

# Presenting a Method for Aligning Ontologies Based on Structural Graphs

Nazar Mohammad Parsa

Department of Computer Science, Education Faculty, Faryab, AFGHANISTAN

Corresponding Author: Email ID ?



[www.sjmars.com](http://www.sjmars.com) || Vol. 4 No. 5 (2025): October Issue

Date of Submission: 11-10-2025

Date of Acceptance: 26-10-2025

Date of Publication: 30-10-2025

## ABSTRACT

The need to give meaning to web content in a way that is understandable to different machines is the main idea behind the emergence of the Semantic Web, which is based on ontologies, as structures that model data in the form of words and concepts. In this paper, we present a method for aligning existing ontologies based on recognizing lexical, semantic, and structural similarities between entities in these ontologies. To identify mappings related to two ontologies automatically, the first and most important step is to identify anchors. All alignment calculations are performed based on the initial anchors. In the proposed method, anchors are identified by lexical and semantic analysis of entities. In the lexical matching step, entity labels are compared based on five criteria. To integrate the effect of the lexical similarity criteria set, averaging of normal values is used. For semantic matching of two ontologies, the Verdant tool is also used. In the structural matching section of ontologies, the similarity between entities is recalculated based on the ontology graph and the similarity of the neighborhoods of the entities. In addition, the feature matching operation is also considered. In the last step, the resulting mapping set is filtered and a 1-1 mapping is presented as the final alignment set. To compare and evaluate the proposed method, the OAEI dataset is used and the results of this method are compared with other proposed systems using the evaluation criteria, precision, recall, and F-criterion. The evaluations performed show the efficiency of the proposed system compared to other systems under consideration.

**Keywords-** Semantic Web, ontologies, alignment, lexical, semantic, structural and graph.

## I. INTRODUCTION

The Semantic Web is based on ontologies, as structures that model data in the form of words and concepts. Ontologies are promising formal representations of knowledge that have been widely used in recent years in various fields, including computer science, medical systems, and knowledge management [1-2]. In their article, they defined the Semantic Web as pursuing a vision of the Web in which increased access to structured content enables higher levels of automation.

The initial work done in the field of ontology alignment often operates semi-automatically (using a scale proposed by experts), in such a way that at each stage of the alignment, a number of matching candidates are suggested to the user by the alignment method and the user selects a number of them from the selected candidates. The system introduces new candidates according to the user's selection in the next stage and the work continues iteratively, in which case the final alignment is achieved in human interaction with the system. In some methods, at the beginning of the work, the user introduces a number of matching options to the system and the system performs the alignment operation based on the user's initial information. However, given the increasing growth of information available on the web and the growing number of ontologies, using the user's knowledge in ontology alignment may not always be possible. Considering this problem, new methods try to perform the alignment process as automatically and without human intervention as possible. Of course, so far, methods that have been designed completely automatically have not been able to produce results that are as good as those produced by semi-automatic methods, but the knowledge available in this field is getting closer to this goal day by day. [3 - 4]. Much work has been done in the field of ontologies, each of which is used in specific fields. Ontologies continue to

play a central and key role in the interaction and coordination of software agents and applications in the semantic web [5]. With the increasing expansion of the semantic web, the need for ontologies to create these interactions has increased. Since the introduction of the Semantic Web by Tim Berners-Lee and the use of ontologies in it, much research has been done in the field of ontology matching, as one of the most important issues in the Semantic Web. Given the increasing growth of information available on the Web and the growing number of ontologies, using user knowledge in ontology matching may not always be possible. Considering this problem, new methods try to perform the matching operation as automatically as possible without human intervention. Of course, so far, methods that have been designed to be fully automated have not been able to produce results as good as those produced by semi-automated methods, but the knowledge available in this field is getting closer to this goal day by day [6-7].

## II. METHODOLOGY

Ontology alignment is performed according to a strategy or combination of methods to calculate similarity measures. There are different methods to calculate similarity between different entities.

### 2.1- Terminological methods

These methods are based on the comparison of terms of strings or texts. They are used to calculate the amount of similarity between text units, such as names, labels, descriptions, etc.

