International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-12, May 2014

A Multi-Agents System for Extraction and Annotation of Learning Objects

Aziz Oriche, Abderrahman Chekry, Md. Khaldi

Abstract—in this paper, we propose an extraction approach of learning objects (LO) (Definition, Example, Exercise, etc...) of documents defined by HTML / XML whose structure is a tree DOM (Document Object Model). Our approach is based on two intelligent agents. An agent of extraction to extract the learning objects independently of the domain and to align them with the concepts of the ontology. An agent of annotation defined by a set of declarative rules for annotates the nodes and their relationships. We defined in the extraction agent a module includes a set of declarative rules of contextual exploration to extract the learning objects contained in the nodes of DOM documents. The result of this extraction is a set of RDF triples generated for alignment the learning objects extracts with concepts of ontology. The agent of annotation is based on a set of annotations metadata representing the results of extraction and annotation. They allow also annotating the neighbor relationship between the nodes. We assume to have a domain ontology defined by concepts, relations between these concepts and properties.

Index Terms— Semantic Annotation, Multi-agent systems, Ontology, learning object (LO), contextual exploration method.

I.INTRODUCTION

To meet the ever increasing number of resources, the search engines should be able to provide more accurate answers and handle more complex queries integrating knowledge of the user [1]. Given the need for a formal description of the content of teaching materials online, we see that it is interesting to integrate software agents able to label semantically textual or multimedia resources through metadata on the one hand and to exploit to improve information retrieval, on the other hand. The formalization of existing web pages is defined as the task of semantic and automatic annotation. Automating the annotation of educational materials can exploit the contextual exploration technique (CE) [2] which aims to locate and classify elements of the text into predefined categories such as definitions, examples, explanations, exercises, etc [3]. The principle of this technique is the search for indicators associated with domain concepts based on a set of linguistic rules. We propose an extraction agent which allows extracting learning object in the documents and bringing them closer to the concepts of domain ontology. This approach uses the rules of contextual exploration domain independent. Alignment of learning object candidates is done with the concepts of the ontology using semantic labels assigned to learning object discovered in the documents.

Manuscript published on 30 May 2014.
***Correspondence Author(s)**
Aziz Oriche, University Abdelmalek Essaâdi Laboratory, LIROSA & LNTE, BP: 93002, Tetouan, Morocco.
Abderrahman Chekry, University Abdelmalek Essaâdi Laboratory, LIROSA & LNTE, BP: 93002, Tetouan, Morocco.
Mohamed Khaldi, Professor NUS (normal upper school) Martil, Tetouan, BP: 209 Morocco.

This article will be devoted to the description of the learning object extraction agent. It has been developed to analyze at first the texts mainly from teaching French corpus.
Recognition of learning objects uses a method based on both the internal structure of learning object and context analysis (CE). Then we will describe the operation of this agent for the identification and semantic tagging of each learning object and the contextual exploration rules used for the extraction. Then we present the algorithm for extracting and aligning that extracts learning object and aligns with concepts of ontology. Finally, we conclude.

II.DEFINITION OF LEARNING OBJECT

The concept of learning object manages to unite the efforts made by the various interest groups because it brings together a number of assets recognized at different levels economic, educational or technical [4].
Different interest groups, educational institutions, training companies in standardization bodies fail to federate a single and general definition of learning object. In [5], a pedagogical object (PO), or Learning Object (LO) is defined as "any entity, on digital medium or not, can be used for learning, education or training". [6] In turn defines learning objects as "learning objects can be, for example, transparencies, course notes, web pages, software simulation, educational programs, learning outcomes, etc."
A learning object can be reused for different purposes, different platforms, or different audiences [7]. The learning object can take the form of any learning material involved in education to help the learner in his course or in monitoring his career in different platforms online learning.
In our case, the learning object is the content of a node of the tree DOM (Document Object Model, a W3C Recommendation). Each node of the tree represents a structural element (tag) and has a son node containing the text. The nodes are connected by links father / son who represent their nesting tree. The annotation of a node is defined by the presence of one or more instances of learning objects in this node and concepts with which they were associated. An annotated node is an instance of one or more ontological concepts.

