An Ontology Alignment Hybrid Method Based on Decision Rules

Mohamed Biniz[1] and Mohamed Fakir[2]
[1]Department of Mathematics and Computer Science, Sultan Moulay Slimane University, Morocco
[2]Department of Informatics, Sultan Moulay Slimane University, Morocco

Abstract: *In this paper, we propose a hybrid approach based on the extraction of decision rules to refine the alignment results due to the use of three alignment strategies. This approach contains two phases: training phase which uses structural similarity, element similarity, instance-based similarity and C4.5 algorithms to extract decision rules, and evaluation phase which refines discovered alignment by three alignment strategies using extracted decision rules. This approach is compared with the best systems according to benchmark OAEI 2016: Framework for Ontology Alignment and Mapping (FOAM), A Dynamic Multistrategy Ontology Alignment Framework (RIMOM), AgreementMakerLight and Yet Another Matcher-Biomedical Ontologies (YAM-BIO), the proposed method gives good results (good recall, precision and F-measure). Experimental results show that the proposed approach is effective.*

Keywords: *Decision rules, alignment, ontology, similarity, similarity flooding.*

Received August 24, 2016; accepted May 6, 2018

1. Introduction

In the field of the semantic Web, ontology intended to be a representation of the real world that can be accepted by all members of a community. The possibilities of representation vary, and therefore ontologies differ despite standardization. These different conceptualizations generate heterogeneous ontologies. These heterogeneity limits the possibilities of interoperability between them. These ontologies have to be linked together to make possible integration tasks, sharing of information, research information from several sources, etc.

Currently, several types of solutions have been proposed for solving the heterogeneity between ontologies in a discipline called alignment of ontologies. Researchers have developed many types of systems as Partial Network Alignment (PNA) with generic stable matching [23], Ontology Alignment using Web-text as Interlingua (PIDGIN) [20], interactive ontology matching based on Markov networks (iMATCH) [2], Yet Another Matcher–Light (LYAM)++ [19], Optimizing Ontology Alignments by using NSGA-II [22], etc., These integrated approaches use multi-strategies, for example: weighted average similarity and combined methods. These give good results, but still have many shortcomings, such as the good threshold or sufficient similarity. To solve these problems, this paper presents a method to solve the alignment of ontology. This method uses Similarity Flooding [14], Wu and Palmer algorithm [21], Jaccard Similarity [18] and instances-based technic. Through the application of these techniques, we get a table of similarity that contains linguistics, syntactic, structural and elemental similarity. To discover alignment between ontologies we use the classification algorithm C4.5 [1].

This paper is organized as follows: section 2 defines ontology alignment problems and gives a brief description of the work done in this area. Section 3 describes our approach; it explains our methodology, the experimental setup and discusses experimental results. Section 4 gives a conclusion of the work.

2. Literature Survey

2.1. Ontology Alignment Problem

The alignment of ontologies is a field that determines the correspondence of a pair of ontologies O1 and O2. Therefore, giving a pair of ontologies, so the corresponding task is to find an alignment between these ontologies. According to Euzenat definition [9], there are other parameters that can extend the definition of alignment (see Figure 1), namely:

- The use of an alignment input A, which must be expanded.
- The corresponding parameters, like weight, or thresholds.
- External resources, such as the common knowledge and the specific thesaurus field.

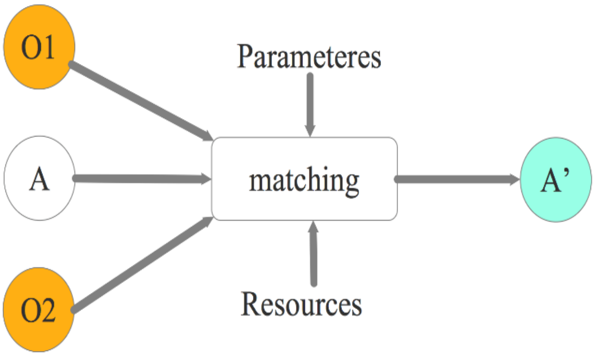


Figure 1. The ontology alignment process [9].