### 2.2- Structural methods

These methods calculate the similarity between two entities by utilizing structural information, when the relevant entities are connected to others by semantic or syntactic links, forming a hierarchy or graph of entities. Which are in the following two forms:

- Internal structural methods: methods that exploit only information related to the attributes of the entity,
- External structural methods: methods that consider the relationships between entities.

### 2.3- Extensive/Structural Methods

These methods infer the similarity between two entities, especially concepts or classes, by analyzing their extensions, i.e. their instances.

### 2.4- Semantic Methods

Methods based on external ontologies: When two ontologies need to be aligned, it is preferable that the comparisons are made based on a common knowledge. Therefore, these methods use an intermediate formal ontology to meet this need. This ontology will define a common ground for the alignment of the two ontologies [3].

## III. PRIOR APPROACH

GAOM [8]. It is one of the first ontology matching systems that uses an evolutionary algorithm to optimize the mapping, which creates a discrete optimal model for the ontology matching problem. A genetic algorithm-based optimization method is presented for the ontology matching problem as a feature matching process. First, from a global perspective, it models the ontology matching problem as a mapping optimization problem between two compared ontologies, and each ontology has its own set of associated features.

YAM++ [9]. It is a flexible and extensible ontology matching system. YAM++ leverages many methods from different fields such as machine learning, information retrieval, graph matching, etc. to improve the matching quality.

LogMap [10]. A scalable ontology matching system with built-in reasoning and recognition capabilities, LogMap is the only matching system that can deal with semantically rich ontologies containing tens (and even hundreds) of thousands of classes. Unlike most existing tools, LogMap also implements algorithms to detect and repair “on the fly” inconsistencies. Furthermore, LogMap can produce “clean” sets of output mappings in many cases, meaning that the ontology obtained by merging LogMap output mappings with the input ontologies is consistent.

KEPLER [11]. An ontology matching system that addresses key challenges related to heterogeneous ontologies in the Semantic Web. Kepler builds on previous approaches [12, 13], and relies on several matching strategies. This method is designed to discover matches for large-scale and common-size ontologies as well as for computational matches in multilingual contexts. The main idea of Kepler is to use OWL language expressions to identify and compute the similarity between entities of two given ontologies. ALIN [14] is an interactive ontology matching system, which performs the following steps to perform the interactive ontology matching process: First, it generates an initial set of mappings, called the mapping proposal set, which are the mappings to receive expert feedback. After that, the interactive phase begins, where in each interaction, the expert gives his feedback for some mapping proposals. After each expert feedback, the mapping proposal set is modified according to the expert feedback. The set of mapping suggestions is refined using structural analysis of the ontology, with interactions continuing until no more mapping suggestions remain.

The AML system [15] is a scalable and extensible ontology alignment system with mapping repair capabilities with a strong focus on the use of external knowledge. AML is primarily based on lexical matching algorithms, but includes structural algorithms for both matching and filtering, as well as its own logical repair algorithm. It is capable of using external

background knowledge, and even automatically selecting background knowledge sources for any given ontology to match. It is also worth noting that expert opinion is involved in this method.

#### IV. OUR APPROACH

Aligning ontologies initially involves finding the relationship between their entities, or rather estimating the degree of similarity by calculating similarity measures. In this thesis, we focused on creating a mapping between ontology elements (classes, features, and instances) through three main levels: lexical, semantic, and structural. The architecture of the proposed system consists of several stages, each of which will be discussed separately and in detail below. Figure 1 shows the overall structure of the proposed system. The source and target ontologies constitute the input to our system. After receiving the two ontologies, in the first step, lexical and semantic matching is performed using entity (class) similarity and feature similarity calculations. Using an initial filter based on the lexical similarity threshold, the lexical mapping results are sent to the next stage. In the second step, the diffusion graph is constructed using the input document and the structural similarity between entities is calculated using the diffusion approach and in an iterative manner, and the pairs of entities with similarity exceeding a certain threshold are declared as the output mappings of this step. In the final step, the mappings are filtered and the final mappings are presented.

