A semantic annotation model for indexing and retrieving learning objects

Boutheina Smine[1,3], Rim Faiz[2], Jean-Pierre Desclés[3]
[1]LARODEC
ISG of Tunis
Le Bardo
Tunisie
Boutheina.Smine@etudiants.univ-paris4.fr

[2]LARODEC
IHEC de Carthage
2016 Carthage Présidence
Tunisie
Rim.Faiz@ihec.rnu.tn

[3]LaLIC
Paris Sorbonne University
28 Rue Serpente
Paris 75006
France
Jean-Pierre.Descles@paris4.sorbonne.fr

Journal of Digital Information Management

ABSTRACT: *The internet is an important part of our world. Offering a large and increasing amount of information, people can use it for learning, teaching, etc. Automatic tools for learning information retrieval based on semantic tags have not been effective yet. We propose here a model which aims at automatically annotating texts with semantic metadata. These metadata would allow us to index and extract learning objects from texts. This model is composed of two parts. While the first part consists of a semantic annotation of learning objects according to their categories (definition, example, exercise, etc.), the second one uses automatic semantic annotation. Generated by the first part, the latter aims at creating a semantic inverted index able to find relevant learning objects for queries. To sort the results according to their relevance, we apply the Rocchio's classification technique to the learning objects. We have implemented a system called SRIDoP, on the basis of the proposed model and we have verified its effectiveness.*

Categories and Subject Descriptors
I.2.6 [Learning]; Knowledge acquisition; **H.3.1 [Content Analysis and Indexing]:** Indexing methods; **I.2.4 [Knowledge Representation Formalisms and Methods]:** Semantic networks

General Terms: Inverted Index, Semantic Indexing, E Learning platforms

Keywords: Semantic annotation, Learning objects, Rocchio algorithm, Contextual exploration

Received: 13 March 2011, Revised 19 April 2011, Accepted 27 April 2011

1. Introduction

With plenty of information available online and in databases and increasing rapidly, several systems such as search engines and e-learning platforms play an important role within eLearning since they can support the learner in finding the necessary information for his learning, training or teaching process.

In general, users generally enter keywords into search engines, and the returned results list all web pages containing the same character string as the chosen keywords. These search engines are based on terms indexation without taking into account neither the semantics of pedagogical content nor the context.

Besides, e-learning platforms using traditional informational retrieval technology are not useful for learning object retrieval. In fact, a keyword based approach may result in retrieving information appearing in the list of results but not relating to the subject of learning.

A better alternative is to create an information retrieval model based on the semantic annotation of learning objects. In this way, the learning information presented by the author of a document is captured and the learning or the teaching process for the student or the instructor is respectively facilitated. Extracting learning objects enables a person to combine multiple objects and compose personal lessons for an individual learner.

There are several systems which offer manual annotation of learning objects. Yet, producing interesting semantic metadata manually is not interesting, because of the diversity and non relevance of the annotations introduced. Automatic procedures exist but they can only fill in "simple" and "low added value" fields (e.g. date, author, title, etc.).

Using discursive organizations of natural language texts is a further approach we support to define another kind of learning objects retrieval. This approach is not in contradiction with the two previous ones, but it is a complement system able to use morpho-syntactical extensions [27] for learning objects terms. This article explains how a new kind of learning objects retrieval system is implemented by using semantic and discourse automatic annotation of learning objects according to their types (Defintion, Example, Exercise, etc.). We note that the automatic annotation of learning objects is not a simple task because the pedagogically related information depends to a great extend on context. Add to that it can not be expressed at a generic level.

We propose in this paper a learning objects retrieval model based on semantic annotation process with Contextual Exploration and on learning objects indexation. To improve the results

obtained, we apply a machine learning technique that sorts the results according to their relevance.

The rest of this paper is organized as follows: Section 2 deals with the presentation of related works on learning information processing. In section 3, we present the semantic learning categories for text mining. Our model for learning objects retrieval is detailed in section 4. Before concluding, we illustrate the evaluation results of the different parts of our model in the fifth section.

2. Related works

Several works provide infrastructure and services for learning information annotation, indexing, and retrieval from documents. Among these works, we can mention:

The work of Puustinen and Rouet [19] classified existing information search systems according to the characteristics of the helpers- their ability to adapt their answers to the learner's need (from "*No adaptation to the learner"* to *"Excellent adaptation to the learner"*). The helper's awareness of both the student and the search context is what truly differentiates searching in a passive information system from interacting with a human helper.

Therefore, we are interested in annotation technique applied within learning information systems. We noticed, in the last decade, two search orientations in learning information retrieval. The first one is the Berners-Lee "Web Semantic" dealing with manual or semi-automatic annotations based on domain ontologies. According to Lee et al., [18], the second one is qualified by the traditional information retrieval technology as keyword-based vector space model [25] and Decision trees [26], [28].