III.EXTENDED ONTOLOGY OF ANNOTATION CONCEPTS

An ontology is a structured set of terms and concepts representing the meaning of a field of information by metadata. The ontology is itself a model representative of a set of data concepts in a domain and the relationships between these concepts. In 1993, Gruber suggested a definition that remains until now the most cited definition: "An ontology is an explicit formalization of a common and shared conceptualization" [8].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

In our work, we consider ontology is defined by a set of concepts, relationships between concepts, objects belonging to these concepts and properties. It allows implementing rich and varied relations between these objects.

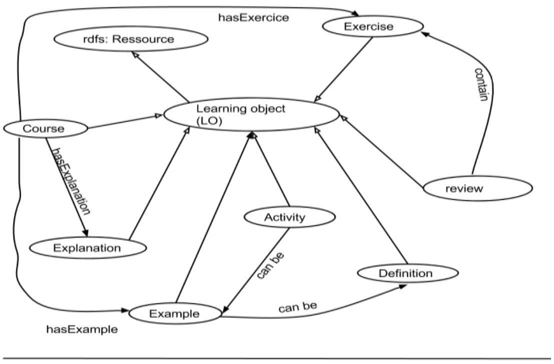


Figure 1: Extract of the domain ontology

The ontology that our system uses for the annotation, defines two sets of concepts: a set of concepts to represent an LO, and a set of concepts describing instances of LO. All instances of concepts are automatically enriched by the information of agent extraction and the set of concepts is enhanced manually by the expert validates the words by changing or deleting some or linking to a more specific or more general concept [9].

In addition to metadata whose semantics is defined in the ontology, we defined a specific set of metadata (Element, LearningElement, LearningObject and the relationship setConceptOf) for the task of annotation; this metadata are defined in [9].

IV.PRESENTATION OF THE EXTRACTION AGENT

The objective of the extraction agent is to generate RDF triples which associating the nodes of DOM document the domain concepts described by learning object that were located in those nodes. Triplets generated using attributes hasValueOf and hasInstanceOf defined for the annotation task. For example, Figure 2 shows the triplets for associating the learning object Explanation the node [http: / / www..../interface.html/ body / .. / p.] That contains:

The main steps of the agent of extraction are:

- XHTML documents relating to the field are cleaned and passed the extraction agent
- The extraction approach starts by applying a set of contextual exploration rules (CE) for candidates learning object for semantic tagging. We call learning object candidate a learning object extract and not yet aligned with a concept of the ontology
- The learning object Candidates obtained are directly aligned with the concepts of the domain ontology and can thus be exploited in subsequent treatments.

We defined an algorithm of locating and alignment; it is described in detail in the section 7.

V.THE KNOWLEDGE BASE OF THE AGENT

In our approach, a semantic tagging task (or contextual exploration task) is defined as a source of knowledge (set of markers and rules of contextual exploration), which represents the expertise to solve a given task. We gather in our knowledge base the different semantic labeling tasks.

A task has the purpose of grouping the rules of contextual exploration and generally corresponds to a semantic labeling process of a node of textual document specified. In a more general manner, a rule may trigger various types of extraction. A task T of semantic label is defined as a source of knowledge comprises of [10]:

- A set of classes of linguistic markers. Two families of classes of markers are used: classes of indicators relevant for the task T and classes of contextual clues that are complementary to indicators;
- A set of contextual exploration rules associated with classes of indicators;
- A set of semantic tags associated with classes of indicators.

His sets are organized in the knowledge base. A semantic label is a semantic value associated with one or more classes of markers. Each task of semantic labeling is composed of a set of rules of exploration contextual and each set of rules is associated with a class of indicators. Each instance of learning object is composed of several labeling tasks containing linguistic knowledge specific to language used.

<p> the declaration of interfaces translates by the interface keyword followed by the name of the interface: [public] InterfaceName interface [extends InterfaceName1, InterfaceName2...] {…………………..} </p>

RDF triples generated by the agent of extraction

(http: //www..../interface.html/body/.../p, hasValueOf, explanation1)
(http: //www..../interface.html/body/.../p, hasInstanceOf, URI(Explanation))

Figure 2: Generation of properties hasValueOf and hasInstanceOf

VI.EXTRACTION OF LEARNING OBJECT

In this section, we present the rules of contextual exploration used by the extraction module of agent to extract the learning object of ontology. We choose to use these rules independent of domain, to locate some groups of linguistic indicators that may represent the learning object of ontology. These rules allow obtaining learning object such as: Definition, Example, Explanation, exercise, etc. they generally have a good reminder because they are not specific to a domain.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In our approach we have defined an alignment method to associate a learning object to a concept in the field and serves also as a filter to the large number of candidates extracted learning objects. We applied a set of rules of contextual exploration to locate and label semantically the learning object candidate before aligned with the concepts of the ontology.

