Relevant learning objects extraction based on semantic annotation

Boutheina Smine*
Languages, Logic, Informatics and Cognition (LaLIC),
University of Paris Sorbonne,
28 Rue Serpente, 75006 Paris, France
Email: boutheina.smine@yahoo.fr
*Corresponding author

Rim Faiz
LARODEC,
IHEC-University of Carthage,
2016 Carthage Présidence, Tunisia
Email: rim.faiz@ihec.rnu.tn

Jean-Pierre Desclés
Languages, Logic, Informatics and Cognition (LaLIC),
University of Paris Sorbonne,
28 Rue Serpente, 75006 Paris, France
Email: Jean-Pierre.Descles@paris4.sorbonne.fr

Abstract: We propose, in this paper, a model that extracts automatically learning objects as response to a user request. To do this, we proceed by automatically annotating texts with semantic metadata. These metadata will allow us to index and extract learning objects from texts. Thus, our model is composed of two principal parts: the first part consists of a semantic annotation of learning objects according to their semantic categories (definition, example, exercise, etc.). The second part uses automatic semantic annotation which is generated by the first part to create a semantic inverted index able to find relevant learning objects for queries associated with semantic categories. We add a secondary part to our model which sorts the results offered to the user according to their relevance. We have implemented a system called SRIDoP, on the basis of the proposed model and we have verified its effectiveness.

Keywords: semantic annotation; learning objects; indexing pedagogical documents; contextual exploration.


Biographical notes: Boutheina Smine received the Bachelor of Science degree (BS) in Computer Science in Management in June 2003 from the High Institute of Management of Tunis. She received the Master of Science (MS) in Computer Science in Management in 2005. She is now a PhD student of Computer Science between University of Tunis and Sorbonne University, France. Her research is focused on learning information extraction and pedagogical documents annotation. She teaches computer science at Faculty of Economic Sciences and Management of Nabeul.

Rim Faiz is currently a Professor of Computer Science at the Institute of High Business Study (IHEC) at Carthage, in Tunisia. She obtained her PhD in Computer Science from the University of Paris-Dauphine, in France. Her research interests include artificial intelligence, machine learning, natural language processing, information retrieval, text mining and semantic web. She is member of scientific and organisation committees of several international conferences. She 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 of 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, Hermes, Paris, 1990. He is Director of the Doctoral School ‘Concepts et Langages’ and the LaLIC laboratory (Languages, Logic, Informatics and Cognition) at Paris-Sorbonne.
1 Introduction

Searching learning information has become requested owing to the rapid development of the e-learning concept within the web technologies. 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 e-learning since they can support the learner in finding the necessary information for his learning, training or teaching process.

The term ‘information search’ covers a broad range of situations. People search for information for personal or work purposes, but also in order to learn. Most information retrieval systems that perform well in keyword matching are deficient in semantic aware search as well as inferring queries.

In general, users usually 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 the semantics of either 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 retrieve learning information basing 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 to retrieve pedagogical information. Yet, producing interesting semantic metadata manually is not interesting. In fact, providing a group of fields for users to fill in is one possible solution. However, this solution produces an inflexible system and learners still need to know various types of metadata which depend on the use of language, glossaries, expert opinions, personal experiences and so on. Automatic procedures exist but they can only fill in ‘simple’ and ‘low-added value’ fields (e.g. date, author, title).

Using discursive organisations 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 (Faiz, 2006) for learning objects terms. This paper 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 (definition, 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 extent on context. Add to that it cannot be expressed at a generic level.

This paper describes two dependent automatic engines for semantic annotation and indexation of learning objects based on linguistic knowledge. We will try to explain how linguistic information helps to retrieve relevant learning objects as response to user’s queries.

The rest of this paper is organised as follows: Section 2 deals with the presentation of related works on learning information annotation and retrieval. In Section 3, we present the learning object approach. Our model for learning information retrieval is detailed in Section 4. Before concluding, we illustrate the evaluation results of the different parts of our model in Section 5.