Within the first orientation presented above, several works provide infrastructure and services for learning information annotation, indexing, and retrieval from documents. Among these works, we can mention:

QBLS [4] is a learning system for instructors and students. It proposes annotations using an RDF description. The course is structured referring to a pedagogical ontology constituted of cards (definition, example, procedure, solution, etc), then the pedagogical resources are created (course, topic, concept, and question). These resources deduced from the initial course are stored with their respective annotations in "a database of pedagogical knowledge". Students can thereafter practice how to resolve some questions, or learn more details about a definition, etc. When the user formulates a request, the search engine *Corese* is activated to search the pedagogical cards as response to the user's query.

The platform TRIAL SOLUTION [5], [22] offers e-books annotations relative to the *pedagogical role* of the resources contents (definition, theorem, explanation, etc), the *key words* and the *relations* with other resources (ex: reference). These annotations refer to a thesaurus of mathematics; both of them are managed by experts. However, we notice that the system allows a multitude of corrections of the annotations introduced by the experts. These can affect the quality of the annotations.

We also denote the SYFAX system [14] which presents several annotations indicating: (1) correspondence of the document with the user profile (Yes/Not), (2) the user point of view on the document (interesting, average, not very interesting), (3) the type of documents (TD, TP, etc) based on the ontology "Type of documents" which was created manually and (4) concepts of the domain treated by the document referring to an ontology of the informatics domain built automatically from a dictionary named FOLDOC.

In order to index pedagogical documents, the various systems mentioned stored the generated annotations in knowledge databases from which relevant results are extracted. A refinement process of the request based on two ontologies is suggested; one dealing with educational material types and the other one with the computer science domain. Thus, relevant documents have the same type and the same concept enounced by the user's request.

For all the systems presented above, the problem of indexing pedagogical documents is discussed from various sides: (1) the course is structured manually according to a pedagogical ontology or a specific architecture [20] in order to use it in an e-learning environment, (2) the course is semi-annotated by users to produce personalized course supports. In all cases, a human intervention is provided to enrich documents with metadata. Therefore, many producers of learning content are not interested in going back and annotating all their work.

Within the second orientation, we can mention the work of Hassan and Mihalcea [11] who explore the task of automatically identifying educational materials, by classifying documents with respect to their educative value. The following features are associated with each document in the dataset: Educativeness (a four point scale ranging from non educative to strongly educative), Relevance (a four point scale ranging from non relevant to very relevant), Content categories (Definition, Example, Question & Answers, etc.), Resource type (Blog, Online book, forums, Presentation, etc.), Expertise (The expertise of the annotater in each of the selected topics on a four point scale). Authors experiment with automatic classifiers (Naïve Bayes and Salton Vector Machine) to annotate the educativeness of a given document.

We also denote the SOAF system [23] which proposes architecture to extract semantic descriptions of multimedia learning resources automatically. It is based on Latent Semantic Indexing using the representation of the resources in a vector space through their visual features. SOAF considers three types of metadata that might describe a learning object : (1) low-level features which generate automatic semantic indexing (2) High level descriptors provided by authors (title, date of creation, etc.), (3) collaborative annotations that are given by end users.

The authors in [17] target the problem of finding educational resources on the web. They suggest providing metadata for educational web pages, considering first text classification, and then information extraction.

The focus of their work was limited to metadata extraction relative to the whole document. A set of properties (Relevance, Content Categories, course title, instructor, year, etc.) was explored to annotate and classify the educational resources. Therefore, their methods don't enable to reach the contents of the documents in order to analyse their textual segments.

To sum up, we can say that, in the context of learning information retrieval, there exist systems in favour of manual or semi-automatic annotation. In this latter, a human intervention is almost necessary, to annotate the documents. Of course, this is not an interesting task for users. When the annotation is automatic, only metadata relative to the whole documents are extracted. We support the task of automatically annotating objects with semantic metadata relative to their learning categories.

In this paper, we present a model which aims at automatically annotating learning objects according to their semantic categories (Definition, Example, Exercise, etc.) in order to index and extract learning objects as response to the user's query. Then,

a machine learning technique is applied to the extracted objects to sort them with regard to their relevance.

The semantic learning categories are an important part of our work. So, we detail it in the next section studying the different information learning categories.

3. Semantic learning information categories for text mining

When working with a computer, learners will manipulate digital artifacts to perform the learning activity they have been assigned to. With the spread of pedagogical resources on the web, the idea has emerged to capitalize these artifacts by learning objects [24].

Wiley [21] defines a learning object as any digital entity which we can use, re-use or refer to during a learning process. Learning objects are supposed to be small parts of courses that may be assembled together. In reality, a complete course is "sliced" to create several learning objects that can be composed together later on.

For the purposes of this paper, we use a rather functional definition of a learning object as a textual segment (sentence, paragraph, and document) used in a learning process. This learning object is assigned to one of the types presented in Figure 2.

The first interest of learning objects is the creation of opportunities for institutions and instructors in their lesson planning and execution. Learning objects are considered as cost and time efficient by emphasizing annotation, retrieval and reuse over individual creation. We propose to categorize learning objects according to 6 categories (Plan, Exercise, Example, Course, Characteristic and Definition). Figure 1 represents learning object categories and relations binding these objects.

