Multistrategy Learning of Rules for Automated Classification of Cultural Heritage Material

G. Semeraro, F. Esposito, S. Ferilli, N. Fanizzi,
T.M.A. Basile, and N. Di Mauro

Dipartimento di Informatica
Università di Bari
via E. Orabona, 4 - 70125 Bari - Italia
{semeraro, esposito, ferilli, fanizzi, basile, nicodimauro}@di.uniba.it

Abstract. This work presents the application of a new, enhanced version of the incremental learning system INTHELEX (INcremental THEory Learner from EXamples), the learning component in the architecture of the EU project COLLATE, dealing with the annotation of cultural heritage documents. Due to the complex shape of the handled material, the addition of multistrategy capabilities was needed to improve the effectiveness and efficiency of the learning process. Some results demonstrating the benefits that the addition of each strategy can bring are also reported.

1 Introduction: The COLLATE Project

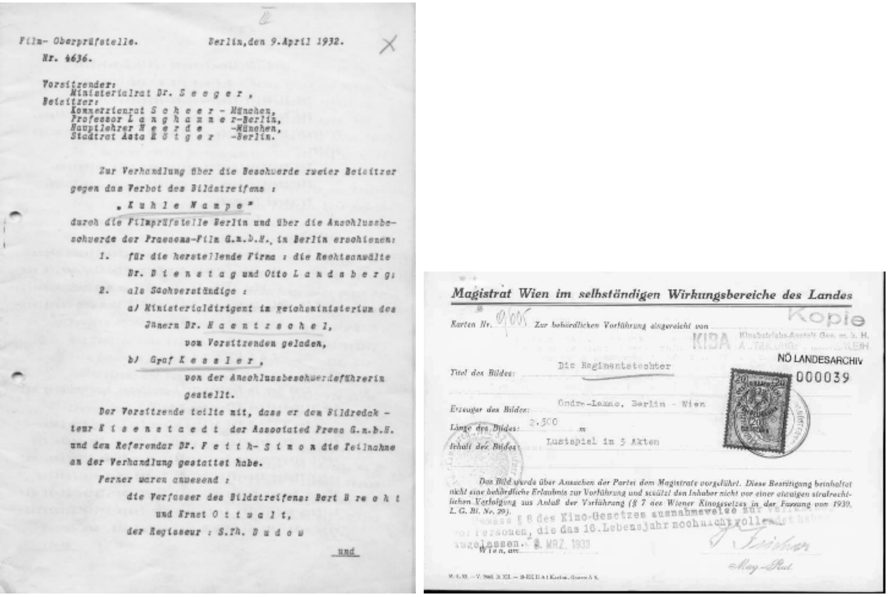
Many important historic and cultural sources, which constitute a major part of our cultural heritage, are fragile and distributed in various archives. Such a situation is an obstacle to full access, knowledge and usage. Also, many informal and non-institutional contacts between archives constitute specific professional communities, which today still lack effective and efficient technological support for cooperative and collaborative knowledge working. The COLLATE project[1] aims at developing a WWW-based *collaboratory* [7] for archives, researchers and end-users working with digitized historic/cultural material.

Though the developed tools and interfaces are generic, the chosen experimental domain concerns historic film documentation. Multi-format documents on European films of the early 20th century, provided by three major national film archives, include a large corpus of rare historic film censorship documents from the 20s and 30s, as well as newspaper articles, photos, stills, posters and film fragments. An in-depth analysis and comparison of such documents can give evidence about different film versions and cuts, allow restoration of lost/damaged films, and identify actors and film fragments of unknown origin.

All material is analyzed, indexed, annotated and interlinked by film experts. The COLLATE system will provide suitable task-based interfaces and knowledge

[1] IST-1999-20882 project COLLATE - *Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material* (URL: `http://www.collate.de`).

E.-P. Lim et al. (Eds.): ICADL 2002, LNCS 2555, pp. 182–193, 2002.