##### 4-1- Lexical matching stage

To identify mappings related to two ontologies automatically, the first and most important stage is to identify anchors. In fact, all alignment calculations are performed based on the initial anchors. Therefore, accurate identification of anchors will have a significant impact on the accuracy of the alignment method. In the proposed method, anchors are identified by lexical and semantic analysis of entities. Each entity in an ontology has labels related to concepts and data features. In the lexical matching stage, entity labels are compared based on several criteria including the Winkler sweep criterion, overlap coefficient, Lowenstein distance, Jaccard and Trigram. To calculate lexical matching, the similarity obtained from the five criteria used is combined together and the lexical similarity between both entities from the two input ontologies is determined. After that, similar entity pairs are filtered based on the lexical similarity threshold to extract the most relevant lexically aligned entities.

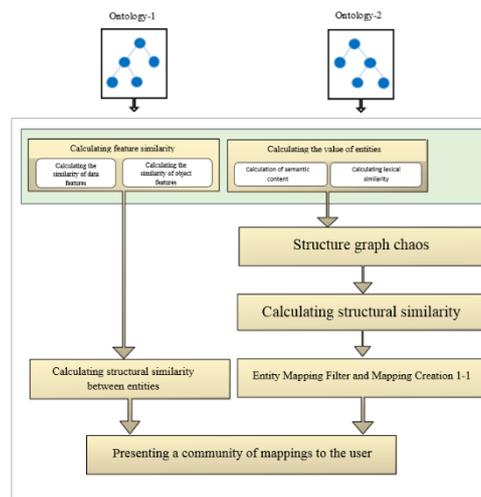


Figure 1: N-level Caching Model

##### 4-1-1. Jaro-Winkler Distance

The Jaro-Winkler distance is introduced to measure the similarity between two strings. In this measure, if there is a shift or misspelling between two strings, the similarity between the two strings is calculated based on the number of common and shifted characters [19]. According to this measure, the greater the distance between two strings, the more similar they are to each other. The Jaro-Winkler distance between two strings is calculated by the following formula:

$$Sim_{Jaro}(s, t) = \left( \frac{M}{|s|} + \frac{M}{|t|} + \frac{M-t}{M} \right)$$

In this equation, M is the number of characters in common between the two strings, provided that:

$$M = |Com(s, t)|$$

$$S[i] \in Com(s, t) \text{ if and only if } \left[ i - \frac{\min(|s|, |t|)}{2}, i + \frac{\min(|s|, |t|)}{2} \right]$$

t is the number of common but transposed characters between the two strings. For example, if we want to compare the word article with the two words aricle and aricle, then the number of common letters is 6 and 7, respectively, and the

number of transposed letters is 0 and 1. Since the possibility of lexical errors in the design of large ontologies is not far-fetched, using the Winkler sweep criterion in the lexical similarity calculation section will reduce the negative effect of spelling errors in identifying mappings.

Also in this section, the following algorithms have been used:

- Overlap Coefficient
- Levenshtein Distance
- Jacquard similarity
- TriGram [22-23].

**4-2- Terminological Aggregation**

The similarity values obtained from each of the above criteria are stored in a local structured matrix called  $M_{(Local)}$  (i). These matrices are combined to obtain a final matrix called  $M_{(Global)}$ . In this method, to integrate the effect of the set of lexical similarity criteria, we use the averaging of normal values and use the following function to calculate the lexical similarity of two entities  $e_1$  and  $e_2$  from two ontologies:

$$\forall e_1 \in O_1, e_2 \in O_2: \\ Sim_T(e_1, e_2) = \sum_{i=1}^5 w_i \times Sim_{t_i}(text(e_1), text(e_2))$$

Given that  $\sum_{i=1}^5 w_i = 1$  is. In this formula  $Sim_{(t_i)}$  represents the similarity value obtained from applying the  $i$ th similarity criterion, which is  $1 \leq i \leq 5$ . Also,  $text(e_1)$  and  $text(e_2)$  represent the text labels related to the two entities  $e_1$  and  $e_2$ . The value  $w_i$  also represents the influence coefficient of the  $i$ th similarity criterion in calculating the final lexical similarity of the two entities.