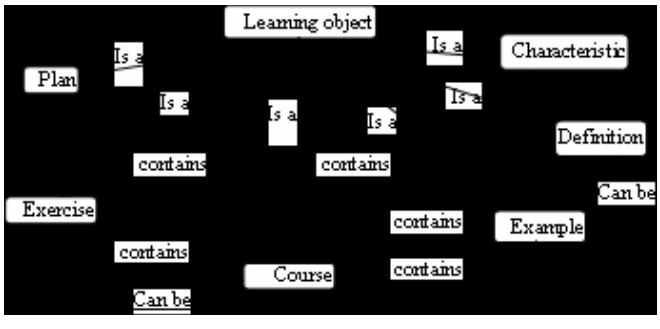

Figure 1. Relations binding learning objects categories

In our research, we are guided by the assumption that is:

"A user who searches relevant learning information proceeds by guided readings giving preferential processing to certain textual segments (sentences or paragraphs)". The aim of this assumption is to reproduce: "What does a human reader do naturally, in particular a learner, who underlines certain segments relating to a particular learning object attracting his attention.

Indeed, such a learner could be interested in a definition by formulating a request such as: find documents which contain "The definition of the SQL Language". Another user will look for examples that can be applied on a certain concept (for instance "social class" in sociology, "inflation" in economy, "polysemic" in linguistics, ..) by exploring many texts (specialized encyclopaedias, handbooks, articles). So, these examples will be integrated to the user's pedagogical resources. While some users may be interested in applying exercises to a concept, others require a course support for learning or teaching.

The aim of the study dealing with learning object categories is a possible annotation of the textual learning objects. These annotations which correspond to a guided research enable to extract learning objects from texts.

Each learning category, as we mentioned above, is explicitly indicated by identifiable linguistic markers in the texts.

Our hypothesis is that semantic learning objects leave some discursive traces in textual document. The learning object categories are described as follows:

- On the one hand, a complex relation between different object categories (cf. Figure 1) and on the other hand a set of classes and subclasses of linguistic units (indicators and indices) structured inside a "semantic map" of learning categories (cf. Figure 2).
- A set of rules : each rule connects a class of indicators with different clues.

The semantic map is like an organization of learning object categories, whose classes of indices are extensional counterparts. The semantic map can also be conceived as an ontology of learning object categories independently of different domains of application. Indeed, the expressions of the semantic map for a learning object category are the same in different domains like Informatics, mathematics, management, …since these expressions are used by the author to express learning information. In some types of texts (narrative texts, news articles,) some learning information will not be present but in others (course support, assignments, tutorials,..) these expressions organize the text and give information about the intention of the author. The noise caused by the polysemy of an indicator is filtered by the rules. The latter researching complementary indices in its context to eliminate false interpretations.

The first level of the semantic map makes it possible to release 6 learning object categories: (i) Course, (ii) Plan, (iii) Exercise, (iv) Example, (v) Definition, (vi) Characteristic. For instance, the Definition learning category rules are triggered by occurrence of Definition indicators and the semantic annotation is assigned if linguistic clues, like prepositions, are found in the indicator's context. For example, a definition can be detected in textual segments as :

- An *explanation* as the linguistic structure "……convey to……..",
- *Significance* in the linguistic structure "…….. means……."
- A *condition formulation* as in the linguistic structure" …is a……if…..".

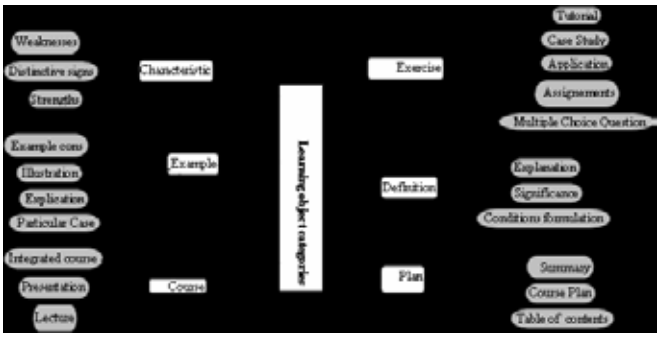

Figure 2. A learning objects semantic map

Our choice of the subtypes is explained by the diversities of the object categories in the learning field. However, we have constructed our semantic map (indicators and clues) on the basis of linguistic structures frequently found in pedagogical documents. This semantic map represents a part of linguistic resources that guide our annotation and indexation work, the latter being detailed in the next section.

4. Our learning information retrieval model

Our model is composed of two parts: The first part concentrates on automatic annotation of pedagogical texts according to learning object categories [15], [16]. The second one uses automatic semantic annotation which is generated by the first part. This part aims at creating a semantic inverted index able to find relevant objects for queries associated with learning categories such as *Definition, Exercise, Example*, etc. Then, we propose to sort these objects according to their relevance using the Rocchio classification algorithm.

4.1 Learning Objects Annotation

4.1.1 Segmentation

Before applying the annotation task, the content of the considered document has to undergo a segmentation action which lies in determining the unit's borders (unit as sentence, paragraph, etc.). We have implemented our own segmentor based on the segmentation rules developed by Mourad [12]. The latter defined a textual segment starting from a systematic study of the punctuation marks. Our plain text documents are then transformed into XML structured documents (titles, sentences, paragraphs, etc.).