A. Labeling the learning object

Semantic labeling works on the representation of the text produced after cleaning and segmentation. For each text segment (node) in triggers all contextual exploration rules associated with each indicator found in this segment. To

₺ in object-oriented programming, an object is defined by a model called class₺ (1).

recognize the indicators in the text we use linguistic knowledge contained in the knowledge base. Each contextual exploration rule executed for labeling assigns a semantic label to textual segment (node) that contains the indicator that triggered the rule. If the conditions imposed by the rule are met the text segment analyzed will receive the label associated with the rule in the knowledge base.

B. Rules of recognition learning object

To write these rules we have listed all possible forms of learning object of ontology present in the studied corpus. These rules are divided into two parts: the extraction and alignment. The representation of rules is one of the most important aspects in the design of our system because the rules are the core of the process of semantic tagging the learning object contained in text. We explored two paradigms of representation to the applied and adapted to our discussion:

1. Representation rules by transducers;
2. Representation rules by methods of Java classes.

In this section we present the two options considered for the implementation of contextual exploration rules.

C. Representation rules by transducers;

The representation of contextual exploration rules by transducers can work directly on the input text without having to create an instance of the text model [10]. In this approach the verification of a condition of a rule is to see that the text segment (node) analyzed coincides with a pattern Matching. That is why the expression of a condition

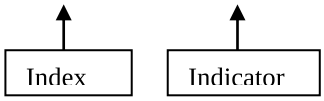
if in text node is found an indicator of the set {example} and this indicator is preceded, at least three words of distance by the index of set {declaration } then assign the semantic label "example1 " to text.

concept "Example". Suppose we want to represent the following rule:

Suppose a document node contains the phrase as a text input for the rule:

this declaration is an example of … .

We add the structural brands to phrase which gives:



We can distinguish in the sentence the blocks of text:
[Block A] declaration [Block B] example [Block C]
Expressing now the conditions of the rule:

1. The index déclaration should belong to the same sentence as the indicator exemple, that is to say, the block B will not contain the tag </ p>
2. The distance between "declaration" and "example" will be equal to or less than three words, that is to say, the block B contain the <w> mark between one and three times.
3. The block D shall satisfy any condition

We can write the first condition as a regular expression that denotes the language of strings that do not contain the string </ p>. The language searched is the complement according Σ * of language of character strings containing at least once the chain </ p>:

The "$" operator of calculation allows us to simplify the expression. $L denotes the language of chains which contain at least one chain of L. The simplified expression is:

~$"</p>"

To express the second condition we will use the operator "$. L" that denotes the language of strings that contain exactly once a chain of L. and the operator "{L ^ n, m}" denotes the language of strings composed with at least n and the maximum m chains sequences of L. The second condition is therefore expressed by:

[$."<w>"] ^ {1,3}

In our example the block B must satisfy the first and the second condition, the blocks A and C must satisfy any condition:

[Bloc A] → Σ*
[Bloc B] → [~$"</p>"] & [[$."<w>"] ^ {1, 3}]
[Bloc C] → Σ*

The regular expression that denotes the conditions of the rule will be:

D. Modeling rules by methods of Java classes

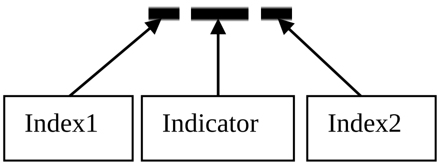
The application of a rule of contextual exploration is analyze the class instances (objects) of text template and create objects of a class of label types. We organize classes so grouped in a class all the rules that assign the same semantic tag. Thus, we have classes RDefinition, RExample, RExercise, etc...