**4-3- Semantic Matching**

Semantic matching is used to complement the lexical matching method in identifying anchors, because the lexical matching method relies only on the textual comparison of the content of entities, while it cannot compare this content semantically and conceptually. For example, consider two ontologies related to two different universities, in one of these two ontologies different departments of the university are named with the word "Faculty" and in the other with the word "Campus". Of course, comparing these two words will not show the similarity of the two, while these two words are semantically equivalent. For the semantic comparison of the content of entities, we used the WordNet database as an electronic database of vocabulary written in English, which is explained in detail in the second chapter about WordNet and we will explain how to use it in the following. This database is used to retrieve all meanings, synonyms and similarities of words related to a concept. We use the following hypothesis to define the similarity of two concepts by WordNet: Hypothesis: If two concepts defined by WordNet have a common synonym, then these two concepts are similar. A similarity degree  $\sigma$  is assigned to both compared concepts, which is a value in the range  $[0,1]$ . Assuming that the vocabulary set recovered by WordNet for concept  $w$  is represented by the symbol  $S(w)$ , the similarity degree between two concepts  $w_1$  and  $w_2$  is defined as follows:

$$\sigma(w_1, w_2) = \frac{|S(w_1) \cap S(w_2)|}{\min(|S(w_1)|, |S(w_2)|)}$$

Given that the textual content of an entity is a set of several words, the semantic similarity between the two sets must be determined. For this purpose, first, the set of words is extracted from the textual content of an entity. Then, the set of meanings and synonyms for each word is extracted from the WordNet database and a comparison between the meanings of both words is performed pairwise. The semantic similarity between the content of two entities will be a combination of the semantic content of the sets of words used in the definition of the two entities. The semantic similarity of the content of two entities is calculated using the following formula:

$$Sim_{wn}(e_1, e_2) = \frac{\sum_{w_1 \in T(e_1)} \sum_{w_2 \in T(e_2)} \sigma(w_1, w_2)}{|T(e_1)| + |T(e_2)|}$$

**4-4- Combining lexical similarity and semantic similarity**

As mentioned earlier, identifying anchors will play an important role in the efficiency of the alignment method. To identify anchors, we have used the maximum lexical similarity and semantic similarity of the entity content. At this stage, the similarity between both entities is calculated using the following formula:

$$Sim_{TW}(e_1, e_2) = (Sim_T(e_1, e_2) > Sim_{wn}(e_1, e_2)) ? Sim_T(e_1, e_2) : Sim_{wn}(e_1, e_2)$$

Once the content similarity of both entities is determined, the entity pairs are sorted in descending order based on the similarity degree, and the entity pairs with a similarity degree greater than the lexical similarity threshold value are selected as anchors.

**4-5- Structural matching of entities**

The identified entity pairs are used as anchors as input to the structural matching stage. At this level, ontology documents are analyzed graphically to consider the relationship between ontology concepts in the alignment calculations. For this purpose, we will use a method similar to the Similarity Flooding algorithm to calculate the structural similarity of entity pairs to calculate the mappings between graph nodes based on their neighborhood content. In this stage, two ontology

graphs along with the identified anchors and their similarity degrees are considered as inputs. Then, a propagation graph, which is a composite graph of the two initial graphs, is constructed. In this graph, each node has a score determined according to the first alignment stage. The scores of each node are propagated in this graph and the new score of each node is calculated based on the scores received from its neighbors. This procedure continues iteratively until the number of iterations reaches a certain threshold or the node scores reach a relative convergence. Nodes with higher scores as the output of this stage will be sent to the next stage.

#### 4-5-1- Graph chaos

The graph data model used to model ontologies is a labeled and directed graph. Each edge in this graph is represented as a triple (s,p,o) where s and o are the source and target nodes of the edge, and the middle element p is the edge label.