Given two ontologies, Euzenat has set the alignment by 4-tuple: <id, E_1, E_2, R>, where:

- id is an identifier for the data correspondence.
- E_1 and E_2 are entities, for example, the classes and the properties of the first and second ontology, respectively.
- R is a relationship between E_1 and E_2, for example, equivalent, more general, disjunct.
- Alignments can be of different cardinalities: one-to-one, one-to-many, many-to-one or many-to-many.

2.2. Existing Matching Tools

In recent years, several systems and tools have appeared [15], we selected a few that attended in Ontology Alignment Evaluation Initiative (OAEI) in order to have a basis for comparison. Below we discuss these systems in more detailed way.

2.2.1. Agreement Maker Light

Agreement Maker Light [10, 11] is an automatic system, it has a sophisticated user interface, and provide a set of evaluation strategies. It was designed to handle large scale ontologies. The system manages ontologies in Extensible Markup Language (XML), Resource Description Framework Schema (RDF) Schema, and Web Ontology Language (OWL). It is structured into two modules: calculation of similarity and alignment selection. The system combines matchers using three layers:

- The matchers of the first layer comparing design features, such as labels, comments and instances, which are represented as Term Frequency–Inverse Documents Frequency[17] (TF-IDF) vectors used with a cosine similarity metric and other string-matchers.
- The second layer uses the structural properties of ontology. It includes two matchers : Descendant Similarity Inheritance (DSI) and the Sibling Similarity Contribution (SSC) [7] matchers.
- In the third layer, a weighted linear combination is calculated on the basis of the results of the first two layers, whose results are filtered based on thresholds.

2.2.2. Yet Another Matcher-Biomedical Ontologies (YAM-BIO)

YAM-BIO [3] is an ontology alignment system, flexible and self-configurable for discovering semantic correspondences between entities (classes, objects properties and data properties) of ontologies. It includes several algorithms for calculating similarity. The system consists of four essential layers:

- Layer 1 called element level matching.
- Layer 2 called structure level matching.
- Layer 3: Post-filter component.
- Layer 4 is the semantic verification component.

2.2.3. Dynamic Multistrategy Ontology Alignment Framework (RIMOM)

RIMOM [24] is a dynamic system of multi-strategy ontology alignment. It focuses on the combination of multiple matching strategies by risk minimization of Bayesian decision. The system quantitatively estimates the characteristics of similarity for each matching task. These characteristics enable dynamically selecting and combining the multiple matching methods.

Two matching methods are used: linguistic similarity (Edit distance) and structural similarity (Similarity Flooding). If the two ontologies have a strong similarity structure factor, the system uses a similarity propagation process to refine the found alignments and to find new alignments that cannot be found using other strategies.

2.2.4. Framework for Ontology Alignment and Mapping (FOAM)

FOAM [8] is a system that fully or semi-automatically align two or more OWL Ontologies, based on heuristics (similarity) of the individual entities (concepts, relations, and instances). it includes six essential steps:

- Feature Engineering.
- Search Step Selection.
- Similarity Assessment.
- Similarity Aggregation.
- Interpretation.
- Iteration.

2.2.5. Ontology Matching Using BabelNet Dictionary and Word Sense Disambiguation Algorithms

Ontology Matching Using BabelNet Dictionary and Word Sense Disambiguation Algorithms (OMBWSD)[6] approach combines different techniques from the field of automatic processing of natural language (Adapted Lesk algorithm, the Wu and Palmer algorithm, Resnik algorithm Leacock and Chodorow algorithm) and BabelNet as reference ontology, to design an efficient similarity measure for comparing ontology entities.

3. Our Approach

In this paper, we formulate the ontology matching problem as a classification problem. We use multiple matchers to calculate several similarities between the entity pair and use those similarities as the features candidate alignment. We use the C4.5 algorithms to extract decision rules to classify candidate alignments to correct and incorrect alignments. This approach is described as follows.