2 Related works

Several works provide infrastructure and services for learning information annotation, indexing and retrieval from documents. Some of these works are mentioned below.

Puustinen and Rouet (2009) 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 ontology. According to Lee et al. (2008), the second one is qualified by the traditional information retrieval technology as keyword-based vector space model (Salton, 1988) and decision trees (Faiz and Elkhliﬁ, 2008; Elkhliﬁ and Faiz, 2010).

Within the first orientation presented above, several works provide infrastructure and services for learning information annotation, indexing and retrieval from documents. Some of these works are mentioned below.

QBLS (Dehors and Faron-Zucker, 2006) 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 practise how to resolve some questions or learn more details about a definition, etc. When the user formulates a request, the search engine Course is activated to search the pedagogical cards as response to the user’s query.

The platform TRIAL SOLUTION (Dahn and Schwabe, 2002; Buffa et al., 2005; Dehors et al., 2006) offers e-books annotations relative to the pedagogical role of the resources contents (definition, theorem, explanation, etc.), the keywords and the relations with other resources (e.g. 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 SYFAK system (Smei and Ben Hamadou, 2005) which presents several annotations indicating: (a) correspondence of the document with the user profile (Yes/Not), (b) the user point of view on the document (interesting, average, not very interesting), (c) the type of documents (TD, TP, etc) based on the ontology ‘Type of documents’ which was created manually and (d) 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 ontology 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 of the user’s request.

For all the systems presented above, the problem of indexing pedagogical documents is discussed from various sides: (a) the course is structured manually according to a pedagogical ontology or a specific architecture (Shi, 2010) in order to use it in an e-learning environment, (b) the course is semi-annotated by users to produce personalised 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 mention the work of Hassan and Mihalcea (2009) who explores 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 data set: educativeness (a 4-point scale ranging from non-educative to strongly educative), relevance (a 4-point scale ranging from non-relevant to very relevant), content categories (definition, example, question and answers), resource type (blog, online book, forums, presentation, etc.), expertise (the expertise of the annotator in each of the selected topics on a 4-point scale). Authors experiment with automatic classifiers (Naive Bayes and Salton Vector Machine) to annotate the educativeness of a given document.

We also denote the SOAF system (Cernea et al., 2008) 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: (a) low-level features which generate automatic semantic indexing, (b) high level descriptors provided by authors (title, date of creation, etc.) and (c) collaborative annotations that are given by users.

Thompson et al. (2003) 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 do not enable to reach the contents of the documents to analyse their textual segments.

Several works are interested in extracting concept’s definition from documents: as in the work of Liu et al. (2003) which is designed for finding salient concepts and definitions of user specified topics. The authors identified many definition identification patterns that are suitable for web pages (e.g. {concept} {is|are} [determiner]). There exists another system called DEFINER (Muresan and Klavan, 2002) which presents a method for automatically extracting definitions and their headwords from text in order to build a new dictionary in the medical domain.

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. This is not, of course, an interesting task for users. When the annotation is automatic, only metadata relative to the whole documents are extracted. Other works extract only definitions from web pages in a specific domain. We support the task of automatically annotating discursive textual segments with semantic metadata relative to different learning categories independently of the domain.

In this paper, we propose a model which aims at automatically annotating learning objects according to their semantic categories (definition, example, exercise, etc.) (cf. Section 3) in order to index and extract learning objects as response to the user’s query. Each category is divided into subcategories. For example, the category ‘Exercise’ can be dived into the subcategories ‘Tutorial’, ‘Case Study’, ‘Application’, ‘Assignments’, ‘Multiple Choice Question’.

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

3 The learning object categories

When working with a computer, learners will manipulate digital artefacts to perform the learning activity they have
been assigned to. With the spread of pedagogical resources on the web, the idea has emerged to capitalise these artefacts by learning objects (Christiansen and Anderson, 2004).

