Received: July 2, 2020. Revised: July 28, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

hMatcher: Matching Schemas Holistically

Aola Yousfi[1]* **Moulay Hafid El Yazidi[1]** **Ahmed Zellou[1]**

[1]*Software Projects Management Research Team Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes Mohammed V University in Rabat, Morocco*
* Corresponding author's Email: aola.yousfi@gmail.com

Abstract: Schema matching is critical for applications that manipulate data across heterogeneous, autonomous and scattered data sources. We pick the schema matching approach based on the total number of data sources we wish to integrate: holistic matching approaches are ideally used for a big to a huge total number of data sources, while pairwise matching approaches are ideally used for a small to a medium total number of data sources. Nonetheless, the state of the art matching approaches obtain a very moderate (sometimes poor) matching accuracy. Furthermore, the state of the art holistic schema matching approaches proceed in a series of two-way matching steps. In this paper, we present hMatcher, an effective approach to holistic schema matching. To perform collective schema matching, hMatcher generates frequent schema elements before proceeding with the matching. To reach high matching accuracy, hMatcher employs a context-based semantic similarity measure. Experimental results on a real-world domain dataset show that hMatcher performs holistic schema matching properly, reaches a high matching accuracy (Precision=0.89;Recall=0.66;Overall=0.57), and outperforms the state of the art matching approaches in terms of matching accuracy.

Keywords: Holistic schema matching, Matching accuracy, Semantic similarity.

1. Introduction

Schema matching aims at finding semantically corresponding elements (according to [1-5], they are also called semantically similar elements or matches) in multiple, autonomous, heterogeneous and distributed schemas of data sources. According to [6], schema matching is very crucial for applications that manipulate data across different data sources, examples of areas where these applications are used involve bioinformatics, data integration on the World Wide Web, e-commerce, data warehousing and scientific collaboration. Therefore, schema matching got loads of attention from the research community over the past decades (and it is still to this day a huge area of interest for researchers) (see [7-10]) for surveys).

Schema matching approaches are grouped into two major categories: pairwise matching and holistic matching. The former aims at finding the semantically corresponding elements between two schemas at a time, which is insufficient when we wish to match a large number of schemas. Therefore, the latter was created to overcome the limitations of pairwise matching approaches as it matches numerous schemas simultaneously.

Nevertheless, the state of the art holistic schema matching approaches (according to [2,3], they are also called collective schema matching approaches) face two main challenges. First, they often operate in a series of two-way matching steps which contradicts the main goal of collective schema matching as they do not necessarily match multiple schemas at once, but instead they operate incrementally: they first match two schemas and combine the results into one integrated schema, and then matches a third schema to the combined schema (e.g. Holontology [11] and PORSCHE [12]). Second, they often achieve a very moderate (even poor in some cases) matching accuracy, which implies a continuous human assistance to correct the matches, that is to say: add missed matches and remove false matches.

DOI: 10.22266/ijies2020.1031.43

In what follows, we present the key challenges we faced when working on this research project:

- Create a well-defined semantic similarity measure between two words, a word and a set of words, or two sets of words.
- Come up with a well-defined approach to generate frequent schema elements.
- Define an approach to decrease the total number of rare schema elements.
- Create an efficient approach to perform holistic schema matching.

The main contribution of this paper is that it proposes hMatcher, an effective approach to holistic schema matching. The key idea of hMatcher is to (1) perform holistic schema matching; and (2) achieve a high matching accuracy. To this end, hMatcher uses a semantic similarity measure, and a hierarchical lexical dictionary along with an abbreviations & acronyms database.

In summary, we make the following concrete additions:

- We define a new Context-based Semantic Similarity Measure (CSSM) to calculate the semantic similarity value between schema elements.
- We propose a new algorithm to generate frequent schema elements.
- We propose a holistic schema matching approach.
- We evaluate hMatcher on a real-world domain dataset and show that it is able to match numerous schemas simultaneously and reach a very high matching accuracy.

The remaining of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the architecture of hMatcher. Section 4 presents experimental results. Section 5 concludes this paper and discusses future research directions.

2. Related work

In this section, we review a variety of the most well-known matching approaches that are most relevant to our present work.

ALIN [13] is a human-interactive ontology matching approach. According to [13], ALIN takes as input two ontologies and delivers as output a set of alignments. It proceeds in two key steps. (1) It defines the initial mappings. (2) It changes the mappings according to human experts' feedbacks which improve the quality of the matches. The second step is repeated till experts run out of suggestions.

ALOD2Vec [14] uses the WebIsALOD database of hypernym relations extracted from the Web. According to [14], ALOD2Vec also uses both element-based information and label-based information. To capture the similarity score between nodes of the knowledge graph (WebIsALOD is viewed as a knowledge graph), ALOD2Vec applies RDF2Vec which converts RDFs into vectors.

