Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# ENRICHMENT AND POPULATION OF AN EDUCATIONAL ONTOLOGY FROM A CORPUS OF MATHEMATICAL ANALYSIS

**[1]IMANE LMATI, [2] HABIB BENLAHMAR, [1]NACEUR ACHTAICH**
[1]Analysis, Modeling and Simulation Laboratory, University Hassan II, Morocco.
[2] Laboratory of Information Technology and Modeling, University Hassan II, Morocco.
E-mail: [1]lmati2010@gmail.com, [2]h.benlahmer@gmail.com, [1]nachtaich@gmail.com,

**ABSTRACT**

The automatic populating of ontologies from pedagogical textbooks is very hard treatment. Several scientific studies are looking to implement solutions based on linguistic tools to automatically or semi-automatically enrich the ontology by the natural language of mathematical text. But there are still difficulties for the extraction of the meaning of mathematical expressions and their relationships. We present in this paper a methodology for populating of Math-Bridge ontology from the mathematical content of analysis, whose goal is the automatic generation of mathematical exercises from ontology populated. The idea is to use lexical and syntactic patterns for extraction and decomposition of instance attributes into linguistic propositions. A syntactic and semantic comparison of attributes allows extracting semantic relations using WordNet ontology for the textual part and Abstract Syntax Tree AST for the logic part. A first implementation for textual part was developed in JAVA, Framework Jena, and SARQL engine for displaying ontology. We envisage in the further research to develop the second part of the prototype (Logical part).

**Keywords:** *Ontology, Population, Semantic Web resources, Information Extraction, Enrichment.*

## 1. INTROCUTION

Automated extraction of knowledge from mathematical textbooks is very hard treatment which consumes time and resources. Several educational studies are looking to implement solutions based on linguistic tools to automatically or semi-automatically extract relevant information. But there are still difficulties for the extraction of the meaning of mathematical expressions and their relationships. Moreover the volume of information of school programs is important and an assistance to transform this information into ontology is necessary.

The literature contains many definitions of ontology. Roughly speaking, ontologies provide a framework for conceptualization and knowledge modeling in a multitude of areas [5]. Ontology in computer science describes: (1) individuals/instances of a class; (2) classes/abstract groups, sets, or collections of objects; (3) relations/properties, ways that objects can be related to one another.

The contribution of this paper is to show how to extend and populate Math-Bridge ontology [3] using the structure of mathematical document, whose goal is the automatic generation of mathematical exercises [11] based on semantic relationships between the learning objects (Instances of Math-Bridge ontology) derived from a corpus of mathematical analysis. The approach [11] allows build an exercise in reverse: From a question (Qi) for given exercise, we can extend it by other supplementary questions (Q-1, Qi-2...) based on concept instances (Math-Bridge Ontology) and their semantic relationships (Relationship: {Qi, Qi-1}, {Qi-1, Qi-2} ...).

In the next section, we present the relevant works about some pedagogical ontologies conceived from mathematical text and the methods for their enrichment. After, we introduce the approach to populate our ontology, in particular the presentation of lexical-syntactic patterns for extracting instance attributes and the definition of the algorithm for extracting semantic relations and conclude with some open perspectives for this work…

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

## 2. STATE OF ART

To put our research into the context, we summarize the most relevant works about some pedagogical ontologies conceived from mathematical text and the methods for their enrichment. For example, NEVZOROVA [15] has developed a semantic publishing platform for scientific collections in mathematics. Every paper in the collection is dissected into a semantic graph of instances of the supported domain models consisting on three ontologies: Mocassin [18] (Representing mathematical structures: Theorem, Definition…) and SALT document ontology [4] (Representing rhetorical structures: Table, figure, Paragraph …) for extraction of logical structure elements, and OntoMathP RO [14] (Includes two taxonomies: taxonomy of mathematical theories: Algebra, Analysis, Geometry, etc, and taxonomy of mathematical objects: Problem, Method, Formula, etc.) for extraction of mathematical named entities from texts in Russian. For the mining of the Logical Structure, a string similarity based method have used and for semantic relations between them, the system use Decision Tree Learner. The platform is capable to understand the meanings of mathematical notation symbols and interpret them as ontology instances by matching of mathematical variables with noun phrases (Named Entity).
Another works is the project of Solovyev [18] which allows building ontology that captures the structural layout of mathematical scholarly papers. It based on Mocassin and SALT ontologies. The project study methods for extracting structural elements from Mathematical Scholarly Papers. This methods base on two tasks: (i) recognizing the types of document segments; (ii) recognizing the semantic relations between them. For the first task, the system compute string similarity between a string and canonical names of ontology. To recognizing the semantic relations. The system select basic semantic relations between segments from the prior-art models like Navigational Relations and Restricted Relations. For restricted relations like “hasConsequence”, “exemplifies and proves”, it occur between consecutive segments (Relation between segment and its predecessor).
For mArachna project [8]-[9], it generates ontology of the mathematical knowledge extracted from the mathematical text using its narrative structure chosen by the author. MArachna extract the mathematical entities, their relations to each other and their internal structure of the input texts. The system segments the entities into single sentences and uses the abstract syntax tree for representing the structure of the analyzed sentence.