3.1. Similarity Measures

The aim of the Ontology is to find corresponding entities from the input ontologies. In this section, we present the similarity measures used in our system.

3.1.1. Jaccard Similarity

Given two Class *C1* and *C2*. The Jaccard similarity [18] is given by:

$$jacc(C1, C2) = \left|\frac{C1 \cap C2}{C1 \cup C2}\right| \qquad (1)$$

3.1.2. Levenshtein Distance Similarity

The Levenshtein distance [5] between two strings is the minimum cost of edit operations which need to transform one string into another. Each operation has a cost function associated, in the simplest form, each has cost 1. The Levenshtein distance can address the typographical errors of name variations very well, and the distance can be transformed to similarity by subtracting normalized distance by 1:

$$Levein(c1, c2) = 1 - \frac{lev(c1, c2)}{\max\{|c1|, |c1|\}} \qquad (2)$$

Where:

- *c1* and *c2*: are the two compared values.
- lev (*c1*, *c2*): the distance (cost) between *c1* and *c2.*
- |c1| and |*c2*|: the length of c1 and *c2* respectively.

3.1.3. Instance-based Similarity

For instance-based similarity, we use the Vector Space Model (VSM)[13], it is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms.

Documents and queries are represented as vectors:

$d_j= \{w_{j1}, w_{j2}, w_{j3}, w_{j4}, \ldots, w_{jn}\}$.
$d_q= \{w_{q1}, w_{q2}, w_{q3}, w_{q4}, \ldots, w_{qn}\}$.

Where:

- w_{ji} the weight of the term in the position i and document j.

In the classical vector space model, keywords in a document are assigned weights reflecting that some words are better at discriminating between documents than others. Similarly, in our approach, annotations have a weight that reflects the relevance of the instance is considered the meaning document. The weights are calculated automatically by an adaptation of the TF-IDF algorithm [10], based on the frequency of occurrence of instances in each document. Specifically, the weight $w_{j,i}$ of instance I_i for d_j document is calculated as follows:

$$W_{j,i} = \frac{Occur_{j,i}}{max_k Occur_{j,k}} \times log\frac{N}{n_i} \qquad (3)$$

Where:

- $Occur_{j,i}$ is the number of occurrences of I_i in d_j
- $max_k\ Occur_{j,k}$ is the frequency of the most repeated instance in d_j.
- n_i is the number of documents annotated with I_i.
- *N* is the total number of documents in the search space.

After modeling documents using vector space model and TF-IDF algorithms, the similarity is calculated by the Cosine algorithm using the following Equation:

$$VSM(d_j, d_q) = \frac{d_j \times d_q}{|d_j| \times |d_q|} = \frac{\sum_i w_{j,i} \times w_{q,i}}{\sqrt{\sum_i w_{j,i}^2} \times \sqrt{\sum_i w_{q,i}^2}} \qquad (4)$$

3.1.4. Wu and Palmer’s Similarity

The similarity metric of Wu and Palmer [21] measure the depth of two concepts in the taxonomy given WordNet, and the depth of their lowest common ancestor (Lowest Common Subsume (LCS)) and combines them to create a similarity score:

$$WUP(C1, C2) = \frac{2 * depth\ (LCS)}{depth(C1) + depth\ (C2)} \qquad (5)$$

3.1.5. Adapted Lesk Similarity

Banerjee [4] proposed an improved Lesk algorithm, called “Adapted Lesk”, based on two axes. The first axis is the incorporation of the definitions of sense connected by WordNet taxonomic relationships in the definition of a given sense and the second is a new way to calculate the overlap between the definitions of words. To calculate the overlap between two senses, they propose to consider not only the overlap between the definitions of the two senses but also the definitions of relations R hyperonyms (has-kind, hyponyms (kind-of) metonym (part-of) homonyms (has-hand), but also by troponymes attribute relations, similar-to, also-see. To ensure that the measure is symmetric, they propose to group the recovery assessments between the definitions of pairs relationships $\mathfrak{R}$.

