen_decision)*, 5 of *DIF censorship card (dif_cen_card)*; other 56 of class Reject (*Newspaper Articles*). Learned rules and runtime are in the following Table 2.

Table 2 Rules for document classification

class(X1)=dif_cen_decision ← part_of(X1,X2)=true, type_of(X2)=text image_lenght(X1)∈[3389 .. 3507], y_pos_centre(X2)∈[670 .. 841]	400.883 s
class(X1)=dif_cen_card ← image_lenght(X1)∈[1211 .. 1227]	3.243 s
class(X1)=faa_cen ← image_lenght(X1)∈[1689 .. 1730], image_width(X1)∈[2417 .. 2483].	74.448 s

The first rule states that a document belongs to *DIF censorship decision* if the image length of its first page is between 3389 and 3507 pixels (as expected from A4 sized documents) and the layout structure of its first page contains a frame2 component of type text with a centroid positioned vertically between row 670 and row 841.

The second rule emphasizes the importance of the descriptor image_length, which is enough to discriminate a DIF censorship card from a DIF censorship decision.

The third clause confirms the importance of handling numerical descriptors in ATRE, since FAA censorship documents are characterized by two conditions: length of the document image between 1689 and 1730, and image width between 2417 and 2483.

4. Document understanding

In document understanding, layout components are associated with logical components. This association can theoretically affect layout components at any level in the layout hierarchy. However, in WISDOM++ only frame2 components are associated with some component of the logical hierarchy. Moreover, only layout information is used in document understanding. This approach differs from that proposed by other authors [9] which additionally make use of textual information, font information and universal attributes given by the OCR. This diversity is due to a different conviction on when an OCR should be applied. We believe that only some layout components of interest for the application should be subject to OCR, hence document understanding should precede text reading and cannot be based on textual features. Two assumptions are made: documents belonging to the same class have a set of relevant and invariant layout characteristics; logical components can be identified by using layout information only.

Document understanding of all pages is performed by matching the layout structure of the each page against *models of logical components*. An example of models for the logical components *running_head* and *paragraph* in the case of papers published in magazines might be:

logic_type(X)= running_head ←
position(X)= top_left, type(X)=text, page_number(X)=even
logic_type(X)= running_head ←
position(X)=top_righ, type(X)= text, page_number(X)=odd
logic_type(Y)= paragraph ←
on_top(X,Y)=true, logic_type(X)=running_head,
type(Y)=text

These rules mean that a textual layout component at the top left (right) hand corner of an even (odd) page is a running head, while a textual layout component below a running-head is a paragraph of the paper.

This example shows that the document understanding problem cannot be effectively reformulated as a simple matching test between logic formulae. The association of the logical description of pages with logical components requires a full-fledged theorem prover.

Attributes and relations used to describe the layout of each page to be “understood”, partially overlaps with those presented in Table 1. The two image attributes *image_length* and *image_width* are no longer necessary, while it is important to introduce a new attribute *page* which specifies the position of the page to be *understood* within the document (first, second, etc.). An example of the page layout for document understanding purposes is:

page(1)=first,
part_of(1,2)=true, part_of(1,3)=true, .., part_of(1,25)=true,
width(2)=15, width(3)=20, .., width(25)=429,
height(2)=239, height(3)=4, .., height(25)=24,
type_of(2)=text, type_of(3)=text, .., type_of(25)=text,

x_pos_centre(2)=20, .., x_pos_centre(25)=334,
y_pos_centre(2)=420, .., y_pos_c entre(25)=558,
on_top(3,9)=true, on_top(9,8)=true, .., on_top(19,20)=true,
to_right(2,3)=true, to_right(2,5)=true,..,to_right(3,5)=true,
alignment(9,12)=only_right_col, ..,
alignment(7,8)=only_upper_row.
where the constant *1* denotes the whole page, and *2–25* the layout components at the frame2 level.

There are three main differences with respect to document classification: in document understanding each document generates as many training examples as the number of layout components at the frame2 level. Furthermore, all pages of the document are of interest, not only the first one as in document classification. Finally, the number of learning problems equals the number of document classes.