All these works have served as a basis for the development of our approach which is the enrichment of Math-Bridge ontology from mathematical corpus. For knowledge extraction, we separated the logic part and the textual part to facilitate the processing. For the textual part, we have exploited the rate of similarity based on the WordNet ontology. For the logic part, we use the Abstract Syntax Tree AST

## 3. APPROACH

Our approach allows enrichment and population of Math-Bridge ontology by concept instances, instance attributes and semantic relations from a pedagogical corpus without formalization of mathematical text.
The methodology can be summarized in five areas:

1- Extraction of instances (Contextual exploration Technique).
2- Creation of lexical-syntactic patterns to list all instance attributes (Natural language and mathematical formulas).
3- Calculation of the rate of similarity between the textual attributes based on the WordNet ontology.
4- Comparison of Abstract Syntax Trees AST of mathematical formulas (logical attributes).
5- Generation of semantic relationships.

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

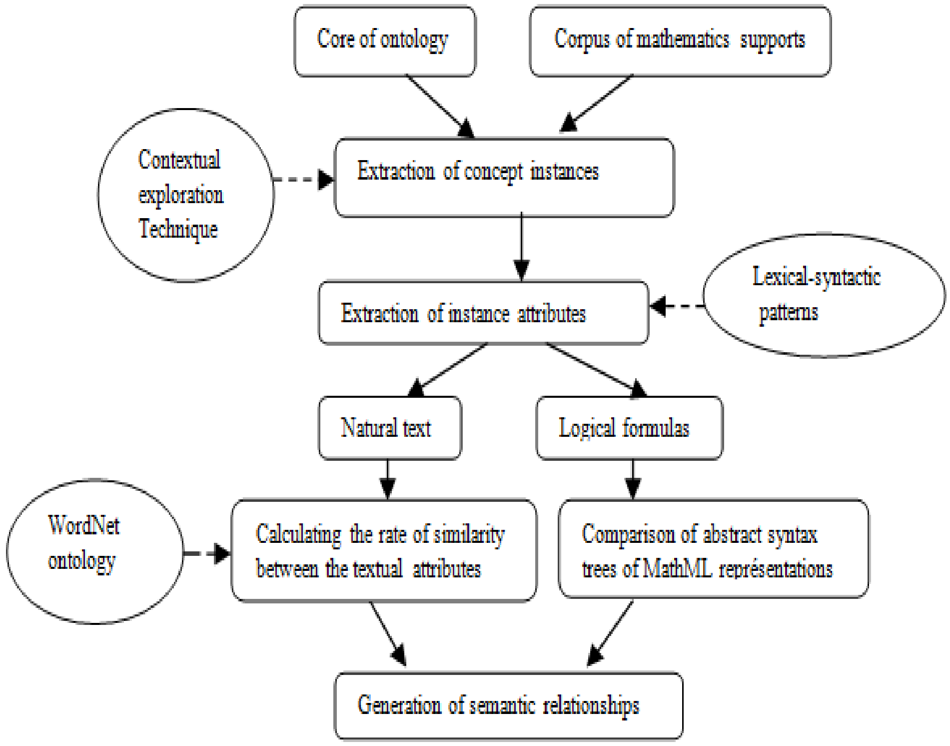



*Figure 1: Process of enrichment and population of Math-Bridge ontology*

Subsequently we detail each axis of the methodology presented.

### 3.1 Core of ontology

According to [10], mathematics texts (article, educational support ...) consist of entities or specific markers we call "math tags" labeled according to specific forms written by authors. The most common tags are "Theorem, Lemmas, and Definition....” The Math-Bridge project allows modeling all of these labels into a core of ontology

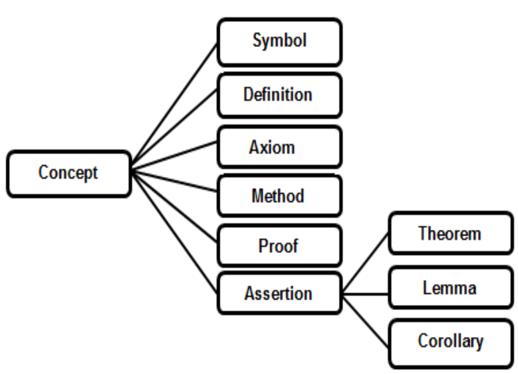



*Figure 2: Core of Math-Bridge ontology*

### 3.2 Instances of concept

To populate the ontology by instances of concept, the project [17] conducted by Smine uses the Contextual Exploration Technique, it allows the extraction of textual segments reflecting educational content based on linguistic rules.