Film- Oberprüfstelle. Berlin, den 9. April 1932.

Nr. 4636.

Vorsitzender:
Ministerialrat Dr. Seeger,
Beisitzer:
Kommerzienrat Scheer - München,
Professor Langhammer-Berlin,
Hauptlehrer Heerde -München,
Stadtrat Asta Rötger -Berlin.

Zur Verhandlung über die Beschwerde zweier Beisitzer gegen das Verbot des Bildstreifens :

„Kuhle Wampe"

durch die Filmprüfstelle Berlin und über die Anschlussbeschwerde der Praesens-Film G.m.b.H., in Berlin erschienen:

1. für die herstellende Firma : die Rechtsanwälte Dr. Dienstag und Otto Landsberg;
2. als Sachverständige :
a) Ministerialdirigent im Reichsministerium des Innern Dr. Haentzschel,
vom Vorsitzenden geladen,
b) Graf Kessler,
von der Anschlussbeschwerdeführerin gestellt.

Der Vorsitzende teilte mit, dass er dem Bildredakteur Eisenstaedt der Associated Press G.m.b.H. und dem Referendar Dr. Feith-Simon die Teilnahme an der Verhandlung gestattet habe.

Ferner waren anwesend :
die Verfasser des Bildstreifens: Bert Brecht
und Ernst Ottwalt,
der Regisseur : S.Th. Dudow

und

Magistrat Wien im selbständigen Wirkungsbereiche des Landes

Kopie

Karten Nr. 9/005 Zur behördlichen Vorführung eingereicht von

KIBA Kinobetriebs-Anstalt Ges. m. b. H.
A. TEILUNG: [illegible]

NÖ LANDESARCHIV
000039

Titel des Bildes: Die Regimentstochter

Erzeuger des Bildes: Ondra-Lamac, Berlin - Wien

Länge des Bildes: 2.500 m

Inhalt des Bildes: Lustspiel in 5 Akten

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. 20).

Gemäss § 8 des Kino-Gesetzes ausnahmsweise zur Vorführung vor Personen, die das 16. Lebensjahr noch nicht vollendet haben zugelassen.

Wien, am 8. MÄRZ 1933

Mag.-Rat

M. 6. 83. — V. 1068. 31. XII. — [illegible]

(a) (b)

Fig. 1. Sample COLLATE documents

management tools to support both individual work and collaboration. Continuously integrating newly derived user knowledge into its digital data and metadata repositories, the system can offer an improved content-based retrieval functionality. Enabling users to create and share valuable knowledge about the cultural, political and social contexts allows in turn other end-users to better retrieve and interpret the historic material.

Supported by successful experience in the application of symbolic learning techniques to the classification and understanding of paper documents [4,6,11], our aim is applying INTHELEX to these documents. The objective is learning to automatically identify and label document classes and significant components, to be used for indexing/retrieval purposes and to be submitted to the COLLATE users for annotation. Combining results from the manual and automatic indexing procedures, elaborate content-based retrieval mechanisms can be applied [2]. The challenge comes from the low layout quality and standard of such materials, which introduce a considerable amount of noise in their description (see Figure 1). Regarding the layout quality, it is often affected by manual annotations, stamps that overlap sensible components, ink specks, etc.. For the layout standard, many documents are typewritten sheets, that consist of all equally spaced lines in Gothic type. Such a situation should account for a profitable use of automated reasoning capabilities in INTHELEX, such as abduction and abstraction. While the former can make the system more flexible in the absence of particular layout components due to the typist's style, the latter can help in focusing on layout patterns that are meaningful to the identification of interesting ones, neglecting less interesting details. Preliminary experiments showed that INTHELEX is able to distinguish at least 3 classes of COLLATE censorship documents, and to single out a number of logical components inside them. For instance, it learns rules that can separate the censorship authority, applicant and decision in documents like the one in Figure 1-b.

