

Improving Automatic Labelling through RDF Management

Floriana Esposito, Stefano Ferilli, Nicola Di Mauro, Teresa M.A. Basile,
Luigi Iannone, Ignazio Palmisano, and Giovanni Semeraro

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

Abstract. Building a shared and widely accessible repository, in order for scientists and end users to exploit it easily, results in tackling a variety of issues. Among others, the need for automatic labelling of available resources arises. We present an architecture in which machine learning techniques are exploited for resources classification and understanding. Furthermore, we show how learning tasks can be carried out more effectively if training sets and learned theories are expressed by means of Resource Description Framework (RDF) formalism and the storage/retrieval/query operations are managed by an ad hoc component.

1 Introduction

Serious problems to full access, knowledge and exploitation of historic and cultural sources are caused both by the fragility of the medium on which they are stored and by their being distributed in various archives throughout the world. Moreover, professional communities working in the cultural heritage field today still lack effective and efficient technological support for cooperative and collaborative knowledge working, thus often rely on informal and non-institutional contacts. A solution to the former problem may come from a process of digitization, that transposes the document in electronic form. Conversely, the latter problem can be overcome by exploiting the Internet as an infrastructure to retrieve/access such a material and bridge the distance among researchers.

The COLLATE project¹ aims at developing a WWW-based *collaboratory* [7] for archives, researchers and end-users working with digitized historic/cultural material. Specifically, it aims at providing suitable task-based interfaces and knowledge management tools to support experts in their individual work and collaboration to analyze, index, annotate and interlink all material. New user knowledge can be continuously integrated into the COLLATE digital data and metadata repositories, so that the system can offer improved content-based retrieval functionality and enable

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

users to create and share valuable knowledge about the cultural and social contexts, allowing in turn other end-users to better retrieve and interpret the historic material.

The need of automatically labelling the huge amount of documents in the COLLATE repository suggested the use of Machine Learning techniques to automatically identify document classes and label 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 a material, which introduces a considerable amount of noise in its description. As regards the layout quality, it is often affected by manual annotations, stamps that overlap to sensible components, ink specks, etc. As to the layout standard, many documents are typewritten sheets, that consist of all equally spaced lines in Gothic type.

This paper is organized as follows. Next section describes the COLLATE system architecture. The documents in the collection that are to be automatically labelled by the embedded learning component are described in Section 3 along with its performance on such a dataset. Then, Section 4 describes the metadata management component in charge of handling all the information flowing in the system, and its interaction with the learning component. Conclusions are drawn in Section 5.

2 The COLLATE Environment

The overall architecture of the COLLATE system (Fig. 1) is structured into several functional layers: *Operational Layer*, *Domain Metadata Layer*, *Task Layer*, *Interface Layer*. The Operational Layer can be described as a digital data repository that comprises a variety of data, ranging from scanned-in text documents to multimedia data and the accumulated annotations related to one or more of these original data. The Domain Metadata Layer provides suitable tools for metadata management in order to organize the stored data in a way that supports the complex knowledge-intensive tasks users will want to perform on the repository contents. The knowledge structures, which are represented by specific XML schemata, constitute the Domain Model. The Task Layer concerns the possibility for the wide variety of the COLLATE types of users to access, work with and evaluate the digitized archive material. In order to support the users in accomplishing their tasks, COLLATE provides appropriate interfaces for convenient work with the digital documents, the Interface Layer. All the information handled by the system is encoded in XML and RDF² format, which allows the various modules to communicate by a single representation language; thus, communication between the layers is realized through XML-based communication protocols, as well.

Now, let us briefly summarize the workflow within the system, focusing our attention on the Document Pre-Processing and XML Content Manager modules. After the historical documents are scanned in, they pass through a pre-processing module which extracts from them relevant information using labelling rules automatically generated by a learning component. The meta-information discovered in this phase is stored together with the Web version of the corresponding documents by the XML

² Resource Description Framework <http://www.w3.org/RDF>

Content Manager, which serves as mediating management layer between the COLLATE system and its underlying databases. It will be used for document retrieval, also managed by the *XML Content Manager*.