4.1.2 Learning Objects Annotation Process

For the annotation process, we make use of the *Contextual Exploration technique* 'EC' [7], [8] which call upon knowledge exclusively linguistic and present in the texts. This linguistic knowledge is structured in form of lists and is capitalized in a knowledge base. There are two kinds of lists: indicator lists on the one hand, contextual index (clue) lists on the other hand. Indicators are specific to a given information learning category (i.e.: to recognize a *Definition*, to locate an *Example*, etc.). Each indicator is seen as associating a set of heuristic rules of Contextual Exploration. The application of a rule called by an indicator, require an explicitly research, in the indicator context, of the linguistic indexes complementary to the indicator to solve the task. In addition, it doesn't need a morpho-syntactic analysis which reduces considerably the execution time of the method [6], [9], [10].

We focus on the learning object categories (see the semantic map) to construct our contextual exploration rules. We go through each document in order to extract linguistic structures that define the learning object categories, i.e. the category "Definition" can be expressed by several structures : "...is defined as…", "The definition of ….is….", "To define …, we say that…". These linguistic structures are expressed by discursive markers (indicators and clues) which are represented in a list of verbs, prepositions, nouns, etc. Relations binding indicators and clues are defined within Contextual Exploration rules. The rule is triggered when one of its indicators is detected within the textual segments. These rules must identify an indicator (Ii) then locate linguistic clues to the left (CLi) and/or to the right (CRi) context of the indicator, which involves the confirmation or not of the semantic value carried by the indicator.

For each category of the semantic map, we defined the set of rules which covers all the possible linguistic forms of learning object. We have developed about 180 rules. We start from a textual example to generalize all linguistic structures. This method permits to define incrementally a solid base of rules. Indeed, we give the permission to the user to manage the EC rule base (adding, updating, deleting rules) through the Access Database system. The Table 1 shows some examples of rules. In this table, IdR denotes the identifier of the rule; CL_1, CL2 denote the left clues and CR_1, CR_2 denote the right clues.

IdR	CL_1	CL_2	Indicator	CR_1	CR_2
RD1	is|are		defined	as	
RD2			is | are	a|an|the	
RC1	The| A		Characteristic| Characteristics	of	is| are
RE1	This is	an| the	example| examples	of	

Table 1. Examples of Contextual Exploration Rules

For example, the EC rule RD1 (see Table 1) would follow these steps to annotate a textual segment as a *Definition*:

- Express the semantic of the "*Definition*" category by means of a relevant indicator, represented in this case by the verb "defined".
- To confirm the indicator's "definition semantic", we need first to identify in the sentence terms of the list CL_1 (the verb "is" or "are") in the left context
- Indicator needs another expression like the preposition "as" in the right context to allow the annotation of the sentence as a definition.

The whole rules relative to the various categories and their respective indicators and clues constitute the linguistic resources that we employed to annotate learning objects.

The annotation process is also expressed by the following figure which can explain the algorithm presented above.

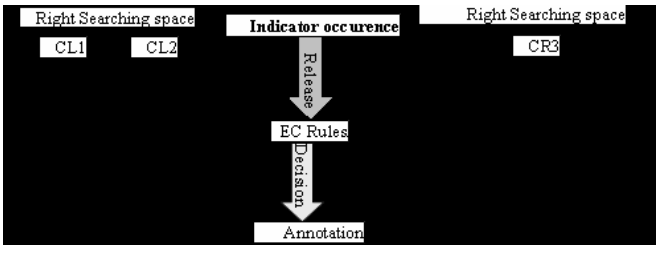

Figure 3. The annotation process

We take an extract from a pedagogical document:

SQL stands for "Structured Query Language". In fact, SQL is a complete language of relational database management. It was designed by IBM in the 70s. It became the standard language of the relational database management systems (RDBMS). SQL is the language used by the major RDBMS: DB2, Oracle, Ingres, RDB, ... However, each of these RDBMS has its own variant of the language. SQL is defined by the parameters of the RDBMS used. This course support presents the commands core available on all of these RDBMS and their implementation in Oracle Version 7.

When a rule of the learning category *Definition* is applied to the example above, it permits to annotate, as a definition, the sentences *"SQL stands for "Structured Query Language""* and

"SQL is a complete language of Relational Database Management". The Definition learning category and its subcategories *Significance* and *Explanation* are detected, respectively, through the expressions "stands for" in the first sentence and *"is"* in the second one which is an occurrence Ii belonging to the definition indicator and the right clue CR1 *"a"*.

To annotate the example *"It was designed by IBM in the 70s "* with the learning category *Characteristic*, we firstly detected the expression *"was designed"* then we searched, to the right context of the indicator, the clue CR1 "*by"*. If the two elements (Ii and CR1) are present then we can annotate the segment as a *Characteristic*.

We noticed that the sentence *"SQL is defined by the parameters of the RDBMS used"* illustrates the case of negative clues. In fact, the presence of the negative clue *"by"* prevents the annotation of the segment as a *Definition* although the presence of the indicator *"is*" and the clue *"defined"*.

