ASSESSING SIMILARITY BETWEEN ONTOLOGIES: THE CASE OF THE CONCEPTUAL SIMILARITY

NgoneNgom Aly[1], Kaladzavi Guidedi[2], Kamara-SangaréFatou[3], Kolyang[4], Lo Moussa[5].

[1,3,5]University of Gaston Berger, Saint-louis, Senegal
[2,4]University of Maroua, Maroua, Cameroon

ABSTRACT

In ontology engineering, there are many cases where assessing similarity between ontologies is required, this is the case of the alignment activities, ontology evolutions, ontology similarities, etc. This paper presents a new method for assessing similarity between concepts of ontologies. The method is based on the set theory, edges and feature similarity. We first determine the set of concepts that is shared by two ontologies and the sets of concepts that are different from them. Then, we evaluate the average value of similarity for each set by using edges-based semantic similarity. Finally, we compute similarity between ontologies by using average values of each set and by using feature-based similarity measure too.

KEYWORDS

Ontology, concept, semantic similarity, set theory

1. INTRODUCTION

With the advent of the Semantic Web, research led to the modeling of multiple ontologies, sometimes for the same domain. However, all these ontologies are sometimes heterogeneous (different terms for the same concept, different relations for the same association, different languages, etc.) and this faces the integration problem. Indeed, there are several tasks that imply collaborative use of several ontologies. When several ontologies are used for an application or a system dedicated for a specific domain, it is necessary that these ontologies present some similarities.

Solving the integration problem related to ontologies is to tackle the problem of ontology alignment or mapping. The ontology alignment consists of taking two ontologies as input and taking out a set of correspondences between their elements (concepts, relations, axioms, etc.)[1]. The correspondence evaluation revises the assessment of the semantic similarity between the ontology components.The assessing of similarity between concepts may be very interesting. Indeed, it can make easy the choice of ontologies in the case of elaboration of a system, which uses them and it can help to evaluate the ontology evolution by comparing its different versions, etc.

This paper presents a method for comparing (similarity and difference) ontologies in the case of concepts component. The method is based on the set theory, edges-based semantic similarity [2] and feature- based similarity [3].

The rest of the paper is schemed as follows. In Section 2 we present the definitions of the some core elements. Section 3 entitled related work, reviews some existing methods devoted to the evaluation of similarity between ontologies. Then, the Section 4 depicts our methodology. Section 5 is devoted to some experiments to evaluate and validate the proposed methodology. The paper ends with a conclusion and future work in Section 6.

DOI : 10.5121/ijwest.2018.9401

2. DEFINITIONS

We present in this section the definition of some core concepts, which could facilitate the understanding of the paper.

2.1. ONTOLOGY

The foundational definition of ontology is proposed by Gruber [4][5]: An ontology is "an explicit specification of a conceptualization". The exact meaning depends on the understanding of the terms "specification" and "conceptualization". According to Genesereth and Nelson[6], conceptualization is a "set of objects, concepts, and other entities that are presumed to exist in some areas of interest and the relationships that hold them". In the Gruber's definition, it's not clear that specification depends on the logical view of ontologists. That is why Guarino and Giaretta introduced the logical theory instead of mere specification. Afterward, Borst[7] enriches the previous definition by adding consensual fact related to knowledge modeling discipline characteristics, such as sharing and reuse. For him, "Ontologies are defined as a formal specification of a shared conceptualization". Finally, Studer et al. [8] merge the existing definitions. For them, "An ontology is a formal, explicit specification of a shared conceptualization". They underline the necessity of formal, explicit and shared paradigms. Even if, it's the merging of the existing definitions, it seems consensual. It is more cited in recent years, demonstrating its compliance with the expectations of the knowledge-based systems designers [9]. In addition, the explicity, formality and share-abilityof the knowledge features in an ontology are carried out by five elements[10]: concepts, relations, functions, axioms, and instances.

2.2. CONCEPT

A concept constitutes a think about something, semantically evaluable andcommunicable [12]. It can be abstract or concrete, elementary (electron) and composite (atom), real or fiction. In short, a concept is a notion that represents synonymous terms or terms representing the same thing in different languages. A concept could be the description of a task, a fact, a function, an action, a strategy, a process, etc. For exemple in an ontology of a library, a "*book*" can be considered as a concept, which refers to the term "*livre*" in French, to the term "*book*" in English, to the term "*Buch*" in Deutsch, to the term "*libro*" in Spanish, to the term "*Derewel*" in Mafa (Cameroonian local language), etc. Thereby enables to the ontology on-based intelligent agents to reason and to inter-comprehend (semantic interoperability) on knowledge as would humans do.