*Table 1: Examples of rules*

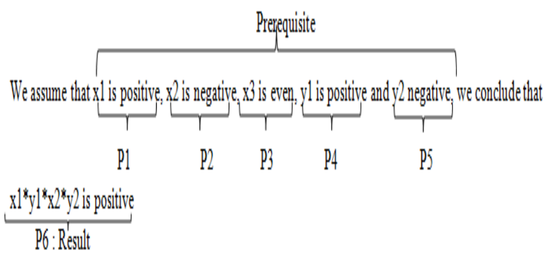
| Rule | CL1 | CL2 | Indicator | CR1 | CR2 | Type / Subtype |
|---|---|---|---|---|---|---|
| R1 | | Is/are | Defined | By | | Definition |
| R2 | | | Is/are | The/a | | Definition |
| R3 | | The/a | Characteristic(s) | of | is/are | Characteristic |
| R4 | | For | Example(s) | It/the/a | | Example |

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 3.3 Instances of attribute

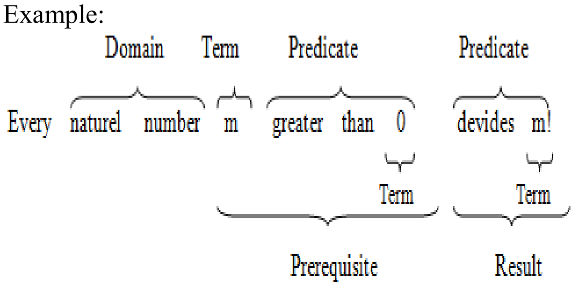
In our previous work [11], we have structured the pedagogical objects PO in three components. First there is the Arg argument representing the variable or variables used in PO. Secondly, there is the prerequisite that designates the constraint of PO and thirdly it is the result of the PO.
It is considered that each attribute of concept (Variable, Prerequisite and Result) consists of (Varible1, Varible2 ... Prerequisite1, Prerequisite2 ... Result1, Result2 ...). According to [6], we can decompose the "Prerequisites" and "Result" in simple proposition (Mathematical sentence) to enumerate the list of attributes. Let P, P2 ... the propositions of the mathematical sentence:




According to this decomposition, each proposition allows to assign simple or quantified variables (X, Y, X*Y, Min(x)…) a property like: Verb, Adjective, Logical formula...
To extract instances of attribute, we operate parsing used by Paskevich [19] to decompose a mathematical proposition, with:

- Domain: to form the domain of definition according to the variable type (element, function ...).
- Term: to form constants, simple variables, quantified variables…
- Predicate: represent property assigned to the term: x positive, Un increasing...
- Symbolic expression: to form a prerequisite or result as a logical formula (arg1 ≤ arg2, arg1= arg2...).

Example:




A symbolic expression can be a prerequisite or result without being tied to a textual predicate.
Example:

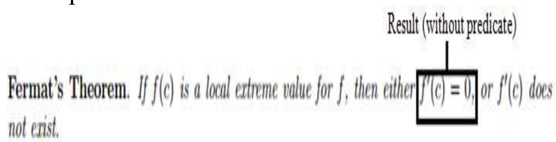
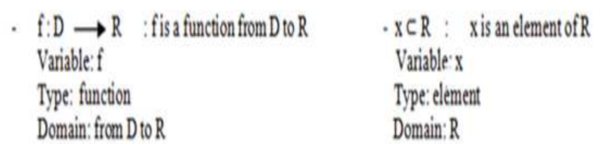



*Figure 3: Fermat's Theorem*

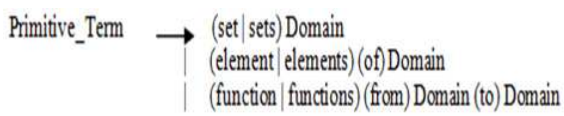
#### 3.3.1 Lexical-syntactic patterns: Term / Domain

In general, we can find in a mathematical text two types of statements: i) natural language, ii) mathematical notation.
Example:




According to the two previous examples, each variable (element, function ...) has a specific declaration.

The project [19] has used the following pattern to declare a variable (term) according to its type:




To solve the problem of logic statement, we can use a pre-annotation in natural language to exploit the lexical-syntactic patterns.

#### 3.3.2 Lexical-synatctic patterns: Term/ predicate

According to Paskevich, there are three primary types of predicate: those built from a primitive verb or adjective, those used to express the class membership « is a» and those used to express the existence of a subordinate object « has ».

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Primitive_Predicate → [does|do][not] primVerbe
| (has|have)[a|an| the] nomPossede
| (is|are|be)[not] primAdjectif
| (is|are|be)[not][a|an]nomClasse
| (is|are|be) [not] termeFixe

PrimVerb → (converges| converge)
| (divides| divide) terme
| (belongs| belong) (to) terme
| ...

primAdjective → (prime)
| (dividing) terme
| (equal) (to) terme
| ...

Each primitive verb or adjective may be followed or preceded by two different terms (Ex: Arg1 divides Arg2). Based on the previous parsing, the model of the ontology of previous work [11] can be extended by other sub-concepts to facilitate the navigation and search of knowledge.