The incremental learning system adopted in document understanding was INTHELEX (INcremental THEory Learner from EXamples) which induces *hierarchical* theories from examples [7]. INTHELEX is *fully incremental*: this means that, in addition to the possibility of taking as input a previously generated version of the theory, learning can also start from an empty theory and from the first available example. INTHELEX can learn simultaneously *multiple concepts*; furthermore, it is a *closed loop* learning system – i.e., a system in which the learned theory is checked to be valid on any new example and, in case of failure, a revision process is activated on it, in order to restore completeness and consistency. INTHELEX learns theories, expressed as sets of function-free clauses [10], from positive and negative examples. It assumes that, within a clause, different terms must denote different objects [17]. According to a *full memory storage* strategy, it retains all the processed examples, so that the learned theories are guaranteed to be valid on the whole set of known examples. It incorporates two refinement operators, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative ones.

After having prepared a set of training documents for each class, rules for document understanding can be learned for each concept in that document class. The user/trainer of WISDOM++ is asked to label layout components of a set of training documents according to their logical meaning. Those layout components with no clear logical meaning are not labeled (*undefined*). Therefore, each document generates a number of positive and negative instances depending on the number of layout components in the documents which constitute the set of observations. The *undefined* play the role of counterexamples for all the concepts to be learned.

Two experiments were carried out: the first on the set of 18 documents belonging to *faa_cen* class and the other on the set of 10 documents belonging to *dif_cen_decision* previously used for classification task. Each class of documents has its own concepts, i.e. layout components corresponding to meaningful logical components. Concepts that can be found in a *dif_cen_decision* are: a*ssessors, cens_authority, cens_signature, cert_signature, chairman, object_title, representative, rep_ producer, session_data.* In the *faa_cen* class, the logical components that can be learned are: *applicant*, *department*, *registration_au*, *date_place*, *reg_numb* and *authorization*.

Being the learning computational model conceptual and fully incremental, also a restricted number of training documents is sufficient for learning, provided that they are appropriately selected by the trainer as the most significant. Indeed, when learning labels for *faa_cen* documents, INTHELEX worked with an average of 10 examples for each concept and generated 2 clauses for *applicant*, *registration_au*, *date_place* and *authorization* and just one for *department* and *reg_numb*.

A learned rule for c*ens_authority* by the INTHELEX is:

```
logic_type_cens_authority(A) :-
  width_medium_large(A),type_of_text(A),
  pos_left(A),pos_upper(A),
  part_of(B,A),page_first(B),
  part_of(B,C),height_very_very_smal(C),
  type_of_text(C),pos_left(C),
  part_of(B,D),type_of_text(D),pos_upper(D)
  on_top(D,E),part_of(B,E),part_of(B,F),
  height_very_very_smal(F),type_of_text(F),
  part_of(B,G),part_of(B,H).
```

The meaning is: *A block A is of kind cens_authority* ***IF*** *"It appears at the top of the first page, in the left, is a text block, has a medium or wide size and in the same page there are other six components (C, D, E, F, G, H), two of which (C and F) are very small and of text type, being C on the left, and another component D, of text type, is placed in the upper part of the page and over a block E."*

Table 3 reports the testing results for understanding image documents for the class *dif_cen_decision*.

Table 3 A4-size DIF censorship cards: testing results

CONCEPT	Accuracy	CONCEPT	Accuracy
assessors	92	*object title*	100
cens_authoriy	100	*representative*	92
cens_signature	74	*rep_producer*	100
cert_signature	96	*session_data*	100
chairman	96		

5. OCR and transformation into HTML/XML format

After the layout structure has been mapped into the logical structure, OCR can be applied to some logical components of interest. Text read by the OCR is then associated to a layout component whose content type has already been determined. Thus, all data of concerning the result of document processing can be stored for future retrieval purposes. Moreover, by transforming the document into HTML/XML formats, it can be made accessible via Web. The realization of this transformation is explained in the following. An XML document has both a logical and a physical structure. The logical structure allows a document to be divided into named units and sub-units (*elements*). The physical structure allows components of the document (*entities*) to be named and stored separately, sometimes in other data files, so that information can be re-used, and non XML data (e.g. images) can be included by reference.