2.3. SEMANTIC SIMILARITY

Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation (e.g. their string format) [13][14][15].From an ontologies point of view, [16][17] consider that two concepts are similar if they are "geographically" close to each other in a conceptual hierarchy.Thus, there is semantic similarity between two concepts (for example, movie dog and comic dog) if:

- From an intensional point of view, the two concepts share a large proportion of their descriptive and functional properties;
- From an expressional point of view, the two concepts share a large proportion of the terms that denote them (for example, Dog, Toutou, Crab, etc.);
- From an extensional point of view, the two concepts share a large proportion of their instances (eg Snowy, Rantanplan, Idefix, etc.).

3. RELATED WORK

The following are some works about similarity between concepts ofontologies. Maedche and Staab [17] propose a method for comparing two ontologies. This method is based ontwo levels:

- The Lexical level, which consists of investigation on how terms are used to convey meanings ;
- The Conceptual level which is the investigation of what conceptual relations exist betweenterms.

The Lexical comparison allows to find concepts by assessing syntactic similarity betweenconcepts. It is based on Levenshtein [18] edit distance (ed) formula, which allows to measure theminimum number of change required to transform one string into another, by using a dynamicprogramming algorithm. The Conceptual Comparison Level allows to compare the semantic ofstructures of two ontologies. Authors use Upwards Cotopy (UC) to compare the Concept Match(CM). Then, they use the CM to determine the Relation Overlap (RO). Finally they assess theaverage of RO. This approach allows to assess similarity between two ontologies by using theLexical andConceptual Comparison Level. However, if we reverse the position of someconcepts in thehierarchy, we can get the same results because the method only considers thepresence of theconcept in the hierarchy.

In [19], authors implement an online ontology comparison tool, which can give a numeric measurement of the difference between two ontologies. The given tool is based on senses refinement (SR) algorithm, which makes use of concepts and senses retrieved from WordNet [27]. The algorithm that implements SR considers the subsumption relation "is-a" (hyponymy) and constructs a set of concepts for each ontology (the source ontology and the target ontology). Each set contains concepts of ontology and synsets of concepts. A synset is a set of concepts that are synonyms. Since a concept can have several meanings in WordNet (polysemy), then the algorithm chooses concepts of the synset that is related to the same semantics as the studied concept. Once the sets of concepts have been formed for each ontology, the ontologies are compared, by assessing their difference. The difference value is obtained by applying the Tversky measure[3]. The method of [19] allows to compare two ontologies on the basis of their difference. This method uses set theory as our proposition in this paper. But, it only gives as result, the value of difference between the two ontologies. Contrary to our method, which evaluates the similarity of the ontologies by taking into account their differences.

4. METHODOLOGY

4.1. PRINCIPLE

The approach we propose is based on the set theory, edges-based semantic similarity [2] and feature-based similarity [3]. We consider ontology as a set of concepts linked together by semantic relations. The main aim of this paper is to compare two ontologies. For this, we compare sets of elements of ontologies by using feature-based similarity rules. Feature-based similarity was introduced by Tversky [3]. In his work, Tverski assess similarity between ob- jects by taking into account their common points and their differences. Figure 6 represents Tversky's feature model. In this figure, we have :

- $S_1$and $S_2$are sets of elements ;
- ($S_1 \backslash S_2$) (respectively ($S_2 \backslash S_1$)) represents set of elements present in $S_1$and not in S_2 (respectively presentinS_2 andnotinS_1);

- $(S_1 \wedge S_2)$ is the intersection between S_1 and S_2; i.e the common elements of sets S_1 and S_2. The Tversky measure is given by the formula 1.

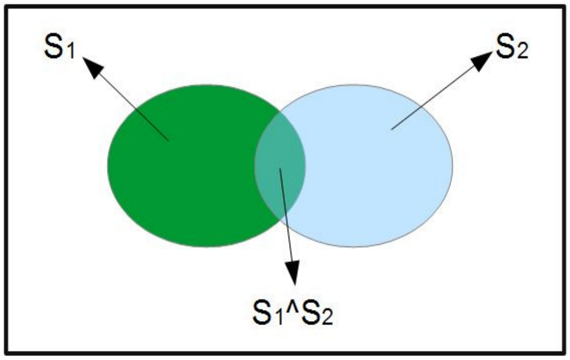

Fig. 1. Example of Tversky's feature model.

The Tversky measure is given by the formula 1.

$$Tvr_{(S_1,S_2)} = \frac{f(S_1 \cap S_2)}{f(S_1 \cap S_2) + \alpha . f(S_1 \setminus S_2) + \beta . f(S_2 \setminus S_1)} \qquad (1)$$

In the formula 1, we have :

- f represents a function that reflects the salience of a set of features ;
- α and β are parameters, which allow expressing the non-resemblance factors between S_1 and S_2 .