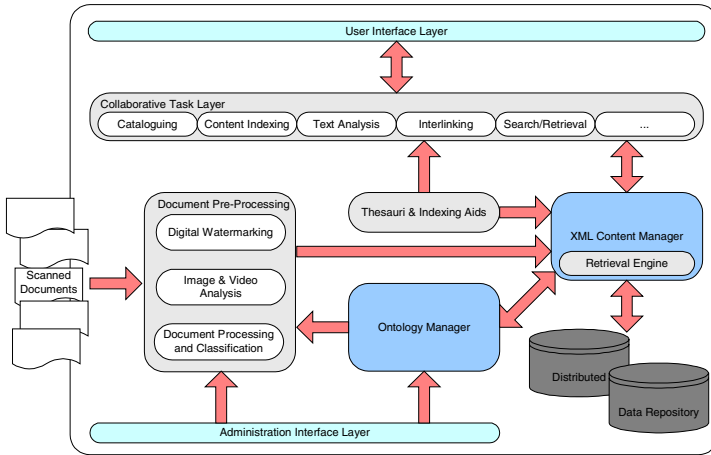


Fig. 1. Overall architecture of the COLLATE system

Though the developed tools and interfaces are generic, the COLLATE experimental domain concerns historic film documentation. Multi-format documents on European films dating from early 20th century are provided by three major European national film archives: DIF (Deutsches Filminstitut, Frankfurt am Main), FAA (Film Archive Austria, Vienna) and NFA (Národní Filmový Archiv, Prague). Such a collection includes a large corpus of rare historic film censorship documents from the 20ies and 30ies, but also 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, and allow to restore lost/damaged films or identify actors and film fragments of unknown origin.

Specifically, the COLLATE repository (Fig. 2 shows examples of documents) includes several thousands comprehensive documents concerning film culture, and focuses on documents related to censorship processes. The importance of censorship for film production distribution lies mainly in the fact that it is often impossible to identify a unique film. Often, there are lots of different film versions with cuts, changed endings and new inter-titles, depending on the place and date of release. Exactly these differences are documented in censorship documents and allow statements about the original film. They define and identify the object of interest. Often they provide the only source available today for the reconstruction of the large number of films that have been lost or destroyed. Censorship documents support this restoration process by identifying and structuring the film fragments. They allow to put together film fragments from various copies in order to obtain a correct reconstruction. Each country developed its own censorship history embedded in the political history. The collection is complemented by further documents like press articles, correspondence, photos, etc.