The following section presents the system INTHELEX along with its multistrategy capabilities. Section 3 describes the experiment and discusses the results. Lastly, Section 4 draws some conclusions and outlines future work directions.

2 INTHELEX: The Learning Component

Incremental learning is necessary when either incomplete information is available at the time of initial theory generation, or the nature of the concepts evolves dynamically. Both cases are very frequent in real-world situations, hence the need for incremental models to complete and support the classical batch ones, that perform learning in one step and thus require the whole set of observations to be available from the beginning.

INTHELEX (INcremental THEory Learner from EXamples) is a learning system for the induction of *hierarchical* logic theories from examples [5]: it learns theories expressed in a first-order logic representation from positive and negative examples; it can learn simultaneously *multiple concepts*, possibly related to each other (recursion is not allowed); it retains all the processed examples, so to guarantee validity of the theories learned from all of them; it is a *closed loop* learning system (i.e. a system in which feedback on performance is used to activate the theory revision phase [1]); it is *fully incremental* (in addition to the possibility of refining a previously generated version of the theory, learning can also start from an empty theory); it is based on the *Object Identity assumption* (terms, even variables, denoted by different names within a formula, must refer to different objects)[2].

INTHELEX incorporates two refinement operators, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative examples. It exploits a (possibly empty) previous version of the theory, a graph describing the dependence relationships among concepts, and a historical memory of all the past examples that led to the current theory. Whenever a new example is taken into account, it is stored in such a repository and the current theory is checked against it.

If it is positive and not covered, generalization must be performed. One of the definitions of the concept the example refers to is chosen by the system for generalization. If a generalization can be found that is consistent with all the past negative examples, then it replaces the chosen definition in the theory, or else another definition is chosen to be generalized. If no definition can be generalized

[2] This often corresponds to human intuition, while allowing the search space to fulfill nice properties affecting efficiency and effectiveness of the learning process [10].

in a consistent way, the system checks if the exact shape of the example itself can be regarded as a definition that is consistent with the past negative examples. If so, it is added to the theory, or else the example itself is added as an exception.

If the example is negative and covered, specialization is needed. Among the theory definitions involved in the example coverage, INTHELEX tries to specialize one at the lowest possible level in the dependency graph by adding to it positive information, which characterize all the past positive examples and can discriminate them from the current negative one. In case of failure on all of the considered definitions, the system tries to add negative information that is able to discriminate the negative example from all the past positive ones, to the definition related to the concept the example is an instance of. If this fails too, the negative example is added to the theory as an exception. New incoming observations are always checked against the exceptions before applying the rules that define the concept they refer to.

Another peculiarity in INTHELEX is the integration of other forms of automated reasoning, that may help in the solution of the theory revision problem by pre-processing the incoming information [6]. Namely, deduction is exploited to fill observations with information that is not explicitly stated, but is implicit in their description. There is thus the possibility of better representing the examples and, consequently, the inferred theories. Conversely, abduction aims at completing possibly partial information in the examples (adding more details), whereas abstraction removes superfluous details from the description of both the examples and the theory. Thus, even if with opposite perspectives, both aim at reducing the computational effort required to learn a correct theory with respect to incoming examples.

2.1 Deduction

INTHELEX requires the observations to be expressed only in terms of the predicates that make up the description language for the given learning problem. To ensure uniformity of the example descriptions, such predicates have no definition. Nevertheless, since the system is able to handle a hierarchy of concepts, combinations of these predicates might identify higher level concepts that are worth adding to the descriptions in order to raise their semantic level. Thus, INTHELEX implements a saturation operator that exploits deduction to recognize such concepts and explicitly add them to the description of the examples.