In our case, we have to assess similarity of two ontologies (O_1 *and* O_2). By analogy with the Tversky's feature model, figure 2 gives representation of ontologies O_1 and O_2. In figure 2, we distinguish three parts :

- $(O_1 \setminus O_2) = \{A, C, E\}$: set of concepts present in O_1 andnotinO_2 ;
- $(O_2 \setminus O_1) = \{R, S, T, W, X, Y\}$: set of concepts present in O_2 and not in O_1 ; $(O_1 \wedge O_2) = \{B,D,F,G\}$:setofconceptspresent in O_1 and O_2.

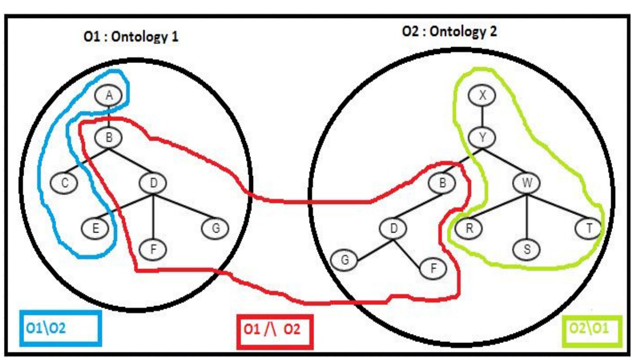

Fig. 2. Representation of ontologies O_1 and O_2 with Tversky's feature model.

The approach can be summarized in 3 steps :

- The Step 1 consists to determine the sets ($O_1 \backslash O_2$), ($O_2 \backslash O_1$) and ($O_1 \wedge O_2$).
- Once the sets are determined, we assess the average of the semantic similarity values between concepts of each set in the step 2.
- Finally, in the step 3, we assess similarity between ontologies by using the results of the step 2 in our measure which is a redefinition of the Tversky measure

4.2. MEASURES

To assess similarity between concepts of two ontologies, we define a measure, which readjusts the Tversky measure. We rely on the tversky measure because it is a reference in the context of feature-based similarity. In addition, Tversky measure inspired many works like [20] and [21]. The measure we propose takes intoaccount the shared features and differences of ontologies. Referring to figure 2, we have the following sets : ($O_1 \wedge O_2$)={B,D,F,G},($O_1 \backslash O_2$)={A,C,E}and ($O_2 \backslash O_1$) = {R, S, T, W, X, Y}. Applying the Tversky measure, the similarity between O_1 and O_2 is given by the formula 2.

$$Tvr_{(O_1,O_2)} = \frac{f(O_1 \cap O_2)}{f(O_1 \cap O_2)+\alpha.f(O_1 \backslash O_2)+\beta.f(O_2 \backslash O_1)} \qquad (2)$$

Instead of the function f, we use one of the edge-based semantic similarity measures that we studied in [2]. For every determined set, we will compute the average of the similarity values between concepts. In [2], we studied edge-based semantic similarity measures. In [22] and [23], we used the measure of Zargayouna and Salotti [24], which extends the measure of Wu and Palmer measure [25]. The measure of Zargayouna and Salotti presents a good correlation with human judgement defined by Miller and Charles [26], but the problem is this measure doesn't take into account the similarity of concepts, which are not in different hierarchy. In this paper, we use the measure of Wu and Palmer because it presents good correlation with the human judgement of Miller and Charles. Using Wu and Palmer similarity measure, the similarity between two concepts c_1 and c_2 is given by the formula 3.

$$Sim(c_1, c_2) = \frac{2 \times depth(c_3)}{depth(c_1) + depth(c_2)} \qquad (3)$$

The concept c_3 represents the Least Common Subsumer (LCS) of concepts c_1 and c_2.By replacing the terms of the Tversky measure with the average of the similarity values between concepts of the determined sets, formula 2 becomes formula 4.

$$T_{Ngom(O_1,O_2)} = \frac{\theta.\overline{x}_{O_1}+\omega.\overline{x}_{O_2}}{\theta.\overline{x}_{O_1}+\omega.\overline{x}_{O_2}+\alpha.\overline{y}_{(O_1 \backslash O_2)}+\beta.\overline{z}_{(O_2 \backslash O_1)}} \qquad (4)$$

With

- $\theta = \frac{cardinality(O_1 \cap O_2)}{cardinality(O_1)}$;
- $\omega = \frac{cardinality(O_1 \cap O_2)}{cardinality(O_2)}$;
- $\alpha = \frac{cardinality(O_1 \backslash O_2)}{cardinality(O_1)}$;
- and $\beta = \frac{cardinality(O_2 \backslash O_1)}{cardinality(O_2)}$;

• *cardinality*(*O*) is the number of elements (concepts) of the set (ontology) *O*

and where :

- x_{O1}(respectivelyx_{O2})istheaveragevalueofsimilarity between concepts (x_i, x_j) in ontology O1 (respectively (x_i, x_j) in ontology O2). i, j ∈N and i≠j.
- $y_{(O1 \backslash O2)}$ (respectively $z_{(O2 \backslash O1)}$) is the average value of similarity between concepts (y_i,y_j) (respectively (z_i,z_j)) present in ontology O1 but not in O2 (respectively present in ontology O2 but not inO1).I,j∈Nand i= j.
- The coefficien ts θ, ω, α and β allow to take into account the similarity values in relation to the number of concepts of the sets and number of concepts of ontologies.