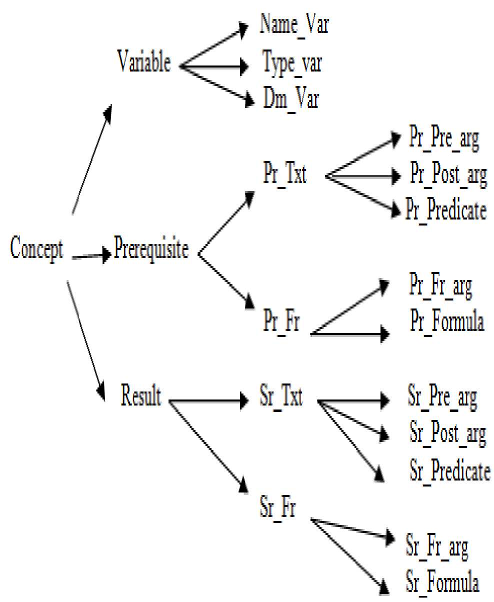



*Figure 4: Extension of educational ontology*

Name_Var, Type_Var and Dm_Var: name, type and domain of Variable.
Pr_Pre_arg / Sr_Pre_arg: List of arguments which precedes the predicate of Prerequisite / Result.
Pr_Post_arg/Sr_Post_arg: List of arguments which follows the predicate of Prerequisite / Result.
Pr_Predicate/ Sr_Predicate: Predicate of Prerequisite / Result.
Pr_Fr_arg/ Sr_Fr_arg: List of argument in the formula (Prerequisite / Result).
Pr_Formula / Sr_Formula: Formula of Prerequisite / Result.

### 3.4 Semantic relation between instances of concept

The semantic description of learning objects defines an interesting number of relationships for navigation or research resources as [2]:

- High substitution (respectively low): a resource R1 substitutes highly (respectively lowly) a resource R2 when the prerequisites of R1 are equal (respectively include) to prerequisites of R2.
- High precedence (respectively low): a resource R1 precedes highly (respectively low) a resource R2, if the result of R1 equals (respectively include) to the prerequisite R2.

To extract the semantic relationships, we must compare textual and logical attributes of instances.

#### 3.4.1 Comparaison of texual attributes based on WordNet ontology

As presented in the previous paragraphs, the Prerequisite / Result are composed of: Prerequisite1, Prerequisite2 ... Result1, Result2...

With: Term (Simple or quantified variable) + Predicate (Verb, Adjective, Name…) →Prerequisite / Result.

Each Prerequisite / Result of an instance will be compared with the attributes of other instances based on predicates and terms. To search the existence of one of the predicates of a given instance in another, we can use the WorldNet ontology [12]. This is an English-language lexical resource; it brings together words (nouns, verbs ...) into sets of synonyms called synsets. Synsets are linked by semantic relationships. For the calculation of the linguistic similarity.The function Syn(c) calculating all Synsets of concept c; let S=Syn(c1)∩Syn(c2) all common sense between c1 and c2 to compare. The cardinality of S is:
λ (S) = |Syn(c1) ∩ Syn(c2)| the similarity between two concepts C1 and C2 will be defined as follows:

$$\mathrm{Sim}(c1, c2) = \frac{\lambda(S)}{\min(|\mathrm{Syn}(c1)|, |\mathrm{Syn}(c2)|)}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

### 3.4.2 Abstract syntax tree of logical formulas

Mathematical formulas can be a Prerequisite or Result without being tied to a verb, adjective ... (Figure 4).
To increase the possibilities to find attributes of instance which use the same formulas, we can browse the Abstract Syntax Tree AST [13] of the MathML representation [1] of logical expressions. After we seek all the possibilities for changing variable to derive the similarity between the formulas (Figure 5).
Example:

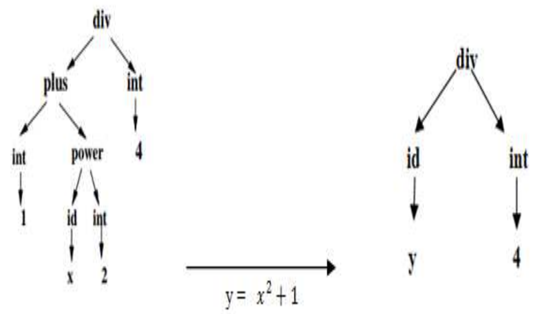



*Figure 5: Change of variable of the abstract syntax tree for the function* $\frac{4}{1+x^2}$

To compare two logical expressions does not have the same number of node, first we compare their first nodes (root) of the tree AST. If they are equal, we compare the descendants.

To compare a descendant with another composed (Node), we use the change of variable as shown in Figure 5. In this case the semantic relationship (Between logical expressions) found shall be annoted by the value of change variable. In this case, we may say that expressions are similar.