The system can be provided with Background Knowledge containing (complete or partial) definitions in the same format as the theory rules. The background knowledge is supposed to be correct, and hence is not modifiable. This way, any time a new example is considered, a preliminary saturation phase can be performed, that adds the higher level concepts whose presence can be deduced from such rules by subsumption and/or resolution. Differently from abstraction, described below, all the specific information used by saturation is left in the example description. Hence, it is preserved in the learning process until other evidence reveals it is not significant for the concept definition, which is a more cautious behaviour. This is fundamental if some concepts to be learnt are related, since their definition could not be stable yet, and hence one cannot afford to drop the source from which deductions were made in order to be able to recover from deductions made because of wrong rules.

2.2 Abduction

Induction and abduction are both important strategies to perform hypothetical reasoning (i.e., inferences from incomplete information). Induction yields the inference, from a certain number of significant observations, of regularities and laws valid for the whole population. Abduction was defined by Peirce as hypothesizing some facts that, together with a given theory, could explain a given observation.

According to the framework proposed in [8], an *abductive logic theory* is made up of a normal logic program [9], a set of *abducibles* and a set of *integrity constraints* (each corresponding to a combination properties/relations that is not allowed to occur). Abducibles are the predicates about which assumptions (*abductions*) can be made: They carry all the incompleteness of the domain (if it were possible to complete these predicates, then the theory would be correctly described). Integrity constraints provide indirect information about them and, since several explanations may hold for this problem setting, are also exploited to encode preference criteria for selecting the best ones.

The proof procedure implemented in INTHELEX starts from a goal and a set of initial assumptions, and results in a set of consistent hypotheses (abduced literals) by intertwining *abductive* and *consistency derivations*. Intuitively, an abductive derivation is the standard Logic Programming derivation suitably extended in order to consider abducibles. As soon as an incompleteness is detected in an observation, the corresponding information, if abducible, is added to the observation itself, provided that any related integrity constraint is satisfied. This is checked by starting a consistency derivation. Each integrity constraint related to the abduced fact is considered satisfied if at least one of its components does not hold. In the consistency derivation, when an abducible is encountered, an abductive derivation for its complement is started in order to prove its falsity.

This procedure can be exploited in order to complete the document descriptions contained in the examples, so that the system is less sensitive to missing information (e.g., missing layout features).

2.3 Abstraction

Abstraction is a pervasive activity in human perception and reasoning. When we are interested in the role it plays in Machine Learning, inductive inference must be taken into account as well. The exploitation of abstraction concerns the shift of representation languages from the language in which the theory is described to a higher level one.

According to the framework proposed in [12], concept representation deals with entities belonging to three different levels. Concrete objects reside in the *world*, but any observer's access to it is mediated by his *perception* of it. To be available over time, these stimuli must be memorized in an organized *structure*, i.e. an *extensional* representation of the perceived world. Finally, to reason about the perceived world and communicate with other agents, a *language* is needed, that describes it *intensionally*. Modifications to the structure and language are just a consequence of differences in the perception of the world (due, e.g., to the medium used and the focus-of-attention). Thus, abstraction takes place at the world-perception level, and then propagates to higher levels, by means of a set of operators. An abstraction theory contains information for performing the shift specified by the abstraction operators.

In INTHELEX, it is assumed that the abstraction theory is already given (i.e. it has not to be learned by the system), and that the system automatically applies it to the learning problem at hand before processing the examples. The implemented abstraction operators allow the system to replace a number of components by a compound object, to decrease the grain-size of a set of values, to ignore whole objects or just part of their features, and to ignore the number of occurrences of some kinds of object.

3 Experiments

Some experiments were run to test the improvement coming from the addition of abduction and abstraction to the process of learning definitions, for some classes of censorship documents provided by FilmArchiv Austria (FAA) and Deutsches FilmInstitut (DIF). Specifically, we used registration cards coming from the former ('`faa_registration_card`', see Figure 1-b) and censorship decisions coming from the latter ('`dif_censorship_decision`', see Figure 1-a). The dataset consisted of 34 documents for the class `faa_registration_card`[3], 19 documents for the class `dif_censorship_decision`[4] and 61 reject documents, obtained from newspaper articles and DIF registration cards. Note that the symbolic method adopted allows the trainer to specifically select prototypical examples to be included in the learning set. This explains why theories with good predictiveness can be obtained even from few observations.