Wiley (2000) 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 purpose 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 six types presented in Figures 1 and 2.

The first interest of learning objects is the creation of opportunities for institutions and instructors in their lesson planning and its execution. Learning objects are considered as cost and time efficient by emphasising annotation, retrieval and reuse over individual creation. We propose to categorise learning objects according to six categories (plan, exercise, example, course, characteristic and definition).

Figure 1 represents learning object categories and relations binding of these objects.

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 events’ in sociology, ‘SQL requests’ in computer science, ‘bacteria’ in biology, …) by exploring many texts (specialised encyclopaedias, handbooks, articles). So, these examples will be included 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 organisation 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.

The first level of the semantic map makes it possible to release six learning object categories: (a) course, (b) plan, (c) exercise, (d) example, (e) definition and (f) 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 …’.

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 model of learning objects retrieval

With the aim of bridging the gap between the information technologies community and the e-learning community, we propose a model which is composed of two engines: the first engine is an automatic annotation system of pedagogical texts according to learning object categories. The second engine uses automatic semantic annotation that is generated by the annotation engine to create a semantic inverted index which is able to find relevant linguistic segments for queries associated with learning object categories such as definition, exercise and example. To validate our model, we implemented the SRIDOP system. The goal of this system is to extract relevant learning objects as response to a user’s query related to a pedagogical concept. Our SRIDOP system architecture is represented Figure 3.
4.1 Automatic annotation of learning objects

A major objective of our annotation engine (Smine et al., 2010; Smine et al., 2011) is to explore the semantics and the discourse organisation of pedagogical texts in order to enhance learning information retrieval through automatic annotation of learning objects according to their categories.

We note that the aforementioned approaches for the annotation of learning objects are semi-automatic or do not reach the segments composing the documents. As well, they are based on existing metadata extraction from files.

The methodology used by our annotation engine, called contextual exploration (Desclés, 1997; Desclés, 2006), call upon knowledge exclusively linguistic and present in the texts. This linguistic knowledge is structured in form of lists and is capitalised in a knowledge base. There are two kinds of lists: indicator lists on the one hand, contextual index lists on the other hand. Indicators are specific to a given object learning category (such as: to recognise a Definition and to filter an important sentence of the pedagogical text). Each indicator is seen as associating a set of heuristic rules of contextual exploration (Desclés and Djoua, 2007). The application of a rule called by an indicator, demands seeking explicitly, in the indicator context, the linguistic indexes complementary to the indicator, in order to be able to solve the task. In addition, it does not need a morpho-syntactic analysis which reduces considerably the execution time of the method (Djoua et al., 2006; Elkhlifi and Faiz, 2010).

Several works, within the LaLIC Laboratory, have applied the contextual exploration method, in one way or another, to segment texts (Mourad, 2002), to annotate events (Elkhlifi and Faiz, 2009), to annotate named entities (Bouhafs, 2005), to identify relations between concepts (Le Priol, 2007), to annotate the specification requirements (Garcia-Flores, 2007), to annotate the definitions existing in texts (Teissedre, 2007), etc.

4.1.1 Linguistic resources

Each linguistic resource represents a set of contextual exploration rules, associated with a learning object category (cf. Figure 4). These rules serve to capture the discursive organisation of a text. Each rule is based on a set of indicators to trigger the rule and a set of clues to apply or not the annotation. These sets of markers (indicators and clues) can be composed of lexical variations or regular expressions (Djoua et al., 2007). For example, the list of indicators relative to a rule of the category ‘Definition’ is:

- identifie (identify)
- identifiable
- a identifié (has identified)
- ont identifié (have identified)
- défini (define)
- définissent
- a défini (has defined)
- ont défini (have defined)
- définiront (will define)
- définira
- définissent

These are grammatical forms of the verb to identify’ and ‘to define’.