The similarity propagation graph is an intermediate data structure, derived from the graphs  $G_{O_1}$  and  $G_{O_2}$  of the two input ontologies, which is used to calculate the structural similarity of the entities. To demonstrate how to calculate the propagation graph, we first construct a pairwise connection graph (Structural Mapping Method) using the following definition:

$$((x, y), p, (x', y')) \in PCG(G_{O_1}, G_{O_2}) \Leftrightarrow (x, p, x') \in G_{O_1} \text{ and } (y, p, y') \in G_{O_2}$$

Each node in this graph is an element of  $A \times B$  that represents a pair of entities. As is clear from the definition, the edges of this graph are created based on the common edges between the two initial graphs.

For each edge in the propagation graph, an additional edge in the opposite direction is also considered. Each edge in this graph has a certain weight that indicates the ability of the edge to propagate information. These weights are in the range [0,1] and can be calculated in various ways.

#### 4-5-2- Calculating the structural similarity of pairs of entities using the diffusion technique

To calculate the structural similarity of pairs of entities, the propagation technique is used on the propagation graph. For this purpose, an initial score is first assigned to each node in the propagation graph. This score is determined based on the lexical-semantic similarity of each entity pair, and the node scores are sent to neighboring nodes using the propagation technique so that the final score of each node is calculated with knowledge of the similarity of neighbors and neighboring nodes.

Assuming that  $\sigma^i$  represents the score of a node at the i-th stage, in the i-th iteration of the propagation algorithm, the value of  $\sigma^i$  of each node is sent to all its neighbors according to the weight of the associated edge, and the score of each node is updated based on the values received at that stage. This update is performed using the following formula:

$$\begin{aligned} \sigma^0(x, y) &= Sim_{TW}(x, y), \\ \sigma^{i+1}(x, y) &= \sigma^i(x, y) + \sum_{(a_u, p, x) \in A, (b_u, p, y) \in B} \sigma^i(a_u, b_u) \cdot w((a_u, b_u), (x, y)) \\ &+ \sum_{(x, p, a) \in A, (y, p, b_v) \in B} \sigma^i(a_v, b_v) \cdot w((a_u, b_u), (x, y)) \end{aligned}$$

At each stage of the algorithm, after updating the scores of all nodes, these scores are normalized according to the highest score in the graph. The propagation algorithm continues iteratively until the scores of the nodes reach a relative convergence. This convergence is achieved when the value of  $\Delta(\sigma^n, \sigma^{n-1})$  for  $n > 0$  is less than a value of  $\epsilon$ . This value is calculated according to the following formula:

$$\Delta(\sigma^1, \sigma^{n-1}) = \sqrt{\frac{|N(PCG)|}{\sum_{i=1}^n (\sigma_i^n - \sigma_i^{n-1})}}$$

In this formula,  $|N(PCG)|$  denotes the number of nodes in the PCG graph and  $\sigma_i^n$  denotes the score of the ith node at the nth stage. If convergence is not achieved during the iterative propagation algorithm, the computations are terminated after the maximum number of iterations [24].

After the algorithm is completed, each node has a final score, which represents a combination of three lexical, structural, and semantic matchings. As mentioned earlier, each node in the propagation graph represents a pair of entities. Therefore, by sorting the nodes by scores and selecting the top k nodes, a set of mappings can be obtained.

#### 4-6- Evaluation of results

One of the most important points that should be considered when presenting a new method is the evaluation of the mentioned methods using existing criteria. As stated so far, the goal of ontology alignment is to find the set of existing mappings between two ontologies. Therefore, considering the specified goal, a way must be found to determine the quality of the result of ontology alignment. In this thesis, Precision, Recall and F criteria are used to evaluate the results of the proposed methods, which we will introduce in the following.

**4-6-1- Data set to be evaluated**

The set of ontologies developed in the OntoFarm project deals with conference organization. It includes 16 ontologies. These ontologies are suitable for ontology matching due to their heterogeneous source nature. Each of these datasets is briefly described in Table (5-1). These ontologies are based on three types of underlying resources:

- The actual conference (series) and its web pages (indicated in the Type column by the term “web”),
- The actual software tool to support the conference organization (indicated in the Type column by the term “tool”),
- The experience of individuals with personal participation in the organization of the actual conference (indicated in the Type column by the term “by expert” [15].