The measure presented by formula 4 respects this properties :

- The measure is symmetric : $TNgom(O1,O2) = TNgom(O2,O1)$;
- The measure is bounded between 0 and 1 ;
- If $TNgom(O1,O2) = 1$ *then* $O1 = O2$.

4.3. ALGORITHMS

In this section, we present the designed algorithms to assess similarity between concepts of ontologies. This algorithms are based on the different steps that we mentioned in section 4.2. The algorithms 1 and 2 respectively aim to form the sets of concepts that represent differences and resemblances between concepts of two ontologies *O*1 and *O*2.

Algorithm 1: diffOnto */* Differences between two ontologies */*

input : $stackO_1$, $stackO_2$: stack of concpts;
output: $stackDiff$: stack of concepts;
variables : c : concept ; $stackDiff$: stack of concepts;
while $(!stackO_1)$ **do**
    $c \leftarrow$depilate (stackO_1) ;
    if $(!checkConcept(c, stackO_2))$ **then**
        $stack(c, stackDiff)$;
    end
end
return $stackDiff$;

Fig. 3. algorithm 1

Algorithm 1 allows to extract the difference between two ontologies. In input, we have two set of concepts stored on stacks. *stackO1*(*respectively stackO2*)stores all *O1*'s concepts *(respectively O2's concepts)*. The function *checkConcept(c, stackO2)* checks if concepts c is present in ontology *O2*. If c is not in *O2*, then c will be added in the stack of difference *stackDiff* In output, the method *diffOnto(stackO1, stackO2)* returns a stack of concepts, which represents all concepts present in *O1*and not in *O2*.

Algorithm 2 called *cCOnto(stackO1, stackO2)* allows to get the set of concepts that belong to ontologies *O1*and *O2*. In input, as for algorithm 1, we have the stacks of concepts *stackO1*and *stackO2,*which store respectively the concepts of *O1*and *O2*. The method *sizeOf (stackO1) (respectively sizeO f (stackO2))* gives the size of the stack *stackO1 (respectively stackO2)* and the method checkConcept checks, if a concept belongs to the stack of concept. If the result is true, then the concept is added in the stack stackCommon. In output, we have a stack of concepts *stackCommon,* which stores all concepts that belong to *O1*and *O2*.

```
Algorithm 2: commonConceptOnto  /* Resemblance
between two ontologies */
input  : stackO1, stackO2 : stack of concpts;
output: stackCommon : stack of concepts;
variables : c : concept ; stackDiff : stack of
  concepts;
if (sizeOf(stackO1) ⩽ sizeOf(stackO2)) then
  | while (!stackO1) do
  |   | c ←depilate (stackO1) ;
  |   | if (checkConcept(c, stackO2)) then
  |   |   | stack(c, stackCommon) ;
  |   | end
  | end
end
else
  | while (!stackO2) do
  |   | c ←depilate (stackO2) ;
  |   | if (checkConcept(c, stackO1)) then
  |   |   | stack(c, stackCommon) ;
  |   | end
  | end
end
return stackCommon ;
```

Fig. 4. algorithm 2

Algorithm 3 is defined to assess the average of similarity values between concepts of a set of concepts (set of differences and set of resemblance).

```
Algorithm 3: aSV /* (Average Similarity Value) Assess the average of similarity values between concepts of a set of concepts */
input : stackConcept : stack of concpts; O : Ontology ;
output: average : real;
variables : c1, c2 : concept ;
/* indexStack is used as pointer of stack */
indexStack : stack of concepts;
meter : integer ;
valueSim : real ;

meter ← 0;
valueSim ← 0;
while (!stackConcept) do
    c1 ←depilate (stackConcept);
    /* to fix the pointer in the first element of the stack of concepts after calling the depilate function, the first element of the stack has changed and becomes the element after c1.*/
    indexStack ←stackConcept;
    while (!indexStack) do
        /* To recover the first element of the stack without to depilate it */
        c2 ←indexStack→value;
        valueSim ←valueSim+Sim(c1, c2, O) ;
        meter + + ;
        /* To move the pointer to the next element */
        indexStack ←indexStack→next;
    end
end
if (meter != 0) then
    average ←valueSim/meter;
end
return average ;
```