We implemented linguistic resources using the access database system. They represent a table composed of the different rules as records. The different fields composing the table are: the rule identifier (IdR), the list of indicators relative to the rule (Indicator), the first clue on the left (CL1), the second clue on the left (CL2), the negative clue on the left (CLN), the first clue on the right (CR1), the second clue on the right (CR2), the negative clue on the right context of the indicator (CRN), the learning category of the rule (learning category). Indeed, we give the permission to the user to manage the EC rule base (adding, updating, deleting rules) through the access database system. The establishment of lists of semantic markers and contextual exploration rules represent the first step in building a linguistic categorisation.

4.1.2 The annotation engine process

The major subdivisions within a semantic categorisation include the structural segments: linguistic marks, search space, indicator, linguistic complement marks and annotation specification from the semantic map (cf. Figure 2).
Relevant learning objects extraction based on semantic annotation

Our annotation engine process is represented as follows:

1. **Pre-processing**: Original documents in HTML-PDF-PPT-DOC are converted to a plain text format.

2. **Segmentation**: The segmentation has an important role in the improvement of the annotation quality. The segmentation is the determination of sentence borders. Given that a point followed by a capital letter is not enough to detect the end or the beginning of a segment, it is necessary to take into account all typographical markers. Moreover, other linguistic bases are engaged like the syntactic structure of a sentence and the significance of each typographical marker in a well-defined context. There exist several works related to the monolingual segmentation, in French language (Dister, 1997), English (Jeffrey et al., 1997) and German (Palmer and Hearst, 1994). Other more recent works consider the multilingual aspect, as Mourad (2002) proposes an approach that consists in defining a textual segment starting from a systematic study of the punctuation marks. We developed our own segmentation engine while basing ourselves on punctuation marks. Due to the great number of the linguistic rules to programme, we have to integrate in our knowledge base all the rules developed in the system Segatex (Mourad, 2002).

3. **Annotation**: Process of learning objects annotation according to their semantic categories:
   - Identification of an indicator in the text (an indicator which belongs to the whole set of indicators).
   - Selection of the rules which have as indicator that was identified in the previous step.
   - For each rule selected, the system generates search spaces parts of texts where the system will search the linguistic clues. These clues will confirm or invalidate the indicator’s value.
   - If the clues relative to the indicator of a selected rule are identified, then an annotation of the current textual segment (sentence, paragraph or document) will be generated according to the learning category (subcategory) of the rule.
   - In case the clues relative to the indicator of a selected rule are not identified, the annotation process with the selected rule of the current textual segment will be cancelled.

The annotation process is expressed in Figure 5.

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

The annotation process, as described above, may fail in the relevant learning object extraction due to the fact that documents unrelated to the subject of learning can be retrieved and shown in the list of results. So, we introduced another parameter to the EC rule which is the emplacement of the query’s term. This parameter permits to perform the indexation process.

The introduction of this parameter is argued by the fact that the place of the term expressed in the user’s query varies according to the rule applied to annotate the learning objects. We illustrate an example to detail this parameter:

- 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 ‘The person 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/right of the indicator, left/right of the clues, title, section title, etc.). Table 1 shows some examples of parameter values.

For each category of the semantic map, we defined the set of rules which covers all the possible linguistic forms of a learning object. We have developed about 180 rules. We started from a textual example to generalise all linguistic structures. This method permits to define incrementally a solid base of rules.

Table 2 shows some examples of rules. In Table 2, IdR denotes the identifier of the rule; CL1, CL2 denote the left clues and CR1, CR2 denote the right clues.