Let Ψ be the series of connections to calculate the recovery. A set is defined by:

$$\mathfrak{R} = \{(R1, R2) | \forall (R1, R2) \epsilon \Psi^2, (R1, R2) \epsilon \mathfrak{R}^2 \Longrightarrow (R1, R2) \epsilon \mathfrak{R}^2\} \qquad (6)$$

Therefore, the score is calculated as the sum of overlap between the definitions of pairs of relationships:

$$AdLesk(c1, c2) = \sum_{\forall(R1,R2)\epsilon\Re^2}(|definition(R1\ (c1)) \cap definition(R1\ (c1))|)^2 \qquad (7)$$

3.1.6. Similarity Flooding

The Similarity Flooding [14] Algorithm takes two graphs as input, and produces as output a mapping between corresponding nodes of the graphs.

- As a first step, we transform ontologies in a graph G in which the vertices are pairs of ontology concepts and edges exist between two nodes, if there is a relationship in both ontologies between the nodes of the two pairs. In fact, the original similarity flooding algorithm only connects the concepts whose edges have the same label.
- As a second step, we assign weight w to the edges, which are typically 1/n, where n is the number of outgoing edges.
- As a third step, we assign initial similarityσ^0 to each node.
- As a fourth step, we compute σ^{i+1} for each node with the following Equation:

$$\sigma^{i+1}(x, x') = \sigma^i(x, x') + \sum_{(\langle y,y'\rangle,p,\langle x,x'\rangle)} \sigma^i(y, y') \times w(\langle y, y'\rangle, p, \langle x, x'\rangle) \qquad (8)$$

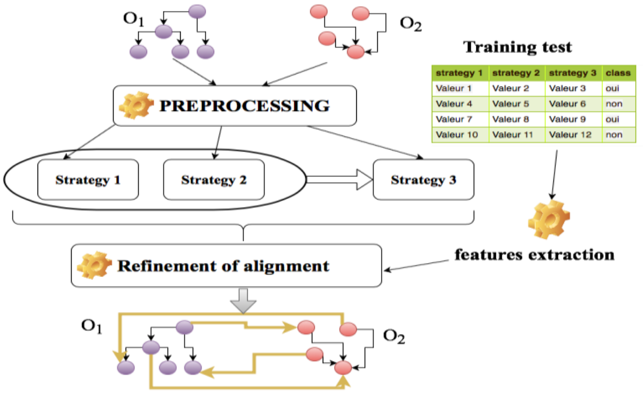
Where:

- $\sigma^{i+1}(x,\ x')$ is the similarity value between two entities $(x,\ x')$ in the iteration *i+1.*
- $\sigma^i(x,\ x')$ is the similarity value between two entities $(x,\ x')$ in the iteration *i.*
- $\sigma^i(x,\ x')$ is the similarity value between two entities $(y,\ y')$ in the iteration *i.*
- $w((y, y'),\ (x,\ x'))$ is the weight of the outgoing arc from entities(y, y') to entities in $(x,\ x')$.

- As a fifth step, we normalize all $\sigma^{i+1}$by dividing by the largest value.
- As a sixth step, if no similarity changes more than particular threshold ϵ, or after prefixed number of steps, stop otherwise go to the fourth step.

3.2. Architecture of Our Approach

The proposed methodology includes two phases essential as shown in Figure 2. First phase, called training phase, for the extraction of features from a training test. The second phase, called evaluation phase, aims to find the corresponding pair between two given ontologies.



strategy 1	strategy 2	strategy 3	class
Valeur 1	Valeur 2	Valeur 3	oui
Valeur 4	Valeur 5	Valeur 6	non
Valeur 7	Valeur 8	Valeur 9	oui
Valeur 10	Valeur 11	Valeur 12	non

Figure 2. The framework of the proposed approach.

3.2.1. Training Phase

After classification of the two entities to be aligned or not from OAEI reference, all validated alignment pairs are treated by three strategies for calculating similarity. Each strategy (explained below) combines similarity function that returns a numeric value that is registered with the pair of the features in the training test. We use C4.5 algorithm as classifier which distinguishes between those entities which align and those which are disjoint.