The relationship between tasks, indicators and rules stored in the knowledge base is behind the process of triggering a rule. The elements of this relationship are a rule name, a task, a class of trigger indicators, the node to analyze and a label to assign. Each trigger indicator found in a node causes the release of all the rules that are associated in the knowledge base of the agent. Suppose a document node containing the following text:

Σ* declaration [~$"</p>"] & [[$."<w>"] ^ {1, 3}] example Σ*

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

We add of structural brands to the sentence which gives:
… an object is defined by a model …



Now accept the following assumptions:

- We give the name RDefinition to this rule
- The Indicators class will contain the word "defined"
- The index2 class will contain The auxiliary "is"
- The index1 class will contain the preposition "by"
- This rule will assign the label "definition1" the sentence that satisfies the conditions
- This rule is part of the task "definition"
- The node treated contains the sentence (1).

The implementation of the algorithm of RDefinition rule is shown in Figure 3.

<u>The rule RDefinition</u>

If the text of node contains the indicator "defined"
And if the index "by" to the right of the indicator
And if the index "is" to the left of the indicator
And if the distance between the indices and the indicator = 0
And if there is text before and after the indices
Then assign the label "definition1" to text of the node

Figure 3: The RDefinition rule

For each object type of ontology, we defined a set of rules that cover almost all possible linguistic forms of learning object. We started with a textual example relating to learning object of type "Definition" then we generalize the language structures of other concepts. This method sets incrementally a solid base of rules. For example the learning object of type "Definition" the spotting one of the indicators and the indices of Table 2 is sufficient to annotate the text node as a "Definition". The same the Table 1 shows the rules of concepts "Explanation".

We applied the same approach to other learning objects described by the concepts of the domain ontology. Just to the learning object of type "course", the spotting of the indicator "course " in the title (the <h> tags) is sufficient to mark the node as a course. The nominal indicator of the learning object is the word "course ", and other names like "chapter", "Notes of course", "theme", "competence sight"… Besides the title, the existence of the "Courses" indicator does not imply annotation of the node as a course.

VII. ALGORITHM OF EXTRACTION AND ALIGNMENT

The extraction and alignment algorithm extract the learning object of HTML documents provided input and aligns with the concepts of the ontology. The algorithm is applied to a set of XML/HTML documents in the same domain. It takes as input the ontology O of this domain and set rules of contextual exploration (RE).

Inputs to the function "extract_align":

Index1	Indicators	Index2
	Explain / Influence / Produce / Determine / Impose / enable / facilitate / Foster / Dash / Cause / Provoke/ Trigger / Assign / Train / Generate	
	Succeed/ Conduct/ Lead / encourage / target / Contribute / Help /	to
	translate	by
be	origin / Manager	of
	act	on
have	for	Effect / result

Table 1: Some rules of the "Explanation" concept

Index 1 (left)	indicators	Index 2 (right)
	Is / are	the / him / her / them / an / a/ of
Is / are / be	set/ Defined	By / As / With / Through / By means of
	appoint / mean	
	reveal / Arrested	Like / be
the / him / her / them / an / a/ of	Call / name	the / him / her / them / an / a/ of

Table 2: Some rules for the "Definition" concept

1. The set corpus of XML documents (HTML) to annotate
2. The ontology O represented in OWL
3. The set rules of contextual exploration to extract the learning object.

The output of the function extract_align :
Pre-annotations the nodes of documents represented in RDF.

A. *Algorithm for extracting the learning object*

We begin with cleaning materials of Corpus. The "Extract" function extracts the learning object candidates using the rules of the contextual exploration of the Knowledge Base. These learning objects are stored in the set "Ilo". Each learning object candidate is identified by a sequence of indicators and indices linguistic. For each candidate extracted using (RE), the function "AlignLO" is called (see Figure 4). If the call to function "AlignLO" succeeds, the sets "lo" and "Ilo" are updated by deleting the object "l" of the vector "lo" and adding in vector "Ilo". The algorithm of function "extract_align" and the alignment function is detailed in (Fig. 4). The function "AlignLO" is used by "extract_align" to bring the learning object candidate with the concepts of ontology.