### 3.4.3 Algorithm for generating of semantic relations

Thereafter we take the example of the high / low substitution relationship (We just work with prerequisites).
Let O the model OWL of ontology populated by instances and attributes, C1, C2… all instances of concept, Domain(Pr_Pre_arg(i)(.)) domain of pre_arguments of predicates Ci, Domain(Pr_Post_arg(i)(.)) domain of post_arguments of predicate Ci, Pr_Predicate (i)(.) all predicates Ci, i, j, k, indicator, p five integers and AST(Pr_Formula (i)(.)) the abstract syntax tree of prerequisites Ci represented by a formula.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
Start
Choice of starting concept Ci from the model O
Int i= j = k = p = indicator = 0
While there is a Prerequisite for Ci

If Prerequisites (i)(j) is textual
While there is a Prerequisites textual for Ci+1
        If (Domain (Pr_Pre_arg(i)(j) ⊆ Domain (Pr_Pre_arg(i+1)(k))),
        Pr_Predicate (i)(j) and Pr_Predicate (i+1)(k)) are similar,
        Domain (Pr_Post_arg(i)(j)) ⊆ Domain (Pr_Post_arg(i+1)(k))
        A low substitution relationship between Ci and Ci + 1 exists
                Indicator ++
    Else
      K++
End while