3.2.2. Evaluation Phase

This phase consists of three steps essential:

a) Pre-processing.
b) Determination of strategies.
c) Alignment refinement.

a) *Pre-processing*:
This step contains two sub-steps. The first extract classes with these annotations (name, labels, comments and instances) and properties with these annotations (range, domain, name, labels and comments) for each ontology. The second includes natural language processing technic such as: tokenization, stop words removing and words stemming.
b) Determination of strategies
This step contains three strategies:

- **Strategy 1**: consists of calculating the similarity by the Equation:

$$Sim_1(c1, c2) = max \begin{cases} jacc(c1_n, c2_n) \\ jacc(c1_l, c2_l) \\ jacc(c1_c, c2_c) \\ VSM(c1, c2) \\ WUP(c1_n, c2_n) \end{cases} \qquad (9)$$

Where:

- jacc($c1_n$,$c2_n$),jack($c1_I$, $c2_I$), and jacc($c1_c$,$c2_c$) are the Jaccard similarity between names, labels and comments respectively.
- VSM(*c1*,*c2*) is the instance-based similarity between two classes.

- *WUP*(*c1ₙ*,*c2ₙ*) is the similarity measure using Wu and Palmer algorithm between two labels of two classes.
- **Strategy 2**: consists of calculating the similarity by the Equation:

$$Sim_2(c1,c2) = max\begin{cases} Levein(c1_n,c2_n) \\ Levein(c1_l,c2_l) \\ Levein(c1_c,c2_c) \\ VSM(c1,c2) \\ AdLesk(c1_n,c2_n) \end{cases} \quad (10)$$

Where:

- Levein($c1_n$,$c2_n$), Levein($c1_l$,$c2_l$), and Levein($c1_c$,$c2_c$) are the Levenshtein similarity between names, labels and comments respectively.
- VSM(*c1*,*c2*)is the instance-based similarity between two classes.
- Adlesk($c1_n$, $c2_n$) is the similarity measure using Adapted Lesk algorithm between two labels of two classes.
- **Strategy 3** uses the similarity flooding to compute the similarity between two classes knowing that the initial matrix of pairwise is calculated from the strategies 1 and 2 by the Equation:

$$\sigma^0(c1,c2) = \max\begin{cases} Sim_1(c1,c2) \\ Sim_2(c1,c2) \end{cases} \quad (11)$$

c) Alignment refinement

This step focuses on refining the mapping generated using decision rules that are extracted from the training phase.

3.3. Experimental Setup and Results

The individual results of benchmark ontologies OAEI 2016 [6] were grouped in five groups below:

- Tests 101 to 104
- Tests 201 to 210
- Tests 221 to 247
- Tests 248 to 266
- Tests 301 to 304

We evaluated the performance and effectiveness of our approach on OAEI 2016 benchmark test.

To measure the degree of alignment precision, three traditional measures borrowed from information retrieval, namely precision, recall and F-measure [16] were used. These three measures are appropriate for the evaluation of Ontology. they are defined in Equations (12) and (13):

$$precision = \frac{Nc}{Nt} \quad (12)$$

$$recall = \frac{Nc}{Nr} \quad (13)$$

Where:

- *Nc*: number of correctly found correspondences.
- *Nt*: numbers of all found correspondences.
- *Nr*: number of all reference alignment

F-measure represents a compromise between precision and recall and it is calculated by Equation (14).

$$F - measure = \frac{2*precision*recall}{precision+recall} \quad (14)$$

The quality of the alignment of the output was judged by regarding the reference alignment that is available early in the process, and the three parameters precision, recall and F-measure.

Figure 3 compares the results of our approach with the most effective systems according Ontology Alignment Evaluation Initiative (OAEI). AML is dependent on entity names of ontology and is sensitive data. When the number of entities of the ontology is very large, AML and OMBWSD works well, but when entities are limited, the result is good. Compared to it, our approach uses all the information about the ontology to find the correspondence pairs. It does not depend only on one aspect of ontology, so it has a better result. Relative to FOAM, recall and precision are both slightly higher because our method uses decision rules for the results of the mapping discovery. This means that the result using the combination similarities with C4.5 is satisfactory.