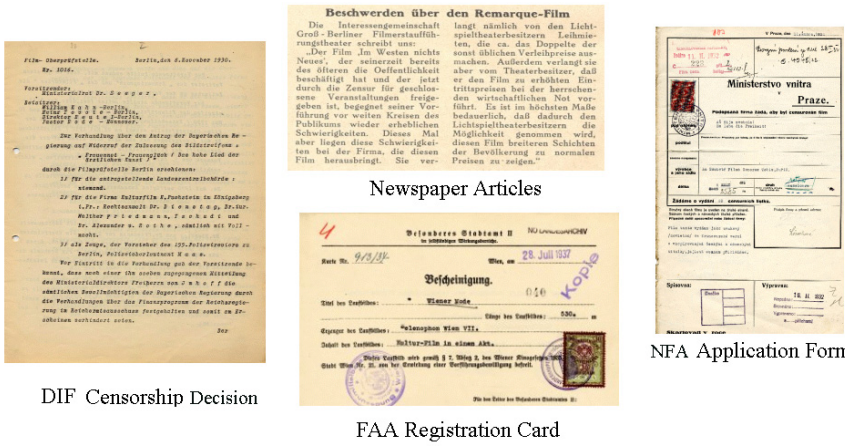


Fig. 2. Sample COLLATE documents

Typical documents analyzed are: Application Forms, Censorship Decisions, Registration Cards. The first kind of documents, Application Form, was required for applying to get the permission to show a film from a production or distribution company. The consequence of this application was the examination by the censorship office. It was usually accompanied by a list of intertitles or dialogue list that served to check whether a film shown in the cinema was the same as the one examined by the censorship office. As regards the Censorship Decision documents, they are about the permission of distributing or showing a film (with relative version) throughout a country. The Registration Card is a certification that the film had been approved for exhibition in the present version by the censoring authority. The registration cards were given to the distribution company which had to pay for this, and had to enclose the cards to the film copies. When the police checked the cinemas from time to time, the owner or projectionist had to show the registration card. Such cards constitute a large portion of the COLLATE collection and are an important source for film reconstruction. All these documents could be a source for information like: *Name of applicant* (production or distribution company), *title of the film*, *year of production*, *length* (before censorship), *brief content*, *information about earlier examinations*, *participants in the examination*, *juridical legitimization for the decision*, *conditions for permission* (for example cuts, change of title, etc.), *reference to previous decisions*, *costs for the procedure*, *production company*, *date and number of examination*, *number of acts*, *forbidden parts*. There are also a lot of documents from the contemporary film press, newspapers or magazines: They are necessary to reconstruct the context of a film, since they enlighten the reception background.

3 The Embedded Learning Component

Supported by previous successful experience in the application of symbolic learning techniques to classification and understanding of paper documents [3,5,9], we focused our attention on first-order logic learning techniques, whose high level representation

can better manage the complexity of the task and allows the use of different reasoning strategies than pure induction with the objective of making the learning process more effective and efficient, to learn rules for such tasks from a small number of selected and annotated sample documents.

3.1 INTHELEX

INTHELEX³ (INcremental THEory Learner from EXamples) [4] is the system embedded in the COLLATE architecture as a learning component. It carries out the induction of *hierarchical* first-order logic theories from positive and negative examples: it learns simultaneously *multiple concepts*, possibly related to each other; it guarantees validity of the theories on all the processed examples; it uses feedback on performance to activate the theory revision phase on a previously generated version of the theory, but learning can also start from scratch; it is based on the *Object Identity assumption* (different names in a description must refer to different objects). It exploits a previous version of the theory (if any), a graph describing the dependence relationships among concepts, and an historical memory of all the past examples that led to the current theory.

The learning cycle performed by INTHELEX can be described as follows. A set of examples of the concepts to be learned, possibly selected by an expert, is provided. Some initial examples, previously classified by the expert, are exploited to obtain a theory that is able to explain them. Such an initial theory can also be provided by the expert, or even be empty. Subsequently, the validity of the theory against new available examples is checked. In the case of incorrectness on an example, the cause of the wrong decision can be located and the proper kind of correction chosen, firing the theory revision process. In this way, examples are exploited incrementally to modify incorrect hypotheses according to a data-driven strategy. Test examples are exploited just to check the predictive capabilities of the theory, intended as the behavior of the theory on new observations, without causing a refinement of the theory in the case of incorrectness on them.

Another peculiarity of INTHELEX is the integration of multistrategy operators that may help solve the theory revision problem by pre-processing the incoming information [5]. The purpose of induction is to infer regularities and laws (from a certain number of significant observations) that may be valid for the whole population. INTHELEX incorporates two inductive refinement operators, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative examples. Deduction is exploited to fill observations with information that is not explicitly stated, but is implicit in their description, and hence refers to the possibility of better representing the examples and, consequently, the inferred theories. Indeed, since the system is able to handle a hierarchy of concepts, some combinations of predicates might identify higher level concepts that are worth adding to the descriptions in order to raise their semantic level. For this reason, INTHELEX exploits deduction to recognize such concepts and explicitly add them to the example description. The system can be provided with a Background Knowledge, supposed to be correct and hence not modifiable, containing complete or