The first order descriptions of such documents, needed to run the learning system, were automatically generated by the system WISDOM++ [3]. Starting from scanned images, such a system is able to identify the layout blocks that make up a paper document, along with their type and relative positions. Each document was then described in terms of its composing layout blocks, along with their size

[3] A certification that the film has been approved for exhibition in the present version by the censoring authority. The "registration cards" were given to the distribution company, who had to pay for this. They enclosed the cards with the prints. The police checked the cinemas from time to time, and the owner or projectionist had then to show the registration card.

[4] Decision whether a film could or could not, and in which version, be distributed and shown throughout a country. The "censorship decision" is often a protocol of the examination meeting and is issued by the censorship office or headquarters.

Table 1. `faa_registration_card` Classification

	Clauses	Length	lgg	Runtime	Accuracy	Pos	Neg	t-test
'Manual' Discretization	1	10,5	9,5	11,9	0,98	0,94	1,0	1,0
Numeric abstraction	1	13,1	8	19,29	0,99	0,97	1,0	1,0
Speckle abstraction	1	12,9	8,7	18,24	0,98	0,94	1,0	1,0
Abduction	1,1	13,1	8,7	123,69	0,99	0,97	1,0	1,0
	(2)	(40,7)	(9,7)	(40,9)	(0,99)	(0,97)	(1,0)	(1,0)
Abduction+Abstraction	1	12,5	8	90,13	1,0	1,0	1,0	1,0
	(2)	(42)	(9)	(36,42)	(0,99)	(0,97)	(1,0)	(1,0)

Table 2. `dif_censorship_decision` Classification

	Clauses	Length	lgg	Runtime	Accuracy	Pos	Neg	t-test
'Manual' Discretization	1,6	52,3	8	71,92	0,94	0,74	0,99	0,98
Numeric abstraction	1	23,3	8,4	49,6	0,97	0,84	1,0	0,99
Speckle abstraction	1	23,3	8,6	47,64	0,97	0,84	1,0	0,99
Abduction	1,1	28,3	8,7	5667,8	0,95	0,74	1,0	0,98
	(1,2)	(33,4)	(8,7)	(50,37)	(0,95)	(0,74)	(1,0)	(0,97)
Abduction+Abstraction	1,1	25,8	9,2	5761,8	0,96	0,79	1,0	0,99
	(1,2)	(33,4)	(8,7)	(50,37)	(0,94)	(0,74)	(0,99)	(0,99)

(height and width), position (horizontal and vertical), type (text, line, picture and mixed) and relative position (horizontal/vertical alignment, adjacency). The description length of the documents for class `faa_registration_card` ranges between 40 and 379 literals (144 on average); for class `dif_censorship_decision`, it ranges between 54 and 263 (215 on average).

Each document was considered as a positive example for the class it belongs to, and as a negative example for the other class (to be learned from); reject documents were considered as negative examples for both classes. Definitions for each class were learned, starting from the empty theory and with all the negative examples at the beginning (in order to simulate a batch approach), and their predictive accuracy was tested according to a 10-fold cross validation methodology, ensuring that all the learning and test sets contained the same proportion of positive and negative examples.

3.1 Experimental Baseline

Various experiments were performed on both classes, whose results (all averaged on the 10 runs) are reported in Table 1 (as regards `faa_registration_card`) and in Table 2 (concerning `dif_censorship_decision`). For each case, the following data are reported - *Clauses* : number of clauses (i.e., alternative definitions for the concept) in the learned theory; *Length* : number of literals composing the clauses; *lgg* : number of generalizations needed to obtain the theory; *Runtime* : computational time required to learn the theory (expressed in seconds); predictive accuracy rate computed on the test set (*Accuracy* overall, *Pos* on positive examples only, *Neg* on negative examples only); *t-test* : expected predictive accuracy according to a t-test with confidence level $\alpha = 0,05$.