Entries of the function "AlignLO"

1. The ontology "O" represented in OWL ;
2. The learning object "lo" to align;

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-12, May 2014

3. The document "D" to be treated;

The outputs of the function "AlignLO"

1. Annotations in RDF of the nodes of document
2. A Boolean value true if the object "lo" is aligned and false otherwise.

```
extract_align (Set<Document> corpus, Ontology o, Set<Rules> rules)
    variables:
        Set<learning Objects> lo; //a set of learning objects of
        //corpus
        Set<learning Objects> llo; //a set of learning objects
        //aligned with ontology
    begin
        Cleaning(corpus);
        lo=Extract(corpus, rules); //extraction of learning objects
        //candidates using the contextual exploration rules
        for each (l in lo) do
            begin
                if (alignLO (o, l, corpus)=true) then
                    begin
                        add l in llo; //Include in a set of
                        learning objects aligned
                        remove l of lo; // remove "l" of a set of
                        learning objects unaligned
                    end
                end if
            end
        end for
    end
end function
```

Figure 4: Algorithm of Extraction and alignment

B. Algorithm for Alignment the learning object

The function "AlignLO" proceeds thus:

It tests whether the "l" object is present in the ontology that is to say the label of the object lo is equal to a value of members of the ontology. If this is the case, it annotates the node containing the object lo by the corresponding concepts and returns true. Otherwise, the function checks whether the object can be aligned with concepts of ontology via a link of subsumption. If successful this alignment, the "AlignLO" function annotates the node containing the object "lo" by the concept found and returns true. The "generatePreAnnotation" function generates the RDF triples describing annotations node containing learning object we give it as an argument. The "subsume" function approximates a learning object with the concepts of the ontology, it uses the subsumption relation defined in the ontology.

```
boolean alignLO (Ontology o, Learning object lo, Document d)
    variables:
        Set<Concepts> concept;
    begin
        if (label of lo included in o) then
            begin
                concept=retrieve_concept(o, lo);
                generatePreAnnotation(concept, d, lo);
                return vrai;
            end
        else //research of concepts that align with "lo" using the
        //subsumption relation between concepts
            concept=subsume(lo, o);
            if (non-empty concept) then
                begin
                    generatePreAnnotation(concept, d, lo);
                    return true;
                end
            end if
        end if
    end
    return false;
end function
```

Figure 5: Algorithm of alignment

VIII.PRESENTATION OF AGENT ANNOTATION

The agent of annotation is based on a set of declarative rules to generate semantic annotations conforming to the metadata annotations: LearningElement, LearningObject, Element and the relationship setConceptOf defined in [8]. These rules use the properties hasInstanceOf and hasValueOf produced by the extraction agent. We introduce some notations before presenting the rules of annotation for typing nodes and annotate the relationship "setConceptOf" between nodes.

The rules use the following RDFS properties. Let c ∈ set of concepts ontology (Co) and r ∈ set of relations between learning objects (Ro) [11]:

- Domain (r, c) is true if "c" is a domain of r;
- Range (r, c) is true if "c" is a co-domain r;
- subClassOf (c, c ') is true if c' subsumes c;

For clarity in the expression of rules we define the following rules:

- singleLO (n) is true for any node "n" containing only one learning object;
- singleConcept (n) is true for any node n containing one or more learning objects that are uniquely aligned to convergent concepts;
- path (ni, nj) is true if the length of the path in the DOM tree from the node ni and node nj is less than a value "d".

The choice of metadata to be used for annotation of a node depends on the presence of one or more learning objects in this node. This metadata is defined in the ontology (Fig. 1), these rules is [8]:

1. If a node n contains only one learning object aligned to a concept c ∈ Co, this node is typed by c.

singleLO (n) and hasInstanceOf (n, c)→ type (n, c).

Example:

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Example:

> hasInstanceOf (http: //www..../poo.html/body/p, Definition) and not singleLO (http://www..../poo.html/body/p) and not singleConcept(http://www..../poo.html/body/p) → type(http://www..../poo.html/body/p, LearningObject).

2. Any node of LearningObject type is indexed using the relationship isPointed by all concepts "c" connected to this node with the relationship hasInstanceOf.

> *type (n, LearningObject) and hasInstanceOf (n, c)* → *isPointed (n, c)*
> *isPointed (n,c) and subClassOf(c,c')* → *isPointed (n, c')*

Example:

> type (http: //www..../cfp.html/body/p, LearningObject) and hasInstanceOf (http://www..../poo.html/body/p, Definition) → isPointed (http://www..../poo.html/body/p, Definition).