³ INTHELEX is currently available in binary format for i586 DOS-based platforms (<http://lacam.di.uniba.it:8000/systems/inthelex/>).

partial definitions in the same format as the theory rules. Abduction aims at completing possibly partial information in the examples, adding more details. Its role in INTHELEX is helping to manage situations where not only the set of all observations is partially known, but each observation could also be incomplete. Indeed, it can be exploited both during theory generation and during theory checking to hypothesize facts that are not explicitly present in the observations. Lastly, abstraction removes superfluous details from the description of both the examples and the theory. The exploitation of abstraction in INTHELEX concerns the shift from the language in which the theory is described to a higher level one. An abstraction theory contains information on the operators according to which the shift is to be performed. INTHELEX, 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 with a compound object, to decrease the granularity of a set of values, to ignore whole objects or just part of their features, and to neglect the number of occurrences of a certain kind of object.

3.2 Experimental Results on the COLLATE Dataset

INTHELEX was considered a suitable learning component for the COLLATE architecture based on its previous successful application to different kinds of documents, indicating a good generality of the approach. Moreover, many of its features met the requirements imposed by the complexity of the documents to be handled. In addition to being a symbolic (first-order logic) incremental system, its multistrategy capabilities seemed very useful [8]. For instance, abduction could make the system more flexible in the absence of particular layout components due to the typist's style, while abstraction could help in focusing on layout patterns that are meaningful to the identification of the interesting details, neglecting less interesting ones. Experimental results, reported in the following, confirm the above expectations. The COLLATE dataset for INTHELEX consisted of 29 documents for the class Registration Card (FAA), 36 ones for the class Censorship Decision (DIF), 37 for the class Application form (NFA). Other 17 reject documents were obtained from newspaper articles. The complexity of the domain is confirmed by the description length of the documents, that ranges between 40 and 379 literals (144 on average) for class Registration Card, between 54 and 263 (215 on average) for class Censorship Decision; between 105 and 585 (269 on average) for class Application Form.

The examples needed to run INTHELEX describe the layout blocks that make up a paper document in terms of their size (height and width, in pixels), position (horizontal and vertical, in pixels from the top-left corner), type (text, line, picture, mixed) and relative position (horizontal/vertical alignment between blocks).

Each document was considered as a positive example for the class it belongs, and as a negative example for the other classes to be learned; reject documents were considered as negative examples for all classes. Definitions for each class were learned, starting from the empty theory, and their predictive accuracy was tested according to a 10-fold cross validation methodology, ensuring that each fold contained the same proportion of positive and negative examples. Table 1 reports the experimental results, averaged on the 10 folds, of the classification and interpretation process in this environment as regards Accuracy on the test set (expressed in percentage) and Runtime (in seconds).

As expected, the classification task turned out to be easier than the interpretation problem concerning the semantics of the layout blocks inside documents, but it should be considered that the high predictive accuracy should ensure that few theory revisions can be expected when processing further documents. Indeed, INTHELEX was able to learn significant definitions for the annotation of the given documents, as evident from the very high predictive accuracy in both cases, reaching even 99.17% for the classification task (in which never falls below 95.74%) and 98.95% in the interpretation problem (in which only 2 cases out of 28 fall below 90%).

Table 1. INTHELEX performance on COLLATE dataset

DIF	Accuracy	Runtime	FAA	Accuracy	Runtime
<i>Classification</i>	99.17	17.13	<i>Classification</i>	94.17	334.05
cens_signature	98.32	1459.88	registration_au	91.43	3739.36
cert_signature	98.31	176.59	date_place	86.69	7239.62
object_title	94.66	3960.83	department	98.95	118.62
cens_authority	97.64	2519.45	applicant	97.89	93.99
chairman	93.10	9332.845	reg_number	91.95	4578.20
assessors	94.48	12170.93	film_genre	93.02	2344.89
session_data	97.68	1037.96	film_length	90.87	3855.39
representative	92.98	13761.96	film_producer	94.05	4717.17
			film_title	89.85	4863.08