For a rule of the learning category *Course*, it is enough to find an occurrence Ii on the title level to annotate the document as a *Course*. The nominal indicator of this rule is the word "*Course*" and other words like "*Chapter", "Course Notes"*, etc.

Beyond the title, the existence of a *Course* indicator does not imply an annotation of the document as a *Course*. The clues *"This"* and *"presents"* are necessary, as the case of the sentence *"This course support presents the commands core available on all of these RDBMS and their implementation in Oracle Version 7"* to annotate the document as a *Course Support*.

With regard to the learning category "Exercise", the indicator can be verbal (*a*) or nominal (*b*), for example:

(*a*) "*Formulate an SQL clause*" The indicator is the verb "*Formulate*"

(*b*) "*Exercises on SQL Requests*" has as indicator the noun "*Exercises*"

We have introduced another parameter to the rule which is the emplacement of the term expressed by the user's query. This is due to the fact that the place of the term expressed in the query varies according to the rule applied to annotate the learning objects, i.e. for the category Definition, the term "*SQL Language*" can exist in the beginning of the sentence "*SQL Language is defined as the .....*", or in the middle of the sentence "*X has defined the SQL Language as ……*". We have designed this emplacement with a set of values, relatively to the indicator, and the clues of the rule (left or right of the indicator or the clues), i.e. ***LIND*** indicates that the emplacement is to the left of the indicator and ***RCL1*** indicates that the emplacement is to the right of the left clue.

4.2 Indexing Annotated Objects

The aim of this step is to build up a multiple index composed of learning objects (sentences, paragraphs …), semantic annotations (Definition, Example, Plan …). Each learning object is associated with several important pieces of information such as:

- a semantic annotation (Definition, Exercise, Plan, …) according to the semantic categories used in the annotation process
- document URI (Uniform Resource Identifier) for the identification of the document path
- the full-text content of the learning object for a relevant answer to users
- the emplacement of the term enounced in the user's query

We have implemented a learning information retrieval system, called SRIDoP (**S**ystème de **R**echerche d'**I**nformations à partir de **Do**cuments **P**édagogiques), on the basis of the proposed model. SRIDoP uses a query language which is based at the same time on both linguistic terms (constitutive of textual segments) and semantic learning categories (definition, example, exercise …). Let us see some queries for the "Exercise" category. The answer to the query "Exercises on SQL Language?", in French "Exercices sur le langage SQL?" gives a set of learning objects (textual segments) grouped through a document URI (the annotated document by the annotation process). Each learning object presents a semantic annotation ("Exercise" annotation for this example).

The search engine proceeds as follows:

- The query, in French, has two important functions: a learning object category ("Exercise") and the term "SQL Language".
- SRIDoP extracts all learning objects fund in the index associated with the annotation "Exercise"
- For each object extracted, SRIDoP searches the term "SQL Language" and its synonyms in the emplacement enounced in the index. For the term synonyms, we used a component of the synonyms dictionary WOLF (a French version of WordNet) to replace the query term by its synonyms. For example, if the term emplacement is *RIND*, the system looks for the term "SQL Language" in the right of the indicator.
- Selection from these learning objects, all objects within an occurrence of the term "SQL Language" or its synonyms in the well emplacement.
- Display all present information in the index related to each learning object selected.

4.3 Sorting the learning objects

We propose to sort the objects displayed by our system according to their relevance. So, we implemented a version of Rocchio algorithm [3], as adapted to text categorization by [1]. Our choice is justified by the fact that a learning object can be classified to more than one class. i.e. An object concerning the SQL Language can also concern the Data Base System and so on. We note that the Vector Salton Machine technique can satisfy this assumption by applying the Rocchio's algorithm.

First, the user has to correspond the terms of his request to a topic from a set of fifteen topics of different fields. The topic chosen represent the class C_{user} against which the objects will be sorted. We note that we consider a learning object as a textual segment having different sizes (sentence, paragraph, document, and so on).

The application of the Rocchio classifier can be divided into three steps: pre-processing, learning and sorting. The pre-processing includes objects formatting and terms extraction. We use single and compound words as terms.

The learning objects are extracted from the learning corpus collected within the annotation and indexation steps. In the learning step, we presented these objects as vectors of numeric weights. The weight vector for the *m*th object is $V^m = (p_1^m, p_2^m, \ldots\ldots, p_l^m)$, where *l* is the number of indexing terms used. We adopted the TF-IDF weighting [13] and define the weight p_k^m to be :

$$p_k^m = \frac{f_k^m \log(N/n_k)}{\sum_{j=1}^{l} f_j^m \log(N/n_j)}$$

Here, N is the number of objects, n_k is the number of objects in which the term index k appears, and f_k^m is:

$$f_k^m = \begin{cases} 0 & q = 0 \\ \log(q) + 1 & Sinon \end{cases}$$

Where q is the number of occurrences of the indexing term K in object m. We produced a prototype for each class C. This prototype is represented as a single vector $\vec{c}_j$ of the same dimension as the original weight vectors $v^1, \ldots, v^N$. For class C, the K'th term in its prototype is defined to be :

$$\vec{c}_j = \frac{\alpha}{|C_j|} \sum_{m \in C_j} p_k^m - \frac{\beta}{|N - C_j|} \sum_{m \notin C_j} p_k^m$$

Where C_j is the set of all objects in class C. The parameters α and β control the relative contribution of the positive and negative examples to the prototypes vector, we use the standard values $\alpha = 4$ et $\beta = 16$ [2].