1. We generate instances LearningElement for nodes may represent a set of instances of convergent types. If a node contains multiple learning objects and if they are only aligned to convergent concepts, this node is typed by metadata "s" which is described in the property "hasSet" of the concept "c_i" the ancestor concept of this set of concepts.

> *hasInstanceOf (n, c_i) and singleConcept (n) and not singleLO (n) and $\exists c_j$ (hasInstanceOf (n, c_j) and subClassOf (c_i,c_j)) and hasSet (c_i,s)* → *type(n, s)*

Example:
Is an http://www..../poo.html/body/p node that contains only a learning object aligned to the concept "Chapter" and a learning object aligned to the concept "Courses", and "Chapter" is a subclass of "Course", then the rule is typing the node by LearningCourse.

> hasInstanceOf (http: //www..../poo.html/body/p, Course) and singleConcept(http://www..../poo.html/body/p) and not singleLO(http://www..../poo.html/body/p) and $\exists c_j$ (hasInstanceOf (http://www..../poo.html/body/p, c_j) and subClassOf (Course, c_j)) and hasSet (Course, LearningCourse) → type (n, LearningCourse)

singleLO (http://www..../poo.html/body/p) and hasInstanceOf (http://www..../poo.html/body/p, Definition) → type (http://www..../poo.html/body/p, Definition).

2. If a node n contains several learning objects of more divergent concepts, then this node is typed by the

> *$n_1 \neq n_2$ and hasInstanceOf (n_1, Exercise) and hasInstanceOf (n_2, Example) and Domain (hasExercise, Review) and range (hasExercise, Exercise) and path(n_1, n_2)* → *setConceptOf (n_1, n_2)*

Moreover, the annotation rules exploit the presence of concepts likely to share a semantic relation between neighboring nodes to generate annotations connecting these two nodes by setConceptOf property defined in metadata

> *$n_i \neq n_j$ and hasInstanceOf(n_i, c_i) and hasInstanceOf(n_j, c_j) and Domain (r, c_i) and range(r, c_j) and path(n_i, n_j)* → *setConceptOf (n_i, n_j).*

learning object aligned with a concept "cj" and such that there is a relationship "r" in the ontology for which Domain (r, ci) and Range (r, cj).

Example:

Consider two nodes n1 = http://www.poo.html/body/p and n2 = http://www.poo.html/body/p/ul/li neighbors (distance ≤ d) respectively typed by LearningObject indexed by "Exercise" and "Review" and by "Example".

IX.EXPERIMENTS AND RESULTS

The corpus is composed of HTML documents resulting from the Web, specifically from sites related to educational documents in computing. The constitution of the corpus was done in two stages: (i) loading the corpus on the domain of interest and (ii) pretreatment of the corpus. The loading was done using the Google search engine. We form a set of keywords from the educational objects of the ontology. The loaded documents are preprocessed to obtain a set of XHTML documents trained with tools such as "W3C Validator", "JTidy", "TidyHtml" and the APIs like SAX and DOM. After pretreatment, the corpus consists of 62 non-empty files, containing text, images and other content types.

A. Extraction the learning objects and evaluation

The Table 3 shows the number of learning objects extracted. These learning objects are obtained after semantic tagging of documents in the corpus and application the contextual exploration rules. The evaluation of the accuracy of the extraction of learning objects focused on 1,393 separate educational objects extracts from documents randomly selected.

The majority of learning objects extracted are meaningful but many false learning objects are explained by the presence of syntactic errors: The presence of linguistic markers of a concept in a node does not always mean that this node is linked to the concept. This is the case of many examples like the case of concept "Definition": "The third word is the name of the class", "The name of this class is an agreement".

B. Alignment the learning objects

After extracting the learning objects, the agent of extraction can align with the concepts of ontology. We present the evaluation measures of the alignment step and results obtained.

alignment	Correct (LO)	False (LO)	Recall	Precision
Total LO	1075	32	77.2%	97.1%

Table 4: Precision of the alignment

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

After alignment, we evaluated the results on 1,107 learning objects from 62 documents we selected. Among these learning objects, 1075 were aligned with concepts of ontology. The alignment precision is 97.1% and the recall is 77.2%. The number of false alignments is 32. The results are shown in table 4.