Fig. 5.algorithm3

In input, we have a stack of concept (*stackConcept*) and an ontology (*O*). The stack *stackConcept* represents a set of concepts (set of difference or set of resemblance). The algorithm compute similarity between all concepts of the set and assess the average of values of similarity. For that, we extract the first concept out of the stack and fix a pointer to the new first concept of the stack in the goal to assess similarity between concepts. The function $Sim(c_i , c_j , O)$ ($i, j \in$N and $i \neq j$) implements an edge-based semantic similarity measure among measures studied in [1]. The variable *meter* allows to count the number of similarity values evaluated and *valueSim* is the sum of similarity values. These operations are repeated until there is no concept in the stack *stackConcept*. In output, the algorithm computes the average and returns it as the final result of the algorithm.

The algorithm 3 gives the average of semantic similarity values of a set of concepts in an ontology. Since edge-based similarity measures are symmetric, then instead to select $Sim(c_i,c_j,O)$ and $Sim(c_j,c_i,O)$ in the calcul, we choose one of those values, because $Sim(c_i,c_j,O) = Sim(c_j,c_i,O)$. The algorithm also does not assess the similarity between a concept and itself.

Algorithm 4: simOnto */* Assess similarity between two ontologies */*

input : O_1, O_2 : ontology ;
output: $valueSim$: real

variables : $stackO_1$, $stackO_2$, $stackCommon$, $stackDiff_{(O_1\setminus O_2)}$, $stackDiff_{(O_2\setminus O_1)}$: stack of concepts ;
α, β, θ, ω : real ;
x, y, z : real ;
$valueSim$: real ;

/ Store O_1's concepts in $stackO_1$ (respectively O_2's concepts in $stackO_2$) */*
$stackO_1$ ←stack(O_1) ;
$stackO_2$ ←stack(O_2) ;
/ create the stack of difference between O_1 and O_2 (respectively O_2 and O_1) */*
$stackDiff_{(O_1\setminus O_2)}$ ←diffOnto(stackO_1, $stackO_2$) ;
$stackDiff_{(O_2\setminus O_1)}$ ←diffOnto(stackO_2, $stackO_1$) ;
/ create the stack of resemblance between O_1 and O_2 */*
$stackCommon$ ←cCOnto(stackO_1, $stackO_2$) ;
/ initialization of α, β, θ and ω */*
θ ←sizeOf(stackCommon) / sizeOf(stackO_1) ;
ω ←sizeOf(stackCommon) / sizeOf(stackO_2) ;
α ←sizeOf(stackDiff$_{(O_1\setminus O_2)}$)$/sizeOf(stackO_1)$;
β ←sizeOf(stackDiff$_{(O_2\setminus O_1)}$)$/sizeOf(stackO_2)$;
/ partial computation of similarity values */*
$x \leftarrow (\theta \times aSV(stackCommon, O_1))+ (\omega \times aSV(stackCommon, O_2))$;
$y \leftarrow \alpha \times aSV(stackDiff_{(O_1\setminus O_2)}, O_1)$;
$z \leftarrow \beta \times aSV(stackDiff_{(O_2\setminus O_1)}, O_2)$;
/ Computation of similarity values between O_1 and O_2 */*
if $((x+y+z)! = 0)$ **then**
    $valueSim$ ←x/(x+y+z);
end
else
    $valueSim \leftarrow -1$;
end
return $valueSim$;

Fig. 6. algoritm 4

Finally, the algorithm 4 implements the formula 4. In algorithm 4, in input, two ontologies O_1 and O_2. The ontologies are stored on stacks $stackO_1$ and $satckO_2$ thanks to a function $stack(O)$, which stores all concepts of an ontology O in a stack. After storing the concepts in the stacks, the sets of resemblance and difference are determined by calling the algorithms 1 and 2. Once the sets have been determined, we initialize parameters α, β, θ and ω by using the size of sets, thanks to the function $sizeOf(stackO)$, which allows counting the number of concepts in a stack $stackO$. Finally, we compute similarity of two ontologies and return the final result. The result is equal to -1 if there are errors in the calculation process.

5. EXPERIMENTATION

This section is devoted to the experimentation of the proposed method. We use semantic similarity measure for assessing similarity between concepts before computing the average similarity values of set of concepts (set ofresemblance and set of difference). We illustrate our proposal with following examples.

Example 1 : The example 1 is about a fragment of Wordnet[1] that weused in our previous works [3] and [4]. The ontologies are represented by figures 7 and 8.