**Table (1) Statistical population and sample size**

| Name      | Type      | Number of Class | Number of data type attributes | Number of properties of objects |
|-----------|-----------|-----------------|--------------------------------|---------------------------------|
| Ekwa      | By expert | 74              | 0                              | 33                              |
| Sofsem    | By expert | 60              | 18                             | 46                              |
| Sigkdd    | Web       | 49              | 11                             | 17                              |
| Iasted    | Web       | 140             | 3                              | 38                              |
| Micro     | Web       | 32              | 9                              | 17                              |
| Confious  | Tools     | 57              | 5                              | 52                              |
| Pcs       | Tools     | 23              | 14                             | 24                              |
| OpenConf  | Tools     | 62              | 21                             | 24                              |
| ConfTool  | Tools     | 38              | 23                             | 13                              |
| Crs       | Tools     | 14              | 2                              | 15                              |
| Cmt       | Tools     | 36              | 19                             | 49                              |
| Cocus     | Tools     | 55              | 0                              | 35                              |
| Paperdyne | Tools     | 47              | 21                             | 61                              |
| Edas      | Tools     | 104             | 20                             | 30                              |
| MyReview  | Tools     | 39              | 17                             | 49                              |
| Linklings | Tools     | 37              | 16                             | 31                              |

**4-7- Investigating the sensitivity of the proposed method to changes in lexical similarity threshold**

As mentioned earlier, after calculating the lexical similarity of the entity pairs, only the group of entities that have a similarity greater than the lexical similarity threshold value are considered in the structural matching stage. We have investigated the sensitivity of the proposed method to the lexical similarity threshold on the Conference-Ekaw dataset. In this study, the values 0.1, 0.3, 0.5, 0.8 and 1 were considered for the Terminological\_Threshold variable and for each of these values, the performance of the proposed method was evaluated using the F-criterion. The results of this study are shown in Figure (2). It is clear from the figure that the proposed method has shown better performance with a lexical similarity threshold of 0.5. This threshold value has also been used in other experiments.



**Figure 2- Sensitivity of the proposed method to changes in the lexical similarity threshold**

**4-8- Examining the sensitivity of the proposed method to changes in the structural similarity threshold**

After the structural matching step, the similarity obtained for each pair of nodes will be a combination of three semantic and structural lexical measures. In this step, a number of entity pairs with a similarity value lower than the SF Sim Threshold threshold will be removed. These mappings will not be shown in the final aligned set due to their low

similarity. In this section, the sensitivity of the proposed method to the SF\_Sim\_Threshold threshold value is examined. For this purpose, the values 0.01, 0.04, 0.1, 0.12, 0.16, 0.2, 0.25, 0.3, 0.4, 0.6, and 0.9 are considered for this threshold and the F-criterion is calculated. The result of this experiment is shown in Figure (5-5). As is clear from this figure, the proposed method performed better with a threshold value of 0.2. Therefore, we chose this value as the structural similarity threshold.

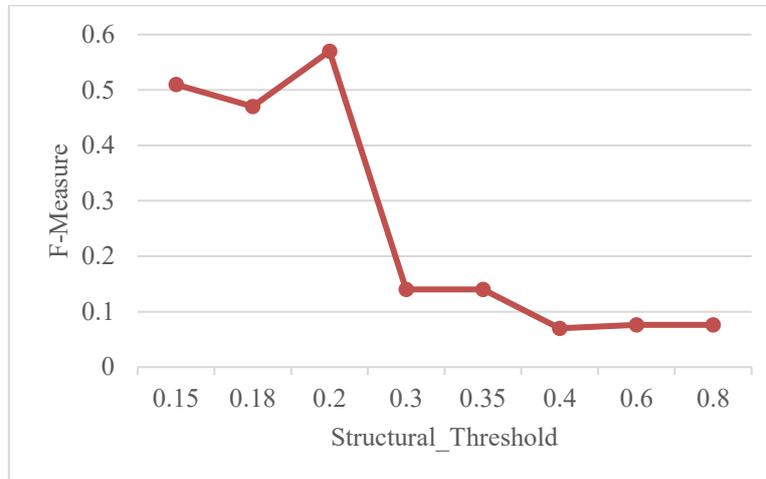


Figure-3 Sensitivity of the proposed method to changes in the structural similarity threshold