**Figure 5** The annotation process
Table 1 Designation of the term emplacement

<table>
<thead>
<tr>
<th>Designation</th>
<th>The term emplacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIND</td>
<td>The left of the indicator</td>
</tr>
<tr>
<td>RCL1</td>
<td>The right of the first left clue</td>
</tr>
<tr>
<td>LCR2</td>
<td>The left of the second right clue</td>
</tr>
</tbody>
</table>

Table 2 Examples of contextual exploration rules

<table>
<thead>
<tr>
<th>IDR</th>
<th>CL1</th>
<th>CL2</th>
<th>Indicator</th>
<th>CR1</th>
<th>CR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD1</td>
<td>is</td>
<td>are</td>
<td>defined</td>
<td></td>
<td>as</td>
</tr>
<tr>
<td>RD2</td>
<td>is</td>
<td>are</td>
<td>a</td>
<td>an</td>
<td>the</td>
</tr>
<tr>
<td>RC1</td>
<td>The</td>
<td>a</td>
<td>Characteristics</td>
<td>of</td>
<td>is</td>
</tr>
<tr>
<td>RE1</td>
<td>This</td>
<td>an</td>
<td>the example</td>
<td>examples</td>
<td>of</td>
</tr>
</tbody>
</table>

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 CL1 (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’.

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 to annotate the document as a Course Support.

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 (1) or nominal (2), for example:

1. ‘Formulate an SQL clause’ The indicator is the verb ‘Formulate’
2. ‘Exercises on SQL requests’ has as indicator the noun ‘Exercises’

The result of the annotation process is a set of the learning objects existing in the documents and annotated with learning categories.

We realise a learning object annotation engine for French language. Therefore, it can be easily extended to other languages such as English by adapting linguistic resources to English language.

4.2 Indexing annotated objects

Most information retrieval systems perform well in keyword matching but are deficient in semantic aware. Users generally enter keywords into these systems, and the returned results list all web pages containing the same character strings as the chosen keywords. This is not suitable for learning object retrieval. In fact, some words in text semantically unrelated to the retrieval context can be indexed and shown in the list of results. And words that are good index terms defining the learning content of a text are not indexed. For example, in the case of the query ‘Exercises on UML’, documents indexed with the term ‘physical exercise’ can be retrieved. However, documents containing the expressions ‘Answer to questions …’ or ‘Resolve the problem …’ are not provided to the user. This does not suit our pedagogical context.

Djoua and Desclés (2007) proposed an index that would allow the user to search and extract semantic information, from texts, about ‘Causality’, ‘relations between concepts’, ‘quotations’, etc. Inspired from this work, we have developed a model which deals with the annotation of
learning objects according to their semantic categories. Then the storage of these annotated objects to provide an answer not only with a list of documents, but also with the annotated objects corresponding to a semantic-based query in a pedagogical context. Thus, we build an index composed of learning objects, their semantic categories (definition, example, exercise) and the emplacement of the request term (RIND, LCL1, etc.).

We illustrate an example of a query composed of the term ‘UML’ and the semantic category ‘Definition’.

In the SRIDOP system, the indexation engine uses a double structure composed of learning objects on the one hand, their learning categories and semantic indicators on the other hand. The indexation engine does not just use linguistic terms such as ‘UML’ and ‘Definition’, but explores also the linguistic marks of the learning category ‘Definition’ to search the linguistic structures expressing this category.

To enrich the query term (e.g. ‘UML’), we used the basis of synonyms WOLF (which is the portion translated into French Dictionary WordNet) to enrich the query by taking into account all terms equivalent to query term. The latter is replaced by the list of synonyms. This extends the scope of research.

The SRIDoP index organisation is as shown in Figure 6.

The organisation of the SRIDoP index presents the relationship between the textual documents, the annotated learning objects constituting the document and their learning categories. A document contains a set of learning objects identified by the annotation engine. Each annotated learning object is associated with several important pieces of information such as:

- The learning category (definition, example, exercise, etc.). Each annotated learning object can be associated with a set of learning categories (defined by index subfields) used in the annotation engine so that a same learning object can be chosen by the search engine SRIDoP as response for a query about examples and definitions.
- The document folder to identify the document
- The rule identifier applied to annotate the learning object
- The emplacement of the query term in the textual segment
- The text content of the relevant learning objects.