NFA	Accuracy	Runtime	NFA	Accuracy	Runtime
<i>Classification</i>	95.74	89.88	applicant	93.66	15588.15
dispatch_office	94.28	13149.31	film_genre	98.53	684.35
applic_notes	98.81	231.05	registration_au	94.64	5159.74
No_censor_card	95.47	8136.79	cens_process	98.51	4027.90
film_producer	93.98	5303.78	cens_card	94.62	3363.86
No_prec_doc	93.97	5561.14	delivery_date	95.52	3827.34

It is worth noting that the rules learned by the system have a high degree of understandability for human experts, which was one of the COLLATE requirements. For instance, Figure 3 shows a definition for the classification of documents belonging to Censorship Decision of DIF class to be read as “*a document belongs to this class if it has long length and short width, it contains three components in the upper-left part, all of type text and having very short height, two of which are medium large and one of these two is on top of the third*”. An interesting remark is that, starting with descriptions whose average length was 215, the average number of literals in the learned rules is just 22.

Figure 4 shows the definition learned by INTHELEX to label the component *film_title* in Registration Cards from FAA and the mapping of the components of the learned description in a sample document. Again, such a rule contains 23 literals only.

```

class_dif_cen_decision(A) :-
  image_lenght_long(A), image_width_short(A),
  part_of(A, B), type_of_text(B),
  width_medium_large(B), height_very_very_small(B),
  pos_left(B), pos_upper(B),
  part_of(A, C), type_of_text(C),
  height_very_very_small(C),
  pos_left(C), pos_upper(C),
  on_top(C, D), type_of_text(D),
  width_medium_large(D), height_very_very_small(D),
  pos_left(D), pos_upper(D).

```

Fig. 3. Learned Rule for Classification

```

logic_type_film_title(A) :-
  page_first(B), part_of(B,A), part_of(B,C), part_of(B,D),
  type_of_text(A), pos_upper(A),
  type_of_text(D), pos_upper(D),
  height_very_small(C), type_of_text(C),
  pos_center(C), pos_upper(C),
  width_very_large(E), height_smallest(E),
  type_of_hor_line(E), pos_center(E), pos_upper(E),
  height_very_very_small(F), pos_left(F), pos_upper(F),
  on_top(E,D), on_top(E,F), alignment_left_col(F,A).

```

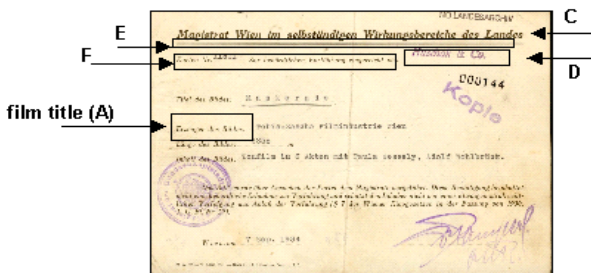


Fig. 4. Learned rules for the semantic component *film title* and the mapping on document

4 RDF Sub Architecture

Semantic Web stands for the next evolution of World Wide Web [1]. It will provide more intelligent services such as clever search engines, user tailored applications and data exchange based on semantics

World Wide Web Consortium (W3C) has been working through the most recent years in the development of technologies that could support the evolution towards the Semantic Web. While some of these technologies are still in early phases, others can already find an exploitation in real world applications. This is the case of Resource Description Framework (RDF). It represents the basic support to write metadata on

Web resources and to grant interoperability among heterogeneous applications when exchanging these metadata.

RDF describes resources in terms of primitives (classes, properties, resources, etc.) without taking into account the description structure itself. In fact, the description can be encoded in XML⁴. This ensures its portability across the Web.