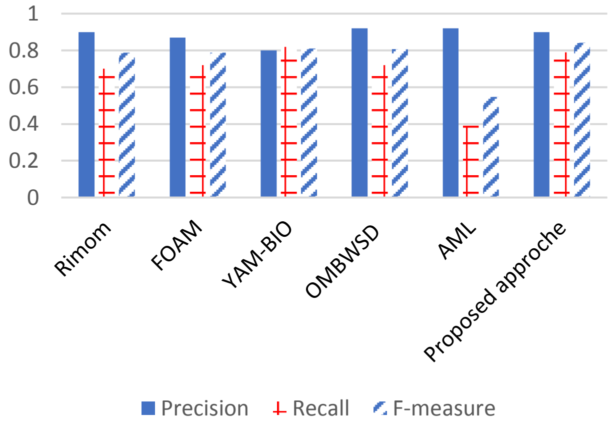


Figure 3. Comparison of the proposed approach and other systems.

RIMOM and YAM-BIO use multi-strategy to compute the similarity of the elements of ontology and combine the results of similarity mapping during the process of discovery. Since RIMOM and YAM-BIO only provide the means of precision and recall for all tests without detail for each test, we can say that our approach gives good results.

4. Conclusions

This paper proposes a hybrid method for Ontology alignment, which achieves multi-strategy mapping and refines the similarity results provided by all these strategies by the use of decision rules, the method has two phases:

1. Training phase.
2. Evaluation phase.

To evaluate the performance of the proposed approach, it was tested with ontologies of Benchmark tests OAEI 2016. the Experiments show that our method is not only effective, but also applicable to process large-scale ontology mapping tasks such as OMBWSD, AML and RIMOM ontologies matching systems. Finally, in our future work we extend the proposed approach to support medical ontologies through the use of Unified Medical Language System (UMLS) [12] dictionary and descriptive logic.

References

[1] Agrawal G. and Gupta H., "Optimization of C4. 5 Decision Tree Algorithm for Data Mining Application," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 3, pp. 341-345, 2013.

[2] Albagli S., Eliyahu-Zohary R., and Shimony S., "Markov Network Based Ontology Matching," *Journal of Computer and System Sciences*, vol. 78, no. 1, pp. 105-118, 2012.

[3] Annane A., Bellahsene Z., Azouaou F., and Jonquet., "YAM-BIO: Results for OAEI 2017," *in Proceedings of International Standard Musical Work Code*, Austria, pp. 1-7, 2017.

[4] Banerjee S., "Adapting the Lesk algorithm for word sense disambiguation to WordNet," University of Minnesota Duluth, 2002.

[5] Biniz M. and El Ayachi R., "Optimizing Ontology Alignments by Using Neural NSGA-II," *Journal of Electronic Commerce in Organizations*, vol. 16, no. 1, pp. 29-42, 2018.

[6] Biniz M., El Ayachi R., and Fakir M., "Ontology Matching Using BabelNet Dictionary and Word Sense Disambiguation Algorithms," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 1, pp. 196-205, 2017.

[7] Cruz I., Antonelli F., and Stroe C., "Using AgreementMaker to Align Ontologies for OAEI 2009: Overview, Results, and Outlook," *in Proceedings of the ISWC 2009 Workshop on Ontology Matchin*g, USA, pp. 135-146, 2009.

[8] Ehrig M. and Sure Y., "Foam-Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative," *in Proceedings of Integrating Ontologies Workshop Proceedings*, Banff, pp. 72-76, 2005.

[9] Euzenat J. and Shvaiko P., *Ontology Matching*, Springer Science and Business Media, 2013.

[10] Faria D., Pesquita C., Balasubramani B., Martins C., Cardoso J., Curado H., Couto F., and Cruz I., "OAEI 2016 results of AML," *in Proceedings of 11th International Workshop on Ontology Matching co-located*, pp. 138-145, 2016.