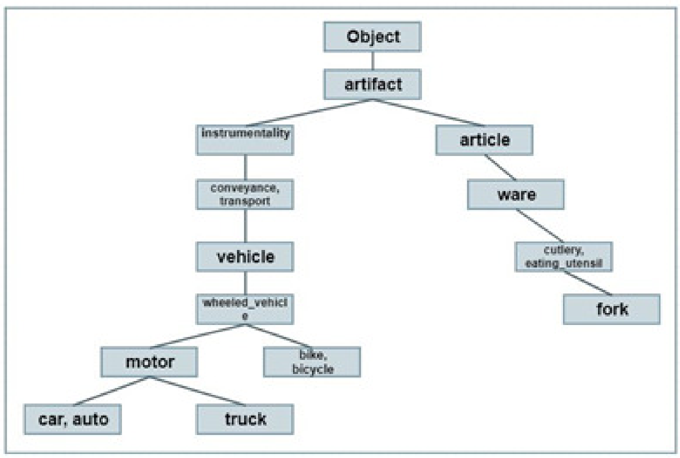

Fig. 7. Representation of an ontology extracted from WordNet (O_3).

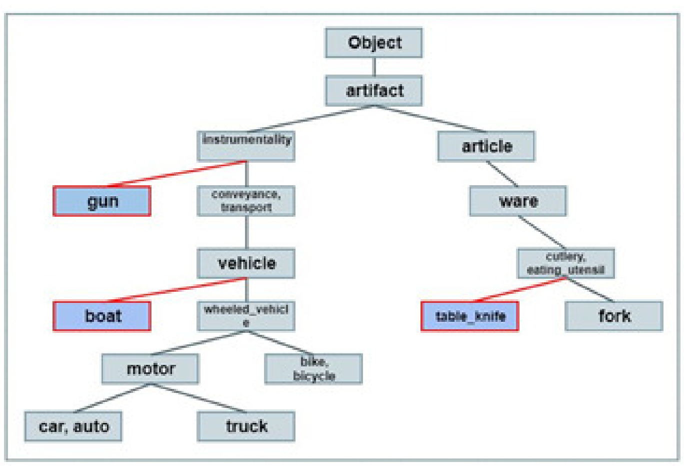

Fig. 8. An extracted from WordNet and extended with some concepts (O_4).

We obtain the following results :

– aSV(stackCommon,O3)=0.5;

- aSV(stackCommon,O4)=0.5;
- aSV(stackDif $f_{(O3\backslash O4)}$,O3) = 0 ;
- aSV(stackDif $f_{(O4\backslash O3)}$,O4) = 0.2875 ;
- θ=1;ω=14/17;α=0;β=3/17;
- TNgom(O3,O4) = 0.95

Example 2 :

In this example, we illustrate the proposition by assessing the similarity between the ontology of the figure 7 and that of the figure 9. We obtain the following results :

- aSV(stackCommon,O3)=0.75;
- aS V(stackCommon, O5) = 0.72 ;
- aSV(stackDif $f_{(O3\backslash O5)}$,O3) = 0.5 ;
- aSV(stackDif $f_{(O5\backslash O3)}$,O5) = 0.6 ;
- θ=6/14;ω=6/13;α=6/14;β=6/13;
- TNgom(O3,O5) = 0.57

In the Experimentation section (section 5), we have given three examples for illustrating our proposition. This section is about analysis of the obtained results.

6. ANALYSIS

Example 1 : Ontologies *O3*and *O4*present a good similarity value *(TNgom(O3,O4) = 0.95)*. Then, we can say that the similarity value between two ontologies is good. We note that ontologies have respectively 14 concepts for *O3* and 17 concepts for *O4.* The ontology O4 contains all O3's concepts and 3 more concepts.

Example 2 : The similarity value between ontologies O3 and O5 is equal to 0.57 *(TNgom(O3,O5) = 0.57).* This similarity value is medium. The ontologies have respectively 14 concepts for O3 and 13 for O5. O3 and O5 each has 6 concepts in their sets of different concepts.

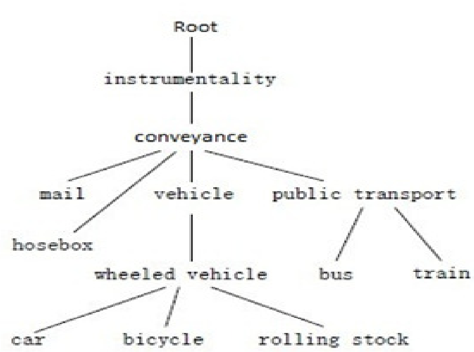

Fig. 9. Representation of an ontology extracted from WordNet (*O5*).