The most significant feature of XML is the concept of *D*ocument *T*ype *D*efinition (*DTD*), which *provides* a formal set of rules to define a logical document structure, *defines* the elements that may be used, and *dictates* where they may be applied in relation to each other. The declarations that comprise the DTD may be stored at the top of each document that must conform to these rules (*internal* DTD), or may be alternatively stored in a separate data file (*external* DTD), which is referred to by a *special instruction* at the top of each document. WISDOM++ adopts the latter solution, which generates, for each document class, a distinct DTD in which each declaration conforms to the markup declaration format <!…>. The keyword ELEMENT introduces an element declaration and specifies its allowed content. An attribute may be associated with an element in order to provide refined information on it. Examples of attributes are the URL, the format and the resolution of a document. All the attributes are declared separately from the element, but are usually declared together, in the *attribute list declaration*. It is also noteworthy that the DTD generated by WISDOM++ distinguishes the logical structure (logic) from the layout structure (geometric). The layout structure is used only for storing purposes: to render the document similar in appearance to the original document, we will use XSL specifications, as explained later.

Once the DTD has been defined, an instance of that document type can be generated and stored in a *.xml* file by respecting constraints defined by the set of rules in the DTD. The first row specifies the set of characters, the second row specifies the name of the style sheet file (with extension *.xsl*), while the third row defines the DTD associated to the document class (file with extension *.dtd*). Then, the file reports the specification of the logical structure. Text extracted with the OCR is intermixed with tags that define its logical structure. The specification of the geometric structure follows that of the logical one and is reported level by level.

The XML specification includes a facility for physically isolating and separately storing any part of a document. Each unit of information is called an *entity*, and each entity is assigned a name, so that it can be identified. The only entity to which an entity name is not assigned is the *document entity*. It is stored in a data file that is considered as representing the entire document. In simple cases, the document entity may be the only entity (main program without sub-programs), in more complex cases the document entity is used to position the call of other entities (main program with only sub-programs). A declaration <!ENTITY…> is required to announce the existence of an entity. No such declaration is reported in XML documents generated by WISDOM++, since they contain only one entity (document entity). The content of an XML element, such as abstract or paragraph, has no explicit text *style or format*. Since XML language is not concerned with visualization aspects, it is necessary to specify the element rendering in a different language. *XSL* (*eXtensible Style Language*) is a language used for expressing style sheets. An XSL style sheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

An *XSL style sheet processor* accepts a document or data in XML and an XSL style sheet and produces the presentation of that XML source content that was intended by the designer of that style sheet. The presentation process involves two distinct steps: the transformation of the original XML source file, and the interpretation of the transformed file to produce formatted results suitable for presentation. To sum up, several pieces of information extracted by WISDOM++, i.e. the layout and logical structures, the textual content of some logical components, and the pictorial content of some graphical layout components, are distributed into different files (DTD, XML, XSL and JPG). Nevertheless, the document conversion into structured formats (HTML/XML) is not straightforward, since a number of factors should be considered in order to render the converted document as similar as possible to the original document image. Layout-based conversion into HTML/XML format is detailed described in [2].

7. Conclusions and future work

In this paper, we presented the application of the document management system WISDOM++, already widely tested in the domain of digital libraries [8], to the problem of automatically processing documents available in film archives for the project COLLATE. With the goal of supporting complex working tasks such as historic film documentation and reconstruction of surrogates of lost or physically damaged films, the project aims at exploiting the chances of the proceeding digitization of cultural and historical document corpora by establishing innovative models and techniques of content-based organisation, handling and presentation of imperiled and precarious historical materials. The philosophy of the COLLATE project is mainly in its specific concept of a dynamic annotation and retrieval, based on the use of automated document processing techniques and content-based access methods, in the domain of heterogeneous multimedia repositories.