We propose below a query example composed of a term (‘UML’ for this example) and a semantic learning category (‘Exercise’ for this example).

To extract relevant answers, the SRIDoP search engine processes as follows:

1. The SRIDoP system extracts all learning objects found in the index associated with the annotation ‘Exercise’.
2. Selection from these learning objects, all objects within an occurrence for the term ‘UML’ and its synonyms situated in the emplacement according to the rule applied to annotate the object.
3. Display all present information in the index related to each learning object selected.

Figure 6  Organisation of the SRIDoP index
The answer to this query (‘Exercise’ on ‘UML’) gives a set of learning objects annotated with the learning category ‘Exercise’ and containing then term ‘UML’. The emplacement of this latter varies according to the rule applied to annotate the object selected as relevant.

After retrieving learning objects from documents as a user’s query response, in the next section we will present the secondary component of our model which consists of sorting these learning objects.

4.3 Sorting the learning objects

This step is a post-retrieval method that organises the results for higher precision. It is somewhat constrained by the limited time available to perform the filtering and organisation of the results. In fact, users expect short response times from a search engine and may not look favourably on a long post retrieval process.

So, we propose to sort the objects displayed by our system according to their relevance. The objects are sorted in descending order of their similarity with the query subject. Many similarity measures have been proposed (Akermi and Faiz, 2010) in information retrieval domain. But, we implemented a version of Rocchio’s algorithm (Rocchio, 1971), as adapted to text categorisation by (Buckley et al., 1994). 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 database system and so on. We note that the Vector Salton Machine technique (Salton, 1988) can satisfy this assumption by applying the Rocchio’s algorithm.

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 \( V^m = (p^m_1, p^m_2, ..., p^m_l) \), where \( l \) is the number of indexing terms used. We adopted the TF-IDF weighting (Salton, 1991) and define the weight \( p^m_j \) to be:

\[
p^m_j = \frac{f^m_j \log (N/n_j)}{\sum_{j=1}^{l} f^m_j \log (N/n_j)}
\]

Here \( N \) is the number of objects, \( n_j \) is the number of objects in which the term index \( k \) appears, and \( f^m_j \) is:

\[
f^m_j = \begin{cases} 
0 & q = 0 \\
\log(q)+1 & \text{otherwise}
\end{cases}
\]

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 \( \tilde{c}_j \) of the same dimension as the original weight vectors \( v^1, ..., v^N \). For class \( C \), the \( K \)-th term in its prototype is defined to be:

\[
\tilde{c}_j = \frac{\alpha}{|C_j|} \sum_{m \in C_j} p^m_j - \frac{\beta}{N-C_j} \sum_{m \notin C_j} p^m_j
\]

where \( C_j \) is the set of all objects in class \( C \). The parameters \( \alpha \) and \( \beta \) control the relative contribution of the positive and negative examples to the prototypes vector, we use the standard values \( \alpha = 4 \) and \( \beta = 16 \) (Ittner et al., 1995).

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 \( \tilde{O} \) using TF-IDF weighting, and then compared against the class prototype \( \tilde{c}_{user} \) using the cosines measure:

\[
\cos(\tilde{c}_{user}, \tilde{O}) = \frac{\tilde{c}_{user} \cdot \tilde{O}} {||\tilde{c}_{user}|| \cdot ||\tilde{O}||}
\]

The learning objects are displayed to the user in descending order of their similarity with the class \( C_{user} \). It will facilitate the navigation for the user in the subset of results.

5 Corpus

For this study, we constituted a corpus composed basically of pedagogical documents from internet. This corpus covers various domains including database, linguistics, biology, management, etc. in order to demonstrate the capacity of our model in the multi-domain context. 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. We note that, as we are interested in pedagogical documents, the meaning of some terms can be ambiguous, e.g. ‘SQL’ or ‘Maintenance’ and thus we explicitly disambiguate the query by adding other words like ‘Course support’, ‘Assignments’, etc. An example of a query, introduced to Google, is ‘Support of Course on SQL’. 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.