7. CONCLUSIONS

In this paper, we proposed a method for assessing similarity between concepts of two ontologies(modeled with the same language). The approach that we adopt is based on set theory, edges based semantic similarity [2] and feature-based similarity [3]. It can be summarized in 3 steps. In the step 1, we determined the sets of concepts, which characterizes the concepts shared by the two ontologies and the sets of concepts that are different from them. In the step 2, we evaluated the aver- age of the semantic similarity values between concepts of each set that we determined in step 1. We used Wu and Palmer [25] semantic similarity, which is an edge- based semantic similarity measure to compute similarity between concepts of the sets in an ontology, before assessing the average value of similarity for each set. Finally, in step 3, we adjusted the Tversky measure to evaluate the similarity between concepts of the considered ontologies.

The method we propose gives satisfactory results. Indeed, it allows to assess the similarity between two ontologies while taking into account the semantic links that exist between the concepts in ontologies. However, it would be interesting to take into account properties of concepts for extending the formed sets (set of resemblance and set of difference). In future work, we will focus on how we could take into account the properties of concepts and relations between to achieve the main goal of this study, which is the modeling of a method for evaluating similarity and differencebetween the formal ontologies.

ACKNOWLEDGEMENTS

The authors would like to thank the CEA-MITIC, which funded our participation to the 4th Annual Conference on Computational Science and Computational Intelligence, from 14th-16th December 2017 in Las vegas, Navava, USA. Thank to the reviewers of the Conference who help us to improve our method.

REFERENCES

[1] Marc, Ehrig, (2007) Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond, isbn-10= 978-0387328058,isbn-13= 978-0387328058, Springer.

[2] Aly Ngoné, Ngom, (2015)''Etude des mesures de similarité sémantique basée sur les arcs'', CORIA, Paris,France, pp535 - 544.

[3] Amos, Tversky, (1977)''Features of similarity'',Psychological Review, Vol. 84, No.4, pp327-352.

[4] Thomas, R. Gruber, (1993) ''Atranslation approach to portableontology specifications, KnowledgeAcquisition journal,Vol. 5, No.2, pp199-220.

[5] Thomas R., Gruber, (2005) ''Towardprinciplesforthedesignofontologiesused for knowledge sharing'',International Journal of Human ComputerStudies,Vol. 43, No.5, pp 907-928.

[6] MichaelR., Genesereth & Nils. J. Nilsson, (1987) Logical foundations of artificial intelligence. MorganKaufmann PublishersInc, 95 First Street, Los Altos, CA 94022.

[7] Camara, Gaoussou, (2013) Conception d'un système de veille épidémiologique à base d'ontologies: Application à la schistosomiase au Senegal, PhD thesis, Université Pierre Marie-Curie & Université Gaston Berger, Saint-Louis.

[8] Rudi, Studer, V. Richard, Benjamins, & Dieter, Fensel, (1998)''Knowledge engineering: Principles and methods'', IEEE Transactions on Data and Knowledge Engineering, 25 edition.

[9] Ana, Gomez-Perez & V.Richard, Benjamins, (1999) ''Overview of knowledge sharing and reus components: Ontologies and problem- solving methods,In Proceedings of the IJCAI-99 workshopon Ontologies and problem-Solving Methods (KRR5), Stockholm (Suede), pp111-115.

[10] Naouel, Ouafek.Lesontologiesspatiales:la generation automatique d'ontologies spatiales à partir du modèle conceptuel spatio-temporel mads, Master's Thesis, University of Mentouri, Algeria.

[11] Harispe S.Ranwez S. Janaqi S.&Montmain J., (2015)"Semantic Similarity from Natural Language and Ontology Analysis". Synthesis Lectures on Human LanguageTechnologies, doi: 10.2200/S00639ED1V01Y201504HLT027.

[12] Feng Y. Bagheri, E.Ensan F.& Jovanovic J., (2017) "The state of the art in semantic relatedness: a framework for comparison". Knowledge Engineering Review, doi:10.1017/S0269888917000029.

[13]. Couto, Francisco&Lamurias A., (2018)"Semantic Similarity Definition »,Reference Module in Life Sciences (Encyclopedia of Bioinformatics and Computational Biology), doi:10.1016/B978-0-12-809633-8.20401-9.

[14] Agirre, E., E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, &A. Soroa, (2009) ''A study on similarity and relatedness using distributional and wordnet-based approaches'', In Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL'09, Strousburg, PA, USA.

[15] Aime, X., F. Furst, P. Kuntz, &F.Trichet, (2011) ''Semiosem et proxima : mesures sémiotiques desimilarité et de proximité conceptuelles, In Actes des 22e journées francophonesd'Ingénierie desConnaissances(IC'2011),7ePlateformeAFIA,pp539-554.