When the learning step is achieved, we launched the sorting step and we measured the similarity between the objects given as response to the user's query and the class chosen by the user C_{user}. Each object is first converted into weight vector $\vec{O}$ using TF-IDF weighting, and then compared against the class prototype $\vec{c}_{user}$ using the cosines measure:

$$\cos(\vec{C}_{user}, \vec{O}) = \frac{\vec{C}_{user} * \vec{O}}{|\vec{C}_{user}||\vec{O}|}$$

Objects having a similarity cosine measure lower then a threshold θ are selected and then sorted ascending against their similarity measure with the prototype $\vec{c}_{user}$. The θ value varies according to the learning objects category. i.e. the object content annotated as a *Course* contains more significant terms than an object of category *Exercise* ($\theta_{Course} < \theta_{Exercise}$). We take into account only positive values of the similarity measures.

5. Experimentation and results

We have implemented the SRIDoP system using the language *Java* and the Platform *Lucene* to annotate, index and sort the learning objects. To constitute the learning corpus for all the steps, we collect a data set covering the fifteen topics used in the step of "Creation of learning card-index"(i.e.Local Networks, Job-shop Scheduling, Programming language, Database, Maintenance, and so on.). Starting with each of these topics, a query is constructed and run against the Google search engine, and the top 20 ranked search results are collected. Note that the meaning of some terms can be ambiguous, e.g.., "Base" or "Record" and thus we explicitly disambiguate the query by adding the word "data". By performing this explicit disambiguation, we can focus on the learning property of the documents returned by the search, rather than on the differences that could arise from ambiguities of meaning.

The set of documents collected is constituted by 60 supports of course, 65 Assignments, 85 PowerPoint Presentations, 30 Syllabus and pages of different natures (scientific articles, web sites pages, etc.). The average length of these documents is about 23 pages.

Our testing corpus is composed of 1000 documents in French, mainly of learning nature: Support of Courses, Assignments, PowerPoint presentations, Syllabus, and documents of different nature. These documents are files in different formats (DOC, PDF, PPT, HTML, TXT, etc.) and have an average length of 53.6 pages.

5.1 First step: Learning objects annotation

To evaluate this step, our testing corpus was annotated by two experts: for each learning object spotted, they affect to it a category. The results of the SRIDoP annotation process are illustrated in the table below where **NOA**: Total number of annotated objects, **NOAC**: Number of objects annotated correctly, **NOMAC**: Number of objects annotated by the experts:

Learning Object Category	NOA	NOAC	NOMAC	Precision (%)	Recall (%)	F-score (%)
Plan	88	85	98	96,59	86,73	91,40
Course	72	60	85	83,33	70,59	76,43
Definition	228	140	266	61,40	52,63	56,68
Characteristic	139	124	156	89,21	79,49	84,07
Example	357	349	376	97,76	92,82	95,23
Exercise	760	705	776	92,76	90,85	91,80

Table 2. Experimentation results of the Annotation step

$$Precision = \frac{NOAC}{NOA}$$

$$Recall = \frac{NOAC}{NOMAC}$$

$$F\text{ - }score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

According to the experimentations presented above, the annotation results are promising. Indeed, the precision of the annotation exceeds the 85% for most learning categories (Example, Exercise, Plan, etc). But, concerning the "Definition" category, the corresponding precision is average. This derives owing to the fact that certain rules can annotate at the same time objects reflecting or not a "definition". Such the case of a "Definition" category rule which has as an indicator the occurrence "*is is*|*are* " and as clue "*a*|*an*|*the*". These indicators and clues may exist within a textual segment of a defining nature or not. During the experimental phase, we could also note that the effectiveness of the annotation is closely related to the document segmentation effectiveness.

5.2 Second step: Indexing annotated objects

To test this module, we formulated 25 queries for each learning category. These queries deal with the fifteen topics of the learning and testing corpus. For each learning category, we illustrated the number of the returned results and the number of the relevant results given the whole set of the entered queries. The results are presented in the table below (Table 3), where **NR**: Total number of results, **NRP**: Number of relevant results, **NRRU**: Number of relevant objects existing in the index.

$$Precision = \frac{NRP}{NR}$$

$$Recall = \frac{NRP}{NRRU}$$

$$F\text{ - }score = 2 * \frac{Precision * Recall}{Précision + Recall}$$

Learning Object Category of the Query	NR	NRP	NRRU	Precision (%)	Recall (%)	F-score (%)
Plan	72	66	77	91,67	85,71	88,59
Course	43	35	54	81,40	64,81	72,16
Definition	156	112	193	71,79	58,03	64,18
Characteristic	94	86	112	91,49	76,79	83,50
Example	213	198	230	92,96	86,09	89,39
Exercise	517	465	520	89,94	89,42	89,68

Table 3. Experimentation results of the Learning Objects indexing

At the end of these experiments, we conclude that the results of the document-query matching depend on the annotation results (see Fig. 4). The F-score value for the searching process progresses with the annotation process one. This is due to the fact that the learning object indexation step is executed from the annotated learning objects. The searching process quality improves with the annotation process one. This latter depends on the segmentation process quality as we have mentioned in the above.