AgreementMakerLight (AML) [15] is an ontology matching approach. It is an updated version of AgreementMaker [16]. According to [15], AML comprises two modules: the ontology loading module and the ontology matching module. On the one hand, the ontology loading module loads the ontologies as well as the external resources, and then generates the ontology objects. On the other hand, the ontology matching module aligns the ontology objects generated by the previous module.

Deep Ontology MatchEr (DOME) [17] uses doc2vec and large texts that describe the concepts of the ontologies. To deal with the main issue of matching similar large texts, DOME uses topic modeling for instance Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA).

LogMap [18] is a scalable and logic-based ontology matching approach. According to [18], LogMap exploits lexical indexation, logic-based module extraction, propositional horn reasoning, axiom tracking, local repair and semantic indexation to match two given ontologies. LogMapLt is a lightweight variant of LogMap.

FCAMapX [19] is an automated ontology alignment system. According to [19], FCAMapX is based on Formal Concept Analysis, which is a mathematical model for analyzing structuring concepts.

KEPLER [20] is an ontology matching system. According to [20], KEPLER takes advantage of the expressiveness of the Web Ontology Language (OWL) statements using six key steps: parsing, partitioning/translation, indexing, candidate mappings identification, filtering and recovery, and alignment generation.

Simulated ANnealing-based Ontology Matching (SANOM) [21] uses the notorious Simulated Annealing (SA) [22] to find out semantically corresponding elements between two ontologies, which results on a potential intermediate alignment. According to [21], the evolution of that alignment needs to use both lexical similarity metrics and structural similarity metrics.

Lily [23] is an ontology alignment approach. According to [23], Lily's main advantage is the following: it is able to process normal ontologies, weak informative ontologies [24], ontology mapping debugging [25] and ontology matching tuning [26].

Holontology [11] is a holistic ontology matching approach based on the Linear Program for Holistic Ontology Matching (LPHOM) approach [27,28]. According to [11], Holontology uses many similarity measures and dissimilarity distances such as exact match, Levenstein, Jaccard and Lin to match two ontologies or multiple ontologies at once after it converts them into an internal predefined format. Then, Holontology converts the results into alignments exported by RDF.

The eXtended Mapping (XMap) algorithm [29] is a lexical and structural-based semantic matching approach. According to [29], XMap uses WordNet [30] and the Unified Medical Language System (UMLS) [31] which is a collection of many vocabularies, key terminology, classification and coding standards related to the biomedical sciences to capture semantic similar concepts from the input ontologies.

These schema matching approaches have three key limitations. First, they often capture multiple possible matches (correct and incorrect matches), which means that they require a human expert to decide on whether the matches are correct or not. The main problem is that the user may not always be familiar with these domain-specific terms. Therefore, [32-34] introduced a new solution to that issue: Single Correspondence Correctness Question (Single CCQ) and Multiple Correspondence Correctness Question (Multiple CCQ), two new crowdsourcing based-approaches. Their main goal is to reduce the set of possible matches hence leaving merely the most likely to be correct. Both Single CCQ and Multiple CCQ formulate simple and non-technical Yes/No questions to the user. Single CCQ determines the most crucial question to ask; while, Multiple CCQ (an extension of Single CCQ) determines the most crucial questions to ask based on the previous answers. Even though this solution may solve the issue of multiple possible matches, it clearly makes schema matching much more human-dependent. Second, schema matching becomes much more time-consuming in particular when we wish to match a huge number of schemas as they match schemas incrementally (rather than simultaneously) in a series of two-way matching steps. Third, current matching approaches often obtain a very moderate or poor matching accuracy.

In the next section, we will present hMatcher, a solution to the schema matching problem (human-dependency, impracticality when matching a huge number schemas, and low matching accuracy) we described above.

3. The hMatcher approach

The hMatcher architecture (see Fig. 1) comprises three key components: frequent elements generator, schema matcher and rare elements matcher. (1) Let $\mathbb{S}$ be a set of schemas, and let $\mathbb{S}_{Learning} \in \mathbb{S}$ be the learning schemas (see definition 3.1.), the frequent elements generator takes as input $\mathbb{S}_{Learning}$, employs an abbreviations & acronyms database as well as a hierarchical lexical dictionary, and generates as output the frequent schema elements $\mathbb{F}$. (2) Let $\mathbb{S}_{Testing} \in \mathbb{S}$ be the testing schemas (see definition 3.3.), the schema matcher takes as input $\mathbb{S}_{Testing}$, exploits the frequent schema elements to identify the matches Φ. (3) The rare elements matcher reuses previous results to identify new matches in the rare schema elements set $\mathbb{R}$. Note that the frequent elements generator takes place solely once which is at the beginning of the matching process.