[16] A. Maedche &S. Staab , (2002) ''Measuring Similarity between Ontologies'', In: Gomez-Pèrez A., Benjamins V.R. (eds) Knowledge Engineering and Knowledge Management: Ontolo gies and the Semantic Web. EKAW. Lecture Notes in Computer Science, vol 247, No.3, pp251-263, , Springer, Berlin, Heidelberg.

[17] I. V. Levenshtein (1966) ''Binary Codes capable of correcting deletions, insertions, and reversals'', Cybernetics and Control Theory journal, Vol.10, No.8, pp707-710.

[18] James Z. Wang and Farha Ali, (2005) An Efficient Ontology Comparison Tool for Semantic Web Applications, Proceedings of the 2005 IEEE/WIC/ACM InternationalConference on Web Intelligence,pp372-378.

[19] GiuseppePIRRO, (1991) ''A semantic similaritymetriccombiningfeaturesandintrinsic informationcontent'',LanguageandCognitive Processes journal, Vol 7. No.6, pp1-28.

[20] Pierro, G. &Euzenat J.,(2009). ''A feature and information theoretic framework for semantic similarity and relatedness'', Data and Knowledge Engineering journal, Vol.68, No.11, pp1289 - 1308.

[21] Aly Ngoné Ngom, Papa Fary Diallo, Fatou Kamara-Sangaré & Moussa LO, (2016)''A method to validate the insertion of a new concept in an ontology'',SITIS 2016 : The 12th International Conference on Signal Image Technology and Internet Systems, Naples, Italy, pp 275-281.

[22] Aly Ngoné Ngom, Papa Fary Diallo, Fatou Kamara-Sangaré & Moussa LO, (2016) ''Amethod to update anontology: simulation'',International Conference in InformationandKnowledgeEngineering16,LasVega, Nevada, USA, 92 - 96.

[23] Haifa, Zargayouna &Simone, Salotti, (2005) Indexation sémantique de documents XML, PhD Thesis, Paris XI University, France, 112 - 115.

[24] Zhibio, Wu & Martha, Palmer, (1994)''Verbs semantics and lexical selection'', In Proceedings of the 32nd annual meeting on ACL, pp 133 - 138.

[25] Georges, A. Miller &Charles, Walter (1991)‘’Contextual Correlates of Semantic Similarity’’, Language and Cognitive Processes journal,Vol 10, No.6, 1-28.

[26] Christine, Fellbaum (1998).WordNet: An Electronic Lexical Database., The MIT Press.

AUTHORS

Alyngoné NGOM is Java JEE Engineer at Senegalese Customs since 2016. Since July 2018, hereceived his Ph.D. degree inComputer Science (Ontologies Evolution, Knowledge Engineering)Gaston Berger University of Senegal. His current research is focused on ontology evolution and intégration.

Guidedi Kaladzavi is a Senior Lecturer andresearcher in computer engineering at the University of Maroua. He received his Ph.D. degree in computer science in 2016 from University of Maroua,Cameroon. Since 2017 he has been postdoctoral researcher in WIMMICS team, INRIA Sophia Antipolis in the MoReWAIS (Mobile Read Write Access and Intermittent to Semantic Web) project under the Co-supervision of Moussa LO and Fabien Gandon. His current research is focused onSemanticWeb, Ontology Management, Mobile Web of Data.

Fatou Kamara-Sangaré is a Senior Lecturer and researcher in computer engineering at the University of Gaston Berger, Senegal. She received his Ph.D. degree in computer science in 2019. His current research is focused on Ontology Evolution and Information Retrieval.

Moussa Lo is a Professor of Computer Science at the University of Gaston Berger, Senegal and Head of the IT team at the Laboratory of Numerical Analysis and Computing (LANI), he was a leader of UFRSAT, then Coordinator of CEAMITIC. Since April 1999, Professor Moussa has been conducting research activities in the 2SIWeb2 group (Information System for the Web) of the University of Pau and Pays de l'amour, under the direction of Mr. Amrane HOCINE. Current- ly, he is the Coordinator of the Virtual University of Senegal. His current research is focused on semantic Web, applications Integration, mobile Web, Web of Data.

Kolyangis a Professor of Computer Science at the University of Maroua, Cameroon. He received his PhD at the University of Bremen, Germany in 1997 on Formal Methods in Software Engineering under the supervision of Prof. Dr. KriegBrückner. In 2008 he got a Habilitation The- sis at the University of Bremen. In 1999 he became the forefounder of Computer Science Studies at the University of Ngaoundere, Cameroon. Since 2015 he has been nominated Research Ambassador of the University of Bremen, Germany. Since 2015, he is also Expert of CAMES for distance and open learning, for the higher education systems in Africa. Currently, his research domains focus on Software Engineering, Learning, Digital Economy and ICT4D.