4-9- Comparison of the proposed method with other alignment methods

The proposed systems in the field of ontology alignment use different techniques in identifying the final mapping. Therefore, by comparing different systems, the efficiency of these techniques in finding the final mapping can be understood. In this section, the performance of the proposed system on the conference dataset is compared with the performance of the systems, Alin, Kepler, LogMap, LogMapLt, OntMat, Sonom and Wikiv3. The alignment results of all these systems are compared with the reference alignment and their precision, recall and F-criterion are calculated. The results of these comparisons are given below.

Table (2) Comparison of the performance of the proposed method with other systems on the Conference and Ekaw ontologies.

| Conference (60) - Ekaw (75) |          |        |           |           |        |           |
|-----------------------------|----------|--------|-----------|-----------|--------|-----------|
| System                      | #Correct | #Found | #Existing | Precision | Recall | F-Measure |
| AML                         | 18       | 23     | 25        | 0.78      | 0.72   | 0.75      |
| Alin                        | 6        | 8      | 25        | 0.75      | 0.24   | 0.36      |
| Kepler                      | 12       | 24     | 25        | 0.5       | 0.48   | 0.49      |
| LogMap                      | 12       | 20     | 25        | 0.6       | 0.48   | 0.53      |
| LogMapLt                    | 8        | 13     | 25        | 0.61      | 0.32   | 0.42      |
| OntMat                      | 48       | 6      | 25        | 0.6       | 0.92   | 0.13      |
| Sanom                       | 1        | 23     | 25        | 0.33      | 0.4    | 0.07      |
| Wikiv3                      | 9        | 14     | 25        | 0.64      | 0.36   | 0.46      |
| Our                         | 18       | 38     | 25        | 0.47      | 0.72   | 0.57      |

Table (3) Comparison of the performance of the proposed method with other systems on ConfOf ontologies with Ekaw

| ConfOf (39) - Ekaw (75) |          |        |           |           |        |           |
|-------------------------|----------|--------|-----------|-----------|--------|-----------|
| Name                    | #Correct | #Found | #Existing | Precision | Recall | F-Measure |
| AML                     | 16       | 17     | 20        | 0.94      | 0.8    | 0.86      |
| Alin                    | 6        | 6      | 20        | 1         | 0.3    | 0.46      |
| Kepler                  | 13       | 21     | 20        | 0.61      | 0.65   | 0.63      |
| LogMap                  | 14       | 15     | 20        | 0.93      | 0.7    | 0.8       |
| LogMapLt                | 48       | 1479   | 20        | 0.03      | 2.4    | 0.06      |
| OntMat                  | 48       | 1579   | 20        | 0.03      | 2.4    | 0.06      |

|               |    |    |    |      |      |      |
|---------------|----|----|----|------|------|------|
| <b>Sanom</b>  | 4  | 4  | 20 | 1    | 0.2  | 0.33 |
| <b>Wikiv3</b> | 8  | 11 | 20 | 0.72 | 0.4  | 0.51 |
| <b>Our</b>    | 17 | 26 | 20 | 0.65 | 0.85 | 0.73 |

#### 4-10- Evaluation analysis

Comparing the performance of different ontologies shows that the proposed system has been able to perform better than other systems in some datasets and has achieved acceptable results in other datasets. In addition, these figures show that the AML system has performed better than the proposed method in many cases. Regarding this system, it is necessary to mention that the AML system is an interactive system with the user and the intermediate results of the system are evaluated and selected by the user, so this system is not fully automated and its performance improvement is due to human interventions.