The transformation of documents into a digital format appropriate for a Web browser, is a complex knowledge-intensive process, involving document image analysis and machine learning techniques as well as multimedia editing techniques for rendering purposes. Machine learning is proposed as a viable solution to the problem of specifying the models used in text-graphics separation, layout analysis, document classification and understanding.

Promising results obtained in preliminary experiments on this kind of documents have been reported; nevertheless, they have to be confirmed by a larger experimentation. Moreover, there are problems that still need to be resolved. First, the extension of WISDOM++ with many segmentation algorithms that work properly on color images as well as on images of forms, where textual content is typically surrounded by frames or by vertical and horizontal lines. Second, we intend to perform a tight integration of the OCR with WISDOM++, in order to simplify the current user interaction and simultaneously take advantage of the font information extracted by the OCR during the HTML/XML rendering process.

8. References

[1] Aiello M., Monz C., Todoran L. and M. Worring (2002). Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition,* vol 5.

[2] Altamura, O.; Esposito, F. & Malerba, D. (2001). Transforming paper documents into XML format with WISDOM++. *International Journal on Document Analysis and Recognition*, vol. 4 pp. 2-17 .

[3] Ceri, S., Gottlob, G., & Tanca, L. (1989). What you always wanted to know about Datalog (and never dared to ask), *IEEE Transactions on Knowledge and Data Engineering*, 1, 1, pp. 146-166.

[4] Dengel, A. (2003). Making Documents Work: challenges for document understanding, *Proc. of Seventh International Conference on Document Analysis and Recognition,* pp.1026-1037.

[5] L. De Raedt (1992). Interactive Theory Revision. *Academic Press*, London.

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

[7] Esposito, F., Semeraro, G., Fanizzi, N. & Ferilli, S. (2000). Multistrategy Theory Revision: Induction and Abduction in INTHELEX. *Machine Learning Journal*, 38(1/2):133-156, Kluwer Academic Publisher.

[8] Esposito F., Malerba, D., Semeraro, G., Fanizzi, N., Ferilli, S. (1998). Adding Machine Learning and Knowledge Intensive Techniques to a Digital Library Service. *International Journal on Digital Libraries*, 2(1): 1-17, Springer Verlag, Berlin.

[9] S. Klink, A. Dengel, and T. Kieninger. Document structure analysis based on layout and textual features. In *Proc. of Fourth IAPR International Workshop on Document Analysis Systems, DAS2000*, pp 99–111.

[10] Lloyd, J.W. (1987). *Foundations of Logic Programming*. Springer-Verlag, Berlin.

[11] D. Malerba, F. Esposito, and F.A. Lisi (1998). Learning recursive theories with ATRE, in H. Prade (Ed.), *Proceedings of the 13th European Conference on Artificial Intelligence*, 435-439, John Wiley & Sons, England.

[12] D. Malerba, F. Esposito, F.A. Lisi, and O. Altamura (2001). Automated Discovery of Dependencies Between Logical Components in Document Image Understanding. *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, Seattle (WA), pp. 174-178.

[13] D. Malerba, F. Esposito, O. Altamura, M. Ceci, and M. Berardi. Correcting the document Layout: a Machine Learning Approach. *Proc. of Seventh International Conference on Document Analysis and Recognition,* pp.97-103.

[14] T.Mitchell (1997). *Machine Learning.* McGraw-Hill, New York.

[15] G. Nagy, S. Seth and M. Viswanathan (1992) A Prototype Document Image Analysis System for Technical Journal, *IEEE Computer*, vol. 25, no. 7, pp.10-22.

[16] Plotkin, G.D., (1971), A further note on inductive generalization, in *Machine Intelligence* 6, B.

[17] Semeraro, G., Esposito, F., Malerba, D., Fanizzi, N. & Ferilli, S. (1998). A Logic Framework for the Incremental Inductive Synthesis of Datalog Theories. In N. E. Fuchs (ed.), *Logic Program Synthesis and Transformation*, Lecture Notes in Computer Science 1463, (pp. 300-321), Springer, Berlin.

[18] Wong, K.Y.; Casey, R.G. & Wahl, F.M. (1982). Document Analysis System. *IBM Journal of Research Development*,26(6):647/656.