Moreover, RDF represents a suitable solution to implement the Semantic Web vision also because it presents three key features:

- **Extensibility:** Each user can add its own description extending pre-existing ones without any limit.
- **Interoperability:** RDF descriptions can rely on XML serialization every time they need to be exchanged among heterogeneous platforms
- **Scalability:** RDF descriptions can be viewed as sets of three fields records (triples) (Subject, Predicate and Object). This makes them easy to fetch and manage even if a single description holds many triples in it.

4.1 RDF Management in COLLATE

COLLATE system could be easily assimilated to the wider scenario foreseen by Semantic Web: a huge quantity of resources (documents, assets) with many relationships among them. COLLATE requirements are:

- A uniform way of identifying resources (films, film related documents, cataloguing and indexing information, scientists' annotations, scientific discourses)
- Distribution of information: in fact archives still keep their resources in a decentralized architecture, in order to avoid the moving of huge data amount, both physically and electronically (for obvious reasons)
- Intelligent navigation through data and metadata: including navigation across scientific discourses on resources

For all these reasons RDF (together with XML – see the section on the COLLATE Environment) is a straightforward solution since it holds in itself the features we underlined earlier.

However, RDF and its most famous application programming interfaces (APIs) lack both in standard specifications (e.g.: there has not been specified a uniform query language yet) and in performances.

That is why the COLLATE consortium developed a component that aims to fill these gaps and embedded it into the aforementioned XML Content Manager.

We go on examining which added value our framework provides to COLLATE. It is quite obvious that a huge collection of documents and metadata such as COLLATE heritage needs a careful devising of a scalable component in order to manage storage and retrieval of both resources and relationships among them. While the solution for the former problem is delegated to efficient RDBMS, as far as the latter we developed a suitable RDF Persistence for granting scalability. This module relies on Jena Toolkit storage model for RDF. It consists on exploiting a relational representation of the RDF Triples (subject, predicate, object) stored in a database. This approach takes advantage of the outstanding performance rates of the most famous RDBMS (such as Oracle, MySQL and PostgreSQL). One of the most immediate benefits is the fact that

⁴ But also in other different formats e.g.: <http://mail.ilrt.bris.ac.uk/~cmdjb/2001/06/ntriples/>.

applications need not to load in-memory RDF *Models* (Descriptions) in order to deal with small portions of them (typically small sets of *Statements*), saving lots of memory and time for each operation.

Moreover, Jena Toolkit offers RDF Description Query Language (RDQL⁵) as language for querying RDF Descriptions. This support has been extended for querying multiple *Models*, that together with multi-user environment and scalability, proved to be a suitable solution for COLLATE requirement.

The query language, however, remains a weakness point of all RDF APIs available, including Jena. At the time of writing, still no standard query language specifications are available. This hampers the interoperability between components and, therefore, between different systems; in other words, two systems using different APIs to manage RDF can exchange data, but cannot easily exchange queries on these data.

To address this issue, the subcomponent, named Enhanced Query Engine, is able to deal with different query languages. The design of this component exploits the Strategy pattern [6], enabling the use of a dynamic set of query languages. In order to add the support for a new query language, only the classes implementing the interfaces to wrap the parser of the language and the query engine are needed, allowing for easy update. This update, obviously, can be the standard query language the W3C (together with other organizations) is working on, as soon as it is available.

4.2 Experiments

In order to prove the efficiency of our framework we hereby provide a small empirical evaluation of its performances. We tested its efficiency and effectiveness as it deals with RDF Resources that are necessary for the integration with INTHELEX. More specifically, one point of integration with INTHELEX consists in storing training examples and subsequently in retrieving them by INTHELEX for theory building and tuning. As we stated earlier, there are many reasons because documents metadata had to be encoded in RDF within COLLATE and INTHELEX learns documents classifiers starting from document layout description (metadata).

In our test cases, we considered sets of examples whose size increased dramatically, as you can see in column “Triples” in table 2. We started from an example test bed that results in 334 Triples in its RDF translation, ranging till the biggest that reaches ~78000 Triples. We measured the times required for the following operations:

- Adding a whole training set
- Querying a training set

Furthermore we present also the average timings weighted on the number of triples (columns “Add time/triples” and “Query time/triples”).