If Prerequisites (i)(j) is logic
While there is a Prerequisites logic for Ci+1 {
    If (AST (Pr_Formula (i)(j))) converge to AST(Pr_Formula (i+1)(p)),
      Domain (Pr_Fr_arg(i)(j))  ⊆ Domain (Pr_Fr_arg(i+1)(p))
      A low substitution relationship between Ci and Ci + 1 exists
          Indicator ++
    Else
     P++
End while
Else
    Go to the next Prerequisites of Ci : Prerequisites (i)(j+1)
End while
If indicator = Number of Prerequisites Ci
    A High substitution relationship between Ci and Ci + 1 exists
    Indicator = 0  (Initializing the indicator)
Repeat the algorithm for Ci+1
End
```

*Figure 6: Algorithm for generating of semantic relation*

As show in Figure (6), Once the user selects the starting instance Ci, the system tests the existence of The semantic relations with other instances (Ci+1, Ci+2 ...) based on attributes (Prerequisites/Result) :

For the substitution relationship, we compare each prerequisite Ci with all the prerequisites of Ci+1.

Textual prerequisites:

If one of the predicate Ci and one of those Ci+1 are similar ( Pr_Predicate(i)(j) and Pr_Predicate(i+1)(k)) are similar ) and the domain of definition of variables (Pr_Pre_arg(i)(j),Pr_Post_arg(i)(j) ) used in prerequisite Ci are respectively included or equal to variables of Ci+1 (Pr_Pre_arg(i+1)(k),Pr_Post_arg(i+1)(k)). Then a low substitution relationship between Ci and Ci+1 exists.

Logical prerequisites:

If the abstract syntax tree AST of one of the prerequisites Ci converges to one of those of Ci+1(AST (Pr_Formula (i)(j)) converge to AST ( Pr_Formula (i+1)(p) )) and the domain of definition of the variables used in the prerequisite Ci is included or equal to domain of variables of prerequisites Ci+1 (Domain (Pr_Fr_arg(i)(j) ⊆Pr_Fr_arg(i+1)(p)). Then a low substitution relationship between Ci and Ci + 1 exists.

Before repeating the process for the next instance, the indicator is incremented when there is a low substitution relationship. In other words, at the end of processing each instance, we calculate the number of indicator. If it is equal to the number of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

prerequisite of instance being treated, a high substitution relationship exists.

The algorithm for generating of semantic relations (Low substitution) is represented schematically in the figure below.

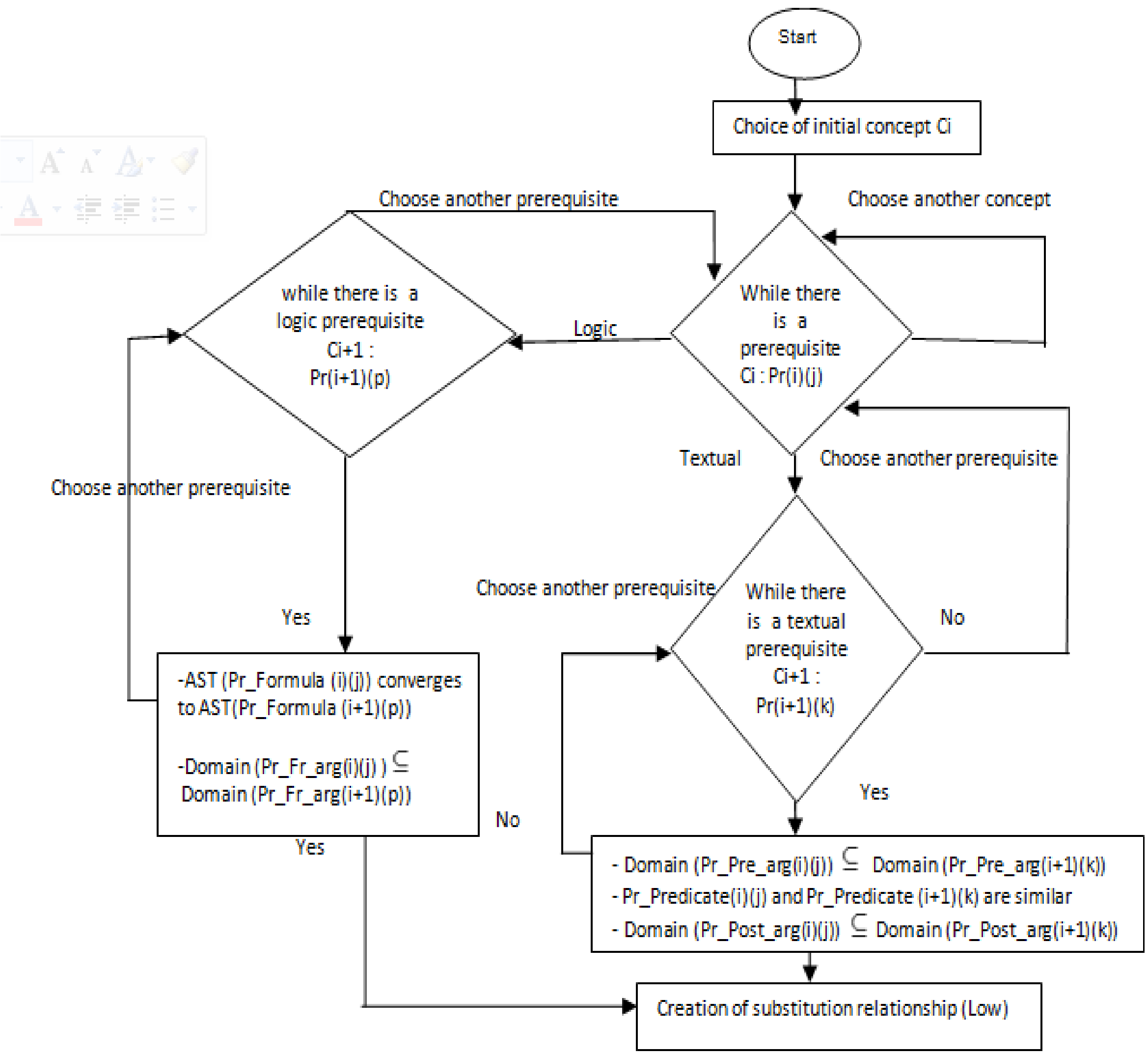



*Figure 7: Algorithm for generating of semantic relations (Low substitution)*

## 4. IMPLEMENTATION AND EXPERIMENTATION

A first prototype instantiation of pedagogical ontology (Textual part) was achieved using the similarity rate of WordNet ontology. The prototype was developed in JAVA, Framework Jena [7], SPARQL engine [16] for the display of the ontology and Framework RiWordNet [21].

To facilitate readability of ontology in the prototype, the name of each educational concept instance was listed with a label (Figure 8): T1, T2… for theorem, D1, D2… for definition…

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

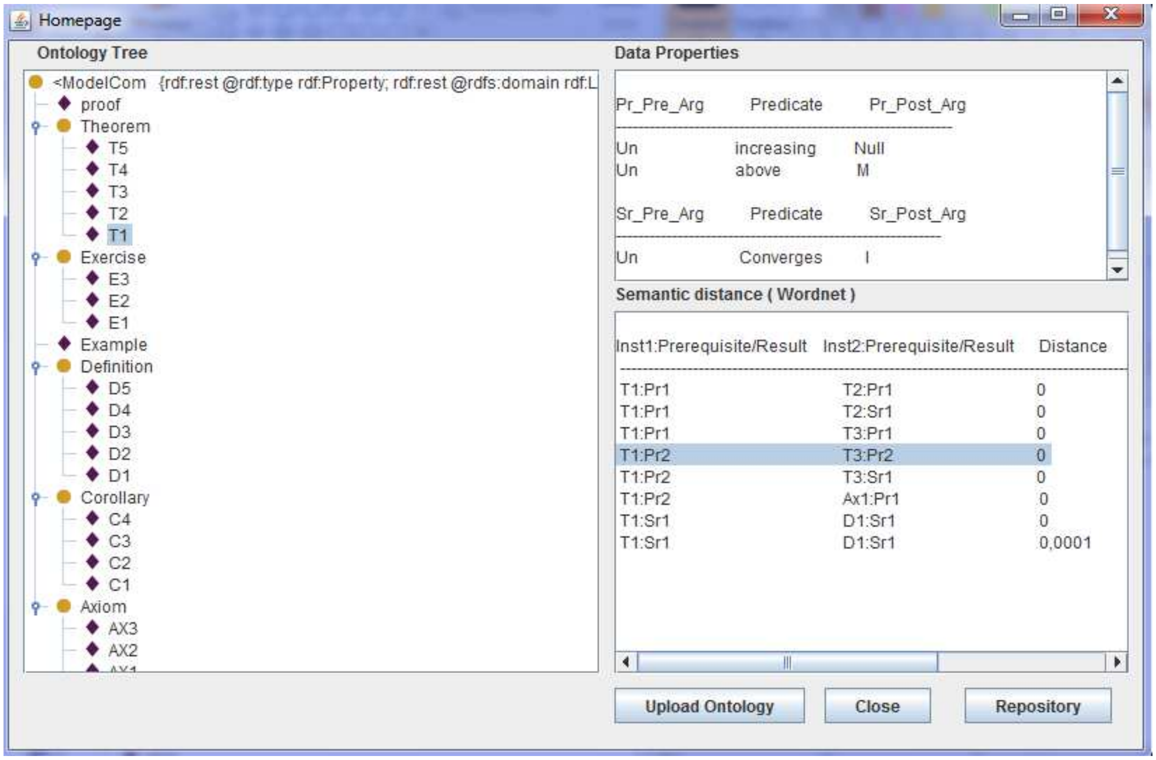



*Figure 8: Homepage of prototype*

The home page (Figure 8) allows editing concepts and instances from ontology downloaded. For each instance, it displays these attributes: Pre_argument, predicate and Post_argument. In parallel, the prototype calculates the rate of similarity of each attribute of the current instance with the attributes of other instances. For example, if we take the case of Monotone Convergence Theorem T1 « if Un is increasing and above then Un converges », the system displays the attributes that are "Un is increasing", "Un above" for prerequisites and "Un converges" for result. After it just displays the rates of similarity that converge 0.

If you click on one of the results of « Semantic distance », a page of domain of definition is displayed (figure 9). The page gives control to the user to determine the inclusion or not of domains of pre_arguments and post_arguments of each attribute of instance ({T1, T2}, {T1, T3}…).