In the following paragraphs, we will discuss the outcomes in more detail, and try to justify the conclusions we draw.

A preliminary problem was the fact that INTHELEX is currently unable to handle numeric descriptors, whereas WISDOM++ expresses the blocks' dimensions and positions as numeric values (number of pixels). Hence, a discretization was needed to assign each specific value to a symbolic label representing an interval. When abstraction had not yet been added to INTHELEX, we had to perform such a transformation through a purposely implemented routine. Now, we are able to delegate this task to the system itself, by exploiting one of its abstraction operators (the one acting on the grain size). Comparing the first two rows of Tables 1 and 2, which report on this performance in the two cases, we note that there is no loss in the 'Numeric abstraction' case (second row) compared to 'Manual discretization' (first row). Actually, there is a slight improvement except for computational time for the first case, probably due to abstraction changing the ordering of the literals in the observations. Thus, in all the subsequent experiments, INTHELEX ran the numeric discretization phase by abstraction.

3.2 Motivation for Multistrategy Learning in COLLATE

Since the available documents were often affected by the presence of speckles, identified by WISDOM++ as layout components and hence appearing in the description of the document, we decided to use abstraction to eliminate them. The underlying rationale is that such a 'cleaning' should hopefully help the system in at least two respects. First, by focusing on significant layout components, that are more discriminant, this should lead to more characterizing definitions of the concepts. Second, having shorter example descriptions can have a positive effect on the learning time. Specifically, the abstraction theory considered as speckles all the blocks without a clear type (mixed) that are short and/or narrow. Another issue to be faced in the COLLATE project is the low quality of some documents, due to their age and to the absence, in many cases, of a standard layout. While the documents in our dataset were chosen to have an acceptable quality and a sufficiently standard layout, it is foreseeable that worse documents will be available in the future. To simulate the behaviour of INTHELEX in such a possible scenario, we corrupted part of the documents in the dataset, and tried to apply abduction in order to overcome the problems raised by missing components. Specifically, incomplete documents were generated by randomly dropping 10% of the description from 30% of the available documents, and then letting INTHELEX use abduction. All the basic predicates in the description language concerning block dimensions, types and positions were considered as abducibles, while integrity constraints were set to express the mutual exclusion among layout block sizes, types and positions. INTHELEX was allowed to exploit abduction to hypothesize facts, concerning the above descriptors, only in case of failure in finding a correct generalization, before adding a new clause to the theory.

Abduction makes sense in this environment since the absence of a layout block in a document could be due to the writer not fulfilling the style requirements, and not to the insignificance of that block for a correct definition. In other words, a block should not be dropped from the definition just because a few examples miss it; conversely, integrity constraints are in place to ensure that superfluous blocks found in the first few examples do not introduce unnecessary blocks that can be always abduced in the future. The last three rows of Tables 1 and 2 report the experiments with speckles abstraction only, abduction on the corrupted dataset only, and both, respectively (in the last case, abstraction precedes abduction). Rows involving abduction also report, in parentheses for comparison purposes, the performance obtained on the corrupted dataset without exploiting abduction.

3.3 Discussion of Experimental Results

Let us first focus on the experiments concerning the `faa_registration_card` class (see Table 1). Abstraction of speckles cannot, of course, improve the number of clauses (that is already 1). Nevertheless, the shorter example descriptions have a beneficial effect on the learning time, in spite of the greater number of generalizations performed (probably due to the absence of speckles in negative examples, that formerly helped to avoid their coverage). The greater difficulty in avoiding coverage of negative examples also results in a more specific clause, as showed by the slight decrease in predictive accuracy on positive examples.

As for abduction, the experiments prove that it is able to balance the corruption in the examples, even if at the cost of a slightly worse number of clauses and lggs, and of a significant increase in the runtime (due to the necessity of checking the consistency of each hypothesized fact). Indeed, predictive accuracy is the same as when only inductive operators are employed, even if a great portion of the descriptions is now missing. The benefit becomes more evident, especially as regards the number of clauses and their length, if we compare the performance to what would be obtained on the corrupted dataset without exploiting abduction, as reported in parentheses.