Our corpus is composed of 1000 documents, mainly of learning nature: support of courses, assignments, power point 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.

At present our corpus is exclusively constituted by text in French. A part of this corpus (50 documents) has been used at the base to develop our methodology. For the practical implementation, we decided to use the language Java to process our corpus. In fact, with contextual exploration rules, we annotate the learning object according to their categories. As a result, we obtain a set of learning objects according to their semantic categories.
6 Experimentation and results

We have implemented a learning object retrieval system, called SRIDoP (in French, ‘Système de Recherche d'Informations à partir de Documents Pédagogiques’: An information extraction system from pedagogical documents), on the basis of the proposed model. It included two modules: the learning objects annotation module and the learning objects indexation module. We used the Java language to implement the source code of our system on the NetBeans 6.9.1. We imported the Lucene Library to index our learning objects. Lucene is a high-performance, full-featured text search engine library written entirely in Java.

Several interfaces have been developed to ensure the management of the EC rules (e.g. Figure 7 which permits to add an EC rule).

This is a visual explanation of informatics application for learning objects annotation and extraction.

Through the interface presented in Figure 8, the user can choose the corpus for annotation and indexation. When the user clicks on the ‘Indexer’ button, an interface appears to choose the corpus to annotate and index it. When the documents selected are indexed, the user has to enter the term to search (here is ‘UML’) and the pedagogical object category (here is ‘Definition’). The results are shown to the user when the latter presses the button ‘Rechercher’ (Figure 9).

Figure 7 Interface for rules management (see online version for colours)

Figure 8 SRIDOP interface of the corpus selection (see online version for colours)
6.1 Learning objects annotation

To evaluate this step, our testing corpus was annotated by two experts: for each learning object spotted, they affected to it a type and subtype. The results of the SRIDoP annotation process are illustrated in Table 3.

Precision = \frac{NOAC}{NOA}

Recall = \frac{NOAC}{NOMAC}

F-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}

where NSA is the total number of annotated objects, NOAC is number of objects annotated correctly and NOMAC is number of objects annotated by the experts.

Table 3

<table>
<thead>
<tr>
<th>Learning object type</th>
<th>NOA</th>
<th>NOAC</th>
<th>NOMAC</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan</td>
<td>88</td>
<td>85</td>
<td>98</td>
<td>96.59</td>
<td>86.73</td>
<td>91.40</td>
</tr>
<tr>
<td>Course</td>
<td>72</td>
<td>60</td>
<td>85</td>
<td>83.33</td>
<td>70.59</td>
<td>76.43</td>
</tr>
<tr>
<td>Definition</td>
<td>228</td>
<td>140</td>
<td>266</td>
<td>61.40</td>
<td>52.63</td>
<td>56.68</td>
</tr>
<tr>
<td>Characteristic</td>
<td>139</td>
<td>124</td>
<td>156</td>
<td>89.21</td>
<td>79.49</td>
<td>84.07</td>
</tr>
<tr>
<td>Example</td>
<td>357</td>
<td>349</td>
<td>376</td>
<td>97.76</td>
<td>92.82</td>
<td>95.23</td>
</tr>
<tr>
<td>Exercise</td>
<td>760</td>
<td>705</td>
<td>776</td>
<td>92.76</td>
<td>90.85</td>
<td>91.80</td>
</tr>
</tbody>
</table>

According to the experimentations presented above, the annotation results are promising. Indeed, the precision of the annotation exceeds the 85% for most learning types (example, exercise, plan, etc). But, concerning the ‘Definition’ type, the corresponding precision is average. This derives owing to the fact that certain rules can annotate at the same time objects that reflect or do not reflect a ‘definition’; as the case of a ‘Definition’ type rule which has as an indicator the occurrence ‘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.