For Pre_Arguments and Post_Arguments compound, we can make use of the contents of variables: Name_Var, Type_Var and Dm_Var.

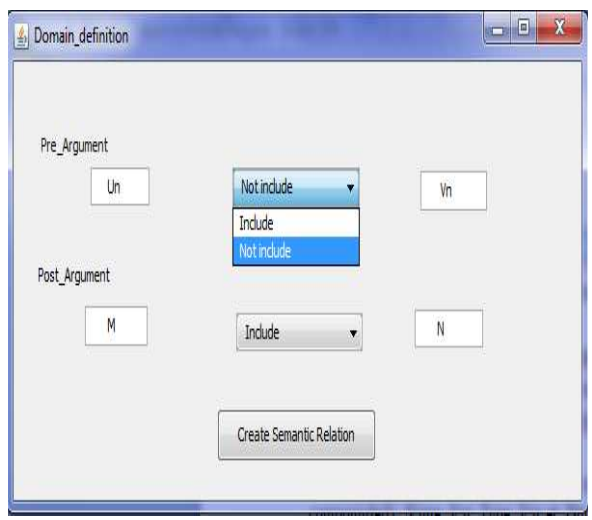



*Figure 9: A definition domain page*

Concerning Pr_Pre_arg, Sr_Pre_arg, Pr_Post_arg and Sr_Post_arg, they can contain several arguments that can be annotated (based on the SPARQL engine) in the order of their appearance in the prerequisites or result (Figure

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10).


*Figure 10: Annotation of attributes in the editor Protégé [20]*

For « x divides y »: "x", "divides" and "y" are annotated by the same number which is 1. The same applies to « z equal to e »: "z", "equal to" and "e" are annotated by 2…

We conducted a first experiment for extracting of semantic relations for three instances: « Theorem of continuity of differentiable functions», « Theorem of continuity on a segment» et « Monotone convergence theorem » applied to 40 supports of mathematical analysis course.

*Table 2: Syntactic segmentation of the three instance*

| Instance | Contents | Attribute | Pre_argument | Predicate | Post_argument |
|---|---|---|---|---|---|
| Theorem of continuity of differentiable functions | Let f a function defined on an interval I and a ∈ I. If f is differentiable at a then f is continuous at a | Prerequisite | f(x) | Differentiable | a |
| | | Result | f(x) | Continuous | a |
| Theorem of continuity on a segment | A continuous function on a segment is bounded. | Prerequisite | f(x) | Continuous | [a, b] |
| | | Result | f(x) | Bounded | Null |
| Monotone convergence theorem | if Un is increasing and above then Un converges | Prerequisite | Un | Increasing | Null |
| | | | Un | Above | M |
| | | Result | Un | Converge | 1 |

To evaluate our approach, we use: recall and precision.

$$\text{Precision} = \frac{\text{Number of relationships correctly generated}}{\text{Number of generated Relations}}$$

$$\text{Recall} = \frac{\text{Number of relationships correctly generated}}{\text{Number of relationships generated by experts}}$$

$$\text{F-Mesure} = \frac{2.(\text{precision}.\text{Recall})}{\text{precision}+\text{Recall}}$$

*Table 3: Qualitative assessment of the number of semantic relations generated for the three instances*

| Instance | Precision (%) | Recall (%) | F-mesure (%) |
|---|---|---|---|
| Theorem of continuity of differentiable functions | 63.63 | 58.33 | 60.86 |
| Theorem of continuity on a segment | 72.72 | 66.66 | 69.55 |
| Monotone convergence theorem | 90 | 75 | 81.81 |

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The method of extraction of semantic relationships gives interesting results (Precision exceeds 60%) but also the limits to apply lexical-syntactic patterns caused by:
• Complex structure of the sentence (implication, parenthesis ...)
• Ambiguity of meaning (Ambiguity of natural language)
• Incorrect labeling (Different statement for variables)

## 5. CONCLUSION

The problem of extracting knowledge from the mathematical corpus still remains a problem which is difficult to treat due to its complexity to treat both textual expressions and mathematical formulas. To facilitate the extraction of mathematical knowledge, we proposed a method, for population of Math_Bridge ontology, based on two parts: Textual and logical part. For the first; we used Contextual Exploration Technique to extract instances of concept. Also the creation of lexical-syntactic patterns allows us to list all the attributes of instance. For the rate of similarity, it facilitates lexical comparison of linguistic attributes for creation of semantic relationship between attributes. Pour the second part; we used the MathML representation of mathematical expressions to compare the logical attributes based on Abstract Syntax Tree AST. To improve the method presented, we envisage further research especially on the following aspects:

1- Use other similarity measures for best results (Textual part).
2- Implementation of the second part of the prototype (logical part).
3- Definition of other lexical-syntactic patterns to reduce the linguistic ambiguity.
4- Creation of patterns for simplification of logical formulas to facilitate their treatments.

## REFERENCES

[1] R. Ausbrooks, S. Buswell , D. Carlisle, et al., “Mathematical Markup Language (MathML) Version 3.0”. W3C Candidate Recommendation of 15 December 2009. World Wide Web Consortium, vol. 13, 2009.
[2] A. Bouzeghoub, B. Defude, J.F. Duitama et al, “Un modèle de description sémantique de ressources pédagogiques basé sur une ontologie de domaine”, Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF), vol. 12, 2005, p. 17 pages.
[3] V. Durand-Guerrier, C. Mercat, et J. Zsidó, “Math-Bridge. Gazette des Mathematiciens ”no 131, 2012, p. 91-96.
[4] T. Groza, S. Handschuh, K. Moller et al, “SALT-Semantically Annotated\ mbox {\ LaTeX} for Scientific Publications”, In : The Semantic Web: Research and Applications. Springer Berlin Heidelberg, 2007. p. 518-532.
[5] T.R. Gruber, “A translation approach to portable ontology specifications”, Knowledge acquisition, 1993, vol. 5, no 2, p. 199-220.
[6] M. Humayoun, et C. Raffalli, “Mathnat-mathematical text in a controlled natural language”, Special issue: Natural Language Processing and its Applications, vol. 46, 2010.
[7] J. Jena, et P. Fuseki, “Semantic web frameworks”, 2004.
[8] S. Jeschke, N. Natho, S. Rittau, et al., “mArachna-Ontology Engineering for Mathematical Natural Language Texts”, In :Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on. IEEE, 2007. p. 1035-1041.
[9] S. Jeschke, N. Natho, et M. Wilke, “KEA-A knowledge management system for mathematics”, In : Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on. IEEE, 2007. p. 1431-1434.
[10] F. Kamareddine, M. Maarek, K. Retel, et al. “Narrative structure of mathematical texts”. In : Towards Mechanized Mathematical Assistants. Springer Berlin Heidelberg, 2007. p. 296-312.
[11] I. Lmati, H. Benlahmar, N. Achtaich, “A new approach for generating exercises based ontologies: A case study of courses in mathematical analysis”. Journal of Theoretical and Applied Information Technology, vol. 70, no 3, 2014.
[12] G.A. Miller, “WordNet-About Us”, WordNet, Princeton University. 2010.
[13] H. Naciri, L. Rideau, “Affichage et manipulation interactive de formules mathématiques dans les documents structurés”. Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées ARIMA, vol. 1 2002,
[14] O. Nevzorova, N. Zhiltsov, A. Kirillovich , et al. “OntoMath PRO Ontology: A Linked Data Hub for Mathematics”, In : Knowledge

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Engineering and the Semantic Web. Springer International Publishing, 2014, p. 105-119.

[15] O. Nevzorova, N. Zhiltsov, D. Zaikin, et al., “Bringing Math to LOD: A Semantic Publishing Platform Prototype for Scientific Collections in Mathematics”, In : The Semantic Web–ISWC 2013. Springer Berlin Heidelberg, 2013. p. 379-394.

[16] E. Prud, et A. Seaborne, “SPARQL Query Language for RDF.(W3C Recommendation) ”, World Wide Web Consortium, 2006.

[17] B. Smine, et R. Desclés, J.P. Faiz2. “Extraction Automatique d'Informations Pédagogiques Pertinentes à partir de Documents Textuels”. Actes des conférences TALN 2011 et Recital 2011, p. 75.

[18] V. Solovyev, et N. Zhiltsov, “Logical structure analysis of scientific publications in mathematics”. In : Proceedings of the International Conference on Web Intelligence, Mining and Semantics. ACM, 2011. p. 21.

Websites:

[19] P. Andrei. The syntax and semantics of the ForTheL language. http://nevidal.org/download/forthel.pdf.

[20] Protégé Project, http://protege.stanford.edu/

[21] RiWordnet API Tool, http://www.rednoise.org/rita/wordnet/documentation/index.htm