5.3 Third step: Sorting Learning objects

Following the extraction of learning objects, we sorted these objects according to their similarity with the class C_{user}. With reference to many experiments, we have fixed the threshold value θ at : (i) 0.1 for the *Course* and *Definition* categories, (ii) 0.3 for the *Plan* and the *Example* categories, (iii) 0.45 for the *Characteristic* and *Exercise* categories.

On one side, decreasing the θ value reduces the set of relevant objects, on the other side, increasing it leads to the selection of irrelevant objects.

We assigned each object into one of the following categories: **A** (objects sorted as relevant), **B** (objects sorted correctly as relevant), **C** (relevant objects). The precision and Recall and F-score for each learning category are calculated as:

$$Precision = \frac{B}{A}$$

$$Recall = \frac{B}{C}$$

$$F\text{ - }score = 2 * \frac{Precision * Recall}{Précision + Recall}$$

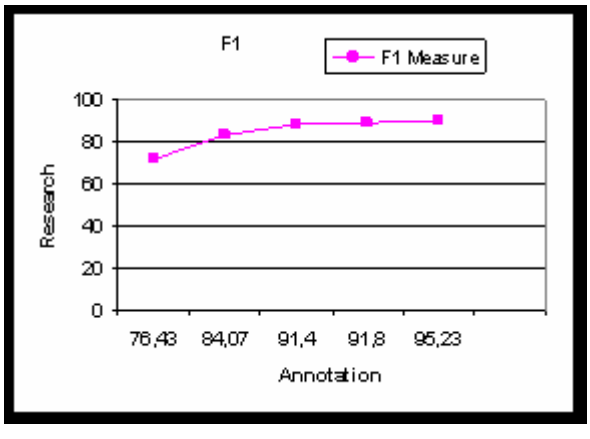

Figure 4. Retrieval results evolution with those of annotation

We obtained an average of Precision=86%, of Recall=75%, and of F-score function= 80,12% for all the studied learning categories. Through our experiments, we conclude that the sorting step results depend not strictly on the annotation and indexation ones. There are other parameters which influence the classification results as the training corpus, the choice of the indexing terms, etc.

Conclusion and Future Works

In this article, we proposed a model for learning objects retrieval from documents. To develop it, we proceed by a semantic annotation of learning objects, then an indexation of these objects to find relevant learning objects for queries associated with semantic categories. Through the evaluation results, we observe the originality of a learning object indexation based on a semantic annotation relatively to a key-words searching system. This work comes within the context of learning objects processing and retrieval. Actually, it constitutes a considerable target in many application domains as the e-learning domain, training courses domain, data management systems, etc. One of the future works that we propose is to extend the semantic map of the pedagogical objects categories by other categories as Method, Author, etc. We also look forward to fuse the annotation and classification results using a score function to perform the accuracy SRIDoP system.

References

[1] Ittner, D.J., Lewis, D.D., Ahn, D. D (1995). Text categorization of low quality images, *In*: Proc. of the SDAIR-95, Las Vegas, p. 301-315.

[2] Buckley, C., Salton, G., Allan, J (1994). The effect of adding relevance information in a relevance feedback environment, *In:* Proc. of the International ACM SIGIR Conference, p. 292-300.

[3] Rocchio, J (1971). Relevance feedback information retrieval. *In:* Gerard Salton editor, The SMART Retrievall System experiments in automatic document processing, Prentice-Hall, Englewood Cliffs, NJ, p. 13-323.

[4] Dehors, S., Faron-Zucker, C (2006). QBLS: A Semantic Web Based Learning System, *In*: Proc. of the World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA, Orlando.

[5] Dehors, S., Faron-Zucker, C., Kuntz, R (2006). Reusing Learning Resources based on Semantic Web Technologies, *In*: Proc. of the International Conference on Advanced Learning Technologies, Kerkrade.

[6] Elkhlifi, A., Faiz, R (2010). French-Written Event Extraction Based on Contextual Exploration, *In*: Proc. of The Florida Artificial Intelligence Research Society (FLAIRS), AAAI Press, Florida.

[7] Desclés, J.P (1997). Systèmes d'exploration contextuelle. In C. Guimier (ed.) Cotexte et calcul du sens, Presses Universitaires de Caen.

[8] Desclés, J.P (2006). Contextual Exploration Processing for Discourse Automatic Annotations of Texts, *In*: Proc. of The Florida Artificial Intelligence Research Society (FLAIRS), invited speaker, Florida, p.281-284, AAAI Press.

[9] Djioua, B., Garcia-Flores, J., Blais, A., Desclés, J.P., Guibert, G., Jackiewe, A., Le Priol, F., Nait-Baha, L., Sauzay, B (2006). EXCOM : an automatic annotation engine for semantic information. *In*: Proc. of The Florida Artificial Intelligence Research Figure 4. Retrieval results evolution with those of annotation Society (FLAIRS), Florida, pages 285-290, AAAI Press.