C. Experimentation and evaluation of the annotation

A sample consisting of 30 documents in the corpus, we annotated 758 nodes: 630 (83%) are typed with concepts of ontology, 43 (5.7%) are typed by sub-concepts of LearningElement and 85 (11.2%) were typed by LearningObject indexed with concepts of ontology (Table 5).

	Element	LearningElement	LearningObject
Number of nodes	630	43	85

Table 5: Annotation of the nodes

The large number of nodes annotated by concepts of ontology be explained by the fact that HTML has tags (<p>, div, span) allowing a structured representation of paragraphs and these paragraphs are widely used in educational documents. We got a fairly high number of nodes LearningObject type because these documents also contain a little structured nodes for which their designers do not care about the separation of information provided according to the concepts describing the domain. The low number of nodes LearningElement type is inherent in its very definition.

X.CONCLUSION

We experiment our approach on a corpus of 62 HTML documents from the Web. These documents describe the computer courses. We assessed separately the results of the contextual exploration rules, those obtained by our method of alignment when the execution of the algorithm "Extract_align". This separation allowed us to better evaluate the errors introduced by the method of extraction and alignment.

After applying the rules of annotation, we get different types of nodes (Element, Learning Element, Learning Object). The earning objects semantically related are often, or in the same node, or in neighbor nodes. We have shown that in the absence of semantic relations annotated, it was interesting to represent the possible presence of a semantic relationship using the property set Concept Of.

REFERENCES

1. Aziz O., Abderrahman C., Mohamed K., "Intelligent Agents for the Semantic Annotation of Educational Resources", International Journal of Soft Computing and Engineering (IJSCE), Volume-3, Issue-5, November 2013.
2. Descles J.-P., Jouis C., "L'exploration contextuelle : une méthode linguistique et informatique pour l'analyse informatique de textes", in Proceedings of the International Langage Natural (ILN'93), Nantes, 1993.
3. Boutheina S., Rim F., Descles J.-P., "Relevant learning objects extraction based on semantic annotation", Int. J. Metadata, Semantics and Ontologies, Vol. 8, No. 1, 2013.
4. Jean-Philippe Pernin, "Objets pédagogiques : unités d'apprentissage, activités ou ressources ?", Sciences et Techniques Educatives, 2003.
5. Learning Technology Standards Committee (LTSC) P1484.12.1-2002 Learning Object Metadata Working Group, "Draft Standard for Learning Object Metadata", Available: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, 2002.
6. Bourda Y., Hélier M., "Métadonnées et XML : application aux objets pédagogiques", In Colloque International Technologies de l'Information et de la Communication dans les Enseignement d'Ingénieurs et dans l'industrie, Troyes, 18-20 octobre 2000.
7. Sabine G., K. Ramesh R., "Concept Guide on Reusable Learning Objects with Application to Soil, Water and Environmental Sciences", Dec. 2007.
8. Gruber T. "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, pp. 199-220, 1993.
9. Aziz O, Abderrahman C, Mohamed K., "Agent of semantic annotation of educational resources based on ontology", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-3, Issue-7, December 2013.
10. Gustavo C., "Une plate-forme informatique de l'Exploration Contextuelle : modélisation, architecture et réalisation (ContextO)", université de paris IV – Sorbonne, Ecole Doctorale Concepts et Langages, Laboratoire LaLICC, thèse de doctorat, 2003.
11. Mouhamadou T., « Annotation Sémantique de Documents Semi-structurés pour la Recherche d'Information », thèse en cotutelle, Universités de Paris-Sud et Gaston Berger, 2010.

	Correct (LO)	False (LO)	By expert (LO)	Precision (%)	Recall (%)	F-score (%)
review	50	11	68	82	73.5	77.5
Course	105	13	127	89	82.7	85.7
Definition	190	55	266	77.5	71.4	74.3
Explanation	124	25	154	83.2	80.5	81.8
Example	215	18	258	92.3	83.3	87.6
Exercise	380	47	453	89	83.9	86.4
Activity	43	16	67	72.8	64.2	68.2
Total LO	1107	185	1393	85.7	79.5	82.5

Table 3: Precision and recall of extraction

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)