6.2 Learning objects indexation

To test this module, we formulated the same 25 queries for each learning type. These queries deal with the 15 topics of the learning and testing corpus. For each learning type, 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 Table 4.

Precision = \frac{NRP}{NR}

Recall = \frac{NRP}{NRRU}

where NR is the total number of results, NRP is number of relevant results and NRRU is number of relevant objects.

At the end of these experiments, we conclude that the results of the document-query matching depend on the annotation results (see Figure 10). The F-score value for the searching process progresses with the annotation process one. This is due to the fact that the learning object i
ndexation 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 above.

Table 4  Experimentation results of the learning objects indexing step

<table>
<thead>
<tr>
<th>Learning object category of the query</th>
<th>NR</th>
<th>NRP</th>
<th>NRRU</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan</td>
<td>72</td>
<td>66</td>
<td>77</td>
<td>91.67</td>
<td>85.71</td>
<td>88.59</td>
</tr>
<tr>
<td>Course</td>
<td>43</td>
<td>35</td>
<td>54</td>
<td>81.40</td>
<td>64.81</td>
<td>72.16</td>
</tr>
<tr>
<td>Definition</td>
<td>156</td>
<td>112</td>
<td>193</td>
<td>71.79</td>
<td>58.03</td>
<td>64.18</td>
</tr>
<tr>
<td>Characteristic</td>
<td>94</td>
<td>86</td>
<td>112</td>
<td>91.49</td>
<td>76.79</td>
<td>83.50</td>
</tr>
<tr>
<td>Example</td>
<td>213</td>
<td>198</td>
<td>230</td>
<td>92.96</td>
<td>86.09</td>
<td>89.39</td>
</tr>
<tr>
<td>Exercise</td>
<td>517</td>
<td>465</td>
<td>520</td>
<td>92.96</td>
<td>89.42</td>
<td>89.68</td>
</tr>
</tbody>
</table>

Figure 10 Retrieval results evolution with those of annotation (see online version for colours)

6.3 Learning objects sorting

This step consists of sorting the learning objects extracted according to their similarity with the class \( C_{\text{user}} \). With reference to many experiments, we have fixed the threshold value \( \theta \) at:

- 0.1 for the Course and Definition categories
- 0.3 for the Plan and the Example categories
- 0.45 for the Characteristic and Exercise categories.

These values are fixed after making many experiments on different learning objects of different categories.

On one side, decreasing the \( \theta \) 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:

\[
\text{Precision} = \frac{B}{A} \\
\text{recall} = \frac{B}{C} \\
F\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

We obtained an average of Precision = 86%, Recall = 75% and 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 indexing ones. There are other parameters which influence the classification results as the training corpus, the choice of the indexing terms, etc.

7 Conclusion

The aim of the present paper is to contribute to extracting relevant learning objects as response to semantic user’s query. We proposed a model for learning objects retrieval from documents. To develop it, we proceeded by a semantic annotation of learning objects, then an indexation of these objects to find relevant learning objects for queries associated with semantic categories (Definition, Example, Exercise, etc.). This work comes within the context of learning objects processing and retrieval. Actually, it constitutes a considerable target in many application domains such as the e-learning domain, training courses domain and data management systems. Through the evaluation results, we observe the originality of a learning object indexation based on a semantic annotation relatively to a keywords searching system. One of the future works that we propose is to extend the semantic map of the pedagogical objects categories by other categories such as method and author. We also look forward to fuse the annotation with a classification method using a score function to perform the accuracy SRIDoP system.

References


Bouhafs, A. (2005) Utilisation de la méthode d’exploration contextuelle pour une extraction d’informations sur le Web dédiées à la veille, réalisation du système informatique JAVA Veille, These de doctorat, Université Paris-Sorbonne.