## V. CONCLUSION

Despite the heterogeneity between ontologies, we need to homogenize them. For this purpose, the proposed method uses three matchings to homogenize ontologies. To identify mappings related to two ontologies automatically, the first step is to identify anchors. All alignment calculations are performed based on the initial anchors. Therefore, accurate and correct identification of anchors has a significant impact on the accuracy of the alignment method. In the proposed method, anchors are identified by lexical and semantic analysis of entities, and the similarity between words is used to describe entities for lexical matching. For semantic matching of two entities, the previous method is also combined with the use of the Verdant auxiliary tool. Also, in the structural matching section between ontologies, the ontology graph and the similarity between two entities are calculated based on their neighborhood similarity using a modified version of the similarity flood algorithm. Then, we have compared our proposed method with other methods presented in OAEI such as: AML, Alin, Kepler, Logmap, LogMaplt, OntMat, Sonam and Wikiv3, which shows the efficiency of the proposed method compared to other methods in comparison and evaluation tables in terms of accuracy, recall and F measurement and the results of multiple evaluations.

It should be noted that the AML and Alin system is an interactive system with the user and the intermediate results of the system are evaluated and selected by the user.

## REFERENCES

- [1] Huang, W., & Harrie, L. Towards knowledge-based geovisualisation using Semantic Web technologies: A knowledge representation approach coupling ontologies and rules. *International Journal of Digital Earth*, 13(9), 976-997. (2020).
- [2] Martinez-Rodriguez, Jose L. Aidan Hogan, and Ivan Lopez-Arevalo. "Information extraction meets the semantic web: a survey." *Semantic Web 11.2* (2020): 255-335.
- [3] Ardjani, Fatima & Bouchiha, Djelloul & Malki, Mimoun. *Ontology-Alignment Techniques: Survey and Analysis*, *International Journal of Modern Education and Computer Science*, 7, 67-78, 2015.
- [4] Ouali, Imene & Ghazzi, Faiza & Taktak, Raouia & Saifeddine Hadj Sassi, Mohamed, *Ontology Alignment using Stable Matching*, *Procedia Computer Science*, 159, 746-755, 2019.
- [5] Mohammadi, M., & Rezaei, J, *Evaluating and comparing ontology alignment systems: An MCDM approach*, *Journal of Web Semantics*, 64, 2020.
- [6] Dhouib, Molka Tounsi & Catherine Faron Zucker & Andrea GB Tettamanzi, *An ontology alignment approach combining word embedding and the radius measure*, *International Conference on Semantic Systems*. Springer, Cham, 2019.
- [7] Algergawy, Alsayed & Faria, Daniel & Ferrara, Alfio & Fundulaki, Irini & Harrow, Ian & Hertling, Sven & Jiménez-Ruiz, Ernesto & et al. *Results of the ontology alignment evaluation initiative 2019*, In *CEUR Workshop Proceedings*, 2536, 46-85. 2019.
- [8] J. Wang, Z, Ding, C. Jiang, GaomL *Genetic Algorithm Based Ontology Matching*, 2006 *IEEE Asia-Pacific Conference on Services Computing*, Guangzhou, China, 2006, pp. 617-620.
- [9] Ngo, DuyHoa, and Zohra Bellahsene. "YAM++: A multi-strategy based approach for ontology matching task." *International Conference on Knowledge Engineering and Knowledge Management*. Springer, Berlin, Heidelberg, 2012.
- [10] Jiménez-Ruiz, Ernesto, Bernardo Cuenca Grau, and Valerie Cross. "LogMap family participation in the OAEI 2017." *CEUR Workshop Proceedings*. Vol. 2032. 2017.
- [11] Kachroudi, Marouen, Gayo Diallo, and Sadok Ben Yahia. "KEPLER at OAEI 2018." *OM@ ISWC*. 2018.
- [12] Drame, K., Diallo, G., Delva, F., Dartigues, J., Mouillet, E., Salamon, R., Mougín, F.: *Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology: An application to alzheimer's disease*. *Journal of Biomedical Informatics* 48 (2014) 171–182

- [13] da Silva, Jomar, Kate Revoredo, and Fernanda Araujo Baiao. "ALIN results for OAEI 2018." OM@ ISWC. 2018.
- [14] Vijaymeena, M. K., and K. Kavitha. "A survey on similarity measures in text mining." Machine Learning and Applications: An International Journal 3.2 (2016): 19-28.
- [15] Lee, Mong Li, et al. "XClust: clustering XML schemas for effective integration." Proceedings of the eleventh international conference on Information and knowledge management. 2002.