Finally, the results of the joint effort of abduction and abstraction are shown in the last row. The theories learned in this case are, indeed, the best of all cases: they outperform all the previous ones as regards predictive accuracy (100%), and are better than any single strategy as to the number of clauses and lgg's performed. Also the runtime, even though worse than that of abstraction only, is far better than that of abduction only. The average number of literals in each clause is almost stable, with a slight decrease when abstraction is used, signifying that the system was always able to grasp the core concept and that abstraction actually eliminated only superfluous information.

Considering the experiments on the other class, `dif_censorship_decision` (see Table 2), we immediately note that runtime is always higher than for the previous class, while the predictive accuracy is always lower (even if still very high). The former suggests that we are facing an intrinsically harder learning problem (indeed, documents in this class have a larger size and a higher number of identified layout components – typically, each row in the document is considered a separate block). The latter may be due to the availability of fewer examples, leading to a less refined theory that is not predictive enough, as supported by the number of literals in the definitions. It seems that more characteristics than in the other class must be preserved to find a solution. Indeed, the portion of the original example length that is dropped ranges from 85,68% to 89,17% (against the 90,91–92,71% for the other class). The above comments concerning speckle abstraction generally still hold, including the improvement that it can bring when added to abduction.

On the other hand, this class seems to be particularly idiosyncratic with respect to abduction. In this respect, it should be noted that runtime in the last two rows is heavily affected by the search for useful abductions in one fold, as proved by the fact that, if we restrict the search to the remaining 9 folds, the mean decreases to 54,87 and 38,87, respectively. The poor performance of abduction in this case seems to be confirmed by comparison with the statistics in parentheses, particularly as regards the clause length, whereas the other class showed a significant improvement. This could be due to the fact that corrupting the descriptions makes an already difficult problem even more difficult. Another possible explanation is the fact that corrupting the description of documents in this class results in the elimination of many literals. This can be compensated up to some extent by the generalizations, that are able to find other common (but probably less characterizing and meaningful) features among the given documents. Such accidental similarities could cause overfitting on the training data, that abduction is not able to compensate during the test phase. This is confirmed by the lower predictive accuracy being concentrated in the coverage of positive examples, whereas negative examples are always rejected.

3.4 Insight into Multistrategy Operators' Performance

The difference in effectiveness of abduction in the two classes led us to closely examine the phenomenon, in order to better understand it and its possible causes. Table 3 summarizes, for both classes, the average number of examples on which literals have been abduced (first row) and the average number of literals abduced for each of them (second row), both with and without abstraction of speckles. Our expectations were confirmed in that, for the class `dif_censorship_decision`, nearly no abduction was made, which explains why no improvement was obtained with respect to the other cases (including learning without abduction on corrupted examples). More specifically, we found that the only abductions were carried out for the 'computation-intensive' fold, which again suggests that this class seldom requires abduction, and in those cases it is a hard task anyway. On the contrary, as regards class `faa_registration_card`, abduction improves performance since it succeeds on hypothesizing more literals (2 out of about 13 that make up the definition) and on more examples.

Similarly, we have collected in Table 4 information on the effects of eliminating speckles from document descriptions through abstraction. For both types of documents, the number of examples in which abstraction took place is reported

Table 3. Abduction performance

	faa_registration_card		dif_censorship_decision	
	w/ abstraction	w/o abstraction	w/ abstraction	w/o abstraction
Examples	1,9	1,6	0,1	0,1
Literals	1,53	1,5	0,2	0,2

Table 4. Speckles abstraction performance

	faa_registration_card	dif_censorship_decision	Reject
Examples	14	9	3
Literals	114	37	140

(first row), along with the average number of dropped literals (second row). It turns out that only a part of the whole training set suffered from speckles, and in those cases a relevant portion of the descriptions was removed. It is noticeable that more speckles were found in faa_registration_card documents than in dif_censorship_decision ones, even if the average description length of the latter was larger than that of the former, which indicates a better layout quality in the latter. This also means that abstraction cannot lower the document descriptions' complexity for the latter class, which results in harder work for the other operators.