[11] Faria D., Pesquita C., Santos E., Palmonari M., Cruz I., and Couto F., "The Agreementmakerlight Ontology Matching System," *in Proceedings of the Move to Meaningful Internet Systems: OTM Conferences*, Berlin, pp. 527-541, 2013.

[12] Liu Y., McInnes B., Pedersen T., Melton-Meaux G., and Pakhomov S., "Semantic Relatedness Study Using Second Order Co-Occurrence Vectors Computed From Biomedical Corpora, UMLS and WordNet," *in Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, Miami, pp. 363-372, 2012.

[13] Manwar A., Mahalle H., Chinchkhede K., and Chavan V., "A Vector Space Model for Information Retrieval: A MATLAB Approach," *Indian Journal of Computer Science and Engineering*, vol. 3, no. 2, pp. 222-229, 2012.

[14] Melnik S., Garcia-Molina H., and Rahm E., "Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching," *in Proceedings of 18th International Conference on Data Engineering*, San Jose, pp. 117-128, 2002.

[15] Otero-Cerdeira L., Rodríguez-Martínez F., and Gómez-Rodríguez A., "Ontology Matching: A Literature Review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949-971, 2015.

[16] Powers D., "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37-63, 2011.

[17] Ramos J., "Using Tf-Idf to Determine Word Relevance in Document Queries," *in Proceedings of the 1st Instructional Conference On Machine Learning*, 2003.

[18] Singthongchai J., Niwattanakul S., Naenudorn E., and Wanapu S., "Using of Jaccard Coefficient for Keywords Similarity," *in Proceedings of presented at the International MultiConference of Engineers and Computer Scientists*, Hong Kong, pp. 380-384, 2013.

[19] Tigrine A., Bellahsene Z., and Todorov K., "Light-Weight Cross-Lingual Ontology Matching with LYAM++," *in Proceedings on the Move to Meaningful Internet Systems: OTM Conferences*, Cham, pp. 527-544, 2015.

[20] Wijaya D., Talukdar P., and Mitchell T., "Pidgin: Ontology Alignment Using Web Text As Interlingua," *in Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management*, San Francisco, pp. 589-598, 2013.

[21] Wu Z. and Palmer M., "Verbs Semantics and Lexical Selection," *in Proceedings of the 32nd Annual Meeting on Association for*

Computational Linguistics, Las Cruces, pp. 133-138, 1994.

[22] Xue X., Wang Y., and Hao W., "Optimizing Ontology Alignments by using NSGA-II," *The International Arab Journal of Information Technology*, vol. 12, no. 2, pp. 176-182, 2015.

[23] Zhang J., Shao W., Wang S., Kong X., and Yu P., "PNA: Partial Network Alignment with Generic Stable Matching," *in Proceedings of IEEE International Conference on Information Reuse and Integration*, San Francisco, pp. 166-173, 2015.

[24] Zhang Y., Jin H., Pan L., and Li J., "RiMOM results for OAEI 2016," *in ISWC*, Japan, pp. 210-216, 2016.

Mohamed Biniz received his master's degree in business intelligence in 2014 and PhD's degree in computer science in 2018 from the Faculty of Science and Technology, University Sultan Moulay Sliman Beni Mellal. He is currently a professor at Polydisciplinary Faculty of Beni Mellal. His research activities are located in the area of the semantic web engineering specifically, it deals with the research question of the evolution of ontology, big data, natural language processing, dynamic programming, etc.

Mohamed Fakir obtained a degree in Master of Electrical Engineering from Nagaoka University of Technology in 1991 and a PhD degree in electrical engineering from the University of Cadi Ayyad, Morocco. He was a team member in Hitachi Ltd., Japan between 1991 and 1994. He is currently a professor at the Faculty of Science and Technology, University Sultan Moulay Slimane, Morocco. His research interest includes image processing, pattern recognition, semantic web, big data and artificial intelligence.