[10] Elkhlifi, A., Faiz, R (2009). Automatic Annotation Approach of Events in News Articles. *International Journal of Computing & Information Sciences*, p. 19-28.

[11] Hassan, S., Mihalcea, R (2009). Learning to identify educational materials, *In*: Proc. of the Conference on Recent Advances in Natural Language Processing (RANLP), Bulgaria.

[11] Mourad, G (2002). La segmentation de textes par Exploration Contextuelle automatique, présentation du module SegATex. Inscription patiale du Langage : structure et processus ISL sp., Toulouse.

[12] Salton, G (1991). Developments in automatic text retrieval. Science 253 (5023) p. 974-980.

[13] Smei, H., Ben Hamadou, A (2005). Un système à base de métadonnées pour la création d'un cache communautaire-Cas de la communauté pédagogique, *In*: Proc. of the International E-Business Conference, Tunisia.

[14] Smine, B., Faiz, R., Desclés, J.P (2010). Pedagogical objects annotation based on Contextual Exploration, *In*: Proc. of the International Arab Conference on Information Technology, Benghazi, Lybie.

[15] Smine, B., Faiz, R., Desclés, J.P (2010). Analyse de documents pédagogiques en vue de leur annotation, *Journal of New Information Technologies* (RNTI), E-19, Ed. Cépaduès, p. 429-434.

[16] Thompson, C., Smarr, J., Nguyen, H., Manning, C (2003). Finding educational resources on the web : Exploiting automatic extraction of metadata, *In*: Proc. of the ECML Workshop on Adaptive Text Extraction and Mining.

[17] Lee, M., Tsai, K., Wang, T. (2008). A practical ontology query expansion algorithm for semantic-aware learning objects retrieval. Computer & Education, Vol 50, 1240-1257.

[18] Puustinen, M. & Rouet, J-F. (2009). Learning with new technologies: Help seeking and information searching revisited. Computer & Education, Vol 53, 1014-1019.

[19] Shi, H. (2010). Developping E-learning materials for software development course. *International Journal of Managing Information Technology*, 2 (2).

[20] Wiley, D.A. (2000). Connecting learning objects to Instructional design theory: a definition, a metaphor, and a Taxonomy, *In:* Wiley (eds.), The Instructional Use of Learning Objects.

[21] Buffa, M., Dehors, S., Faron-Zucker, C., Sander, P. (2005). Vers une approche Web Sémantique dans la conception d'un système d'apprentissage. Revue du projet TRIAL SOLUTION. WebLearn.

[22] Cernea, D., Moral, E.,Gayo, J.E. (2008). SOAF: Semantic indexing system based on collaborative tagging. Interdisciplinary *Journal of E-learning and Learning Objects*, V 4. [23] Christiansen, J.-A., Anderson, T. (2004). Feasibility of course development based on learning objects: Research analysis of three case studies, *International Journal of Instructional Technology and Distance Learning*, 1(3).

[24] Salton, G. (1988). Automatic Text Indexing Using Complex Identifiers. ACM Conference on Document Processing Systems, New Mexico.

[25] Elkhlifi, A., Faiz, R (2008). Approche d'annotation automatique des événements dans les articles de presse. EGC'2008. 37-42.

[26] Faiz, R (2006). Identifying Relevant Sentences in News Articles for Event Information Extraction. *Int. J. Comput. Proc. Oriental Lang.,* 1-19.

[27] Elkhlifi, A., Faiz, R (2007). Machine Learning Approach for the Automatic Annotation of the Events, *In*: Proc. of The Florida Artificial Intelligence Research Society (FLAIRS), AAAI Press, Florida.

Authors Biographies

Boutheina Smine received the Bachelor of Science degree (B.S.) in Computer Science applied to Management in June 2003 from the High Institute of Management of Tunis. She received the Master of Science (M.S.) in Computer Science applied to Management in 2005. She is now a PhD Student in Computer Science between University of Tunis and Sorbonne University, France. Her research is focused on learning information extraction and pedagogical documents annotation.

Dr. Rim Faiz obtained his Ph.D. in Computer Science from the University of Paris-Dauphine, in France. She is currently a Professor of Computer Science at the Institute of High Business Study (IHEC) at Carthage, in Tunisia. Her research interests include Artificial Intelligence, Machine Learning, Natural Language Processing, Information Retrieval, Text Mining, and Semantic Web. She is member of scientific and organization committees of several international conferences. Dr. Faiz is also the responsible of the Professional Master "Electronic Commerce" and the Research Master "Business Intelligence applied to the Management" at IHEC of Carthage.

Jean-Pierre Desclés is a Professor in Computer Science and Linguistics at the University of Paris-Sorbonne. He teaches computational linguistics, theoretical linguistics, logic and language engineering. He also conducts research in these areas. He has published numerous articles and several books, including *Langues naturelles, langages applicatifs et cognition*, Hermès, Paris, 1990. He is Director of the Doctoral School "*Concepts et Langages*" and the LaLIC laboratory (*Languages, Logic, Informatics and Cognition*) at Paris-Sorbonne.