4 Conclusions and Future Work

Multistrategy approaches to machine learning can help to improve efficiency, and are necessary in a number of real-world situations. The incremental learning system INTHELEX works on first-order logic representations. Its multistrategy learning capabilities have been further enhanced to improve effectiveness and efficiency of the learning process, by augmenting pure induction and abduction with abstraction and deduction.

This paper presented and discussed some experimental results demonstrating the benefits that the addition of each strategy can bring to the task of document classification. Even if the performance obtained exclusively by the inductive operator was very good, multistrategy operators contributed to make it even better from both the effectiveness and the efficiency viewpoints. INTHELEX is included in the architecture of the EU project COLLATE, to learn rules for automated classification of cultural heritage documents dating back to the 20s and 30s.

Future work will concern more extensive experimentation, aimed at finding tighter ways of cooperation among the learning strategies. It will also be interesting to apply the same techniques to learn rules for interpreting the semantic role played by meaningful layout components in the documents. An analysis of the complexity of the presented techniques is also planned. Moreover, the addition of numeric capabilities can be considered fundamental for effective learning in some contexts, and hence deserves further study.

Acknowledgments. This work was partially funded by the EU project IST-1999-20882 COLLATE "Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material".

References

[1] J. M. Becker. Inductive learning of decision rules with exceptions: Methodology and experimentation. B.s. diss., Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1985. UIUCDCS-F-85-945.

[2] H. Brocks, U. Thiel, A. Stein, and A. Dirsch-Weigand. Customizable retrieval functions based on user tasks in the cultural heritage domain. In P. Constantopoulos and I.T. Sølvberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in LNCS. Springer, 2001.

[3] 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.

[4] F. Esposito, D. Malerba, G. Semeraro, N. Fanizzi, and S. Ferilli. Adding machine learning and knowledge intensive techniques to a digital library service. *International Journal on Digital Libraries*, 2(1):3–19, 1998.

[5] F. Esposito, G. Semeraro, N. Fanizzi, and S. Ferilli. Multistrategy Theory Revision: Induction and abduction in INTHELEX. *Machine Learning Journal*, 38(1/2):133–156, 2000.

[6] S. Ferilli. *A Framework for Incremental Synthesis of Logic Theories: An Application to Document Processing.* Ph.D. thesis, Dipartimento di Informatica, Università di Bari, Bari, Italy, November 2000.

[7] R.T. Kouzes, J.D. Myers, and W.A. Wulf. Collaboratories: Doing science on the internet. *IEEE Computer*, 29(8), 1996.

[8] E. Lamma, P. Mello, F. Riguzzi, F. Esposito, S. Ferilli, and G. Semeraro. Cooperation of abduction and induction in logic programming. In A. C. Kakas and P. Flach, editors, *Abductive and Inductive Reasoning: Essays on their Relation and Integration.* Kluwer, 2000.

[9] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Berlin, second edition, 1987.

[10] G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of datalog theories. In N. E. Fuchs, editor, *Logic Program Synthesis and Transformation*, number 1463 in LNCS. Springer, 1998.

[11] G. Semeraro, S. Ferilli, N. Fanizzi, and F. Esposito. Document classification and interpretation through the inference of logic-based models. In P. Constantopoulos and I.T. Sølvberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in LNCS. Springer, 2001.

[12] J.-D. Zucker. Semantic abstraction for concept representation and learning. In R. S. Michalski and L. Saitta, editors, *Proceedings of the 4th International Workshop on Multistrategy Learning*, Desenzano del Garda, Italy, 1998.