As we can see in Figure 5 (where the Y scale is logarithmic for the sake of readability), the complexity of the insertion of a new description is linear with the size of the description itself. This is outlined by “Add time/triples” ratio, that is almost constant. The time elapsed to execute queries on a particular description, on the other side, does increase with the size of the description but in a non-linear way. As clearly

⁵ <http://www.hpl.hp.com/semweb/rdql-grammar.html>

shown by “Query time/triples” ratio, a query costs around 0,1 milliseconds per statement when the number of statements is limited, but costs less than 0,01 milliseconds per statement when the number of statements is over 78000. From the data in the table 2 and figure 5, it appears that the increase of size of a 230 factor (from 334 to over 78000) produces an absolute increase in time of about a 5 factor. On the basis of these data, we can assert that the size of the descriptions being queried is of very little relevance when evaluating application performances.

Table 2. Test Results

Name	Triples	Add time	Query time	Add time/triples	Query time/triples
first_rdf	334	2594	34	7,77	0,1
second_rdf	637	5458	43	8,57	0,07
third_rdf	1295	10104	85	7,8	0,07
fourth_rdf	2750	22282	42	8,1	0,02
fifth_rdf	6341	52927	167	8,35	0,03
sixth_rdf	17534	161573	235	9,21	0,01
seventh_rdf	78445	683813	182	8,72	<0,01

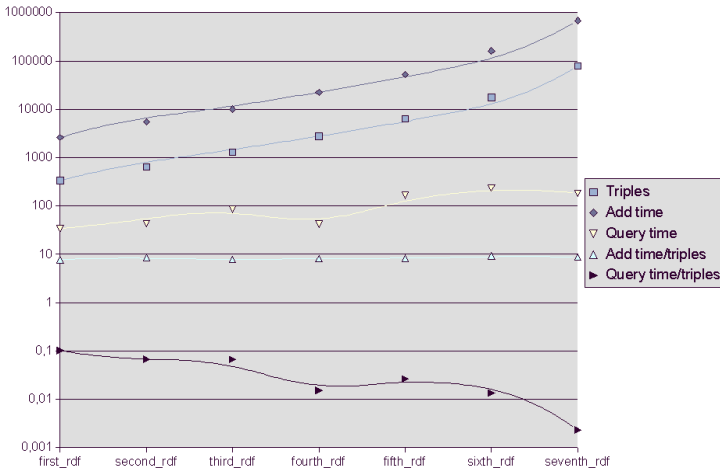


Fig. 5. Graphical representation of test results

Acknowledgments. This work has been partially funded by the European Commission under the IST Project COLLATE (IST-1999-20882) “Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material”.

References

- [1] T. Berners-Lee, J. Hendlers and O. Lassilla, The Semantic Web Scientific American, May 2001
- [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. Sølvsberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in LNCS, pages 37–48. Springer, 2001.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] E.Gamma, R.Helm, R.Johnson, J.Vlissides, *Design Patterns* Addison-Wesley Pub Co (1995)
- [7] R.T. Kouzes, J.D. Myers, and W.A. Wulf. Collaboratories: Doing science on the internet. *IEEE Computer*, 29(8):40–46, 1996.
- [8] G. Semeraro, F. Esposito, S. Ferilli, N. Fanizzi, T.M.A. Basile, and N. Di Mauro. Multistrategy learning of rules for automated classification of cultural heritage material. In E.P. Lim, C. Khoo, E. Fox, T. Costantino, S. Foo, H. Chen, and S. Urs, editors, *In Proc of the 5th International Conference on Asian Digital Libraries (ICADL)*, number 2555 in Lecture Notes in Computer Science, pages 182–193. Springer-Verlag, 2002.
- [9] 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. Sølvsberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in LNCS, pages 59–70. Springer, 2001.