Definition 3.1. (Learning schemas). The learning schemas refer to the schemas we use to generate the initial set of frequent schema elements.

Definition 3.2. (Schema element; Frequent schema element; Rare schema element). A schema element e is an element from a schema S such that e represents a particular data stored in the data source of S.

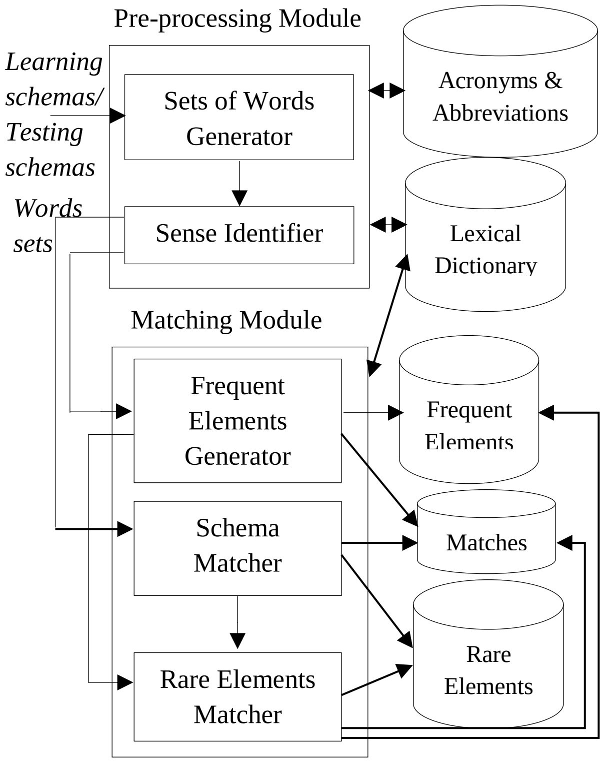

Figure. 1 The hMatcher architecture

Received: July 2, 2020. Revised: July 28, 2020.

We say that e is a frequent schema element if and only if it has duplicates in a certain number of schemas describing the same domain.

A rare schema element is a schema element that does not belong to $\mathbb{F}$.

Definition 3.3. (Testing schemas). The testing schemas refer to the schemas we match using the frequent schema elements.

The rest of this section first describes the frequent elements generator (see subsection 3.1), then the schema matcher (see subsection 3.2), and finally the rare elements matcher (see subsection 3.3).

3.1 The frequent elements generator

Let $\mathbb{S}_{Learning} = \{S_1, S_2, \dots, S_p\}$ be the learning schemas. Inspired by the success of the pre-processing strategy introduced in [35], the frequent elements generator first employs that strategy in order to generate from every schema element e (see definition 3.2.) a words set θ that fully describes its meaning. The words sets generated from S_1 are denoted by Θ_1, the words sets generated from S_2 are denoted by Θ_2, etc. The frequent elements generator then operates in three main steps:

a. Capture the matches

Let $e_1 \in S_1$ and $e_2 \in S_2$ be two schema elements, and $\theta_1 = \{w_{1,1}, w_{1,2}, \dots, w_{1,|\theta_1|}|\}$ and $\theta_2 = \{w_{2,1}, w_{2,2}, \dots, w_{2,|\theta_2|}|\}$ be their respective words sets. The frequent elements generator first employs the Context-based Semantic Similarity Measure (CSSM) Eq. (1) presented below so as to tell whether e_1 and e_2 are semantically similar or not.

$$CSSM_{sets}(\theta_1, \theta_2) = \frac{1}{\min(|\theta_1|,|\theta_2|)} \times \sum_{i=1}^{\min(|\theta_1|,|\theta_2|)} \max\left(m_{i,j}\right)_{1 \le j \le \max(|\theta_1|,|\theta_2|)} \quad (1)$$

Where:

- $|\theta_1|$ and $|\theta_2|$ are the cardinalities of θ_1 and θ_2, respectively.
- $M = \left(m_{i,j}\right)_{\substack{1 \le i \le |\theta_1| \\ 1 \le j \le |\theta_2|}}$ is the similarity matrix whose individual items are defined as follows: $m_{i,j} = CSSM_{words}(w_{1,i}, w_{2,j})$ (see Eq. (11)).

For every word w from the hierarchical lexical dictionary entries, we have (1) the hypernyms of w constitute a general definition of w; (2) the direct hyponyms of w constitute a more specific definition of w; and (3) together form a complete definition of w. As a result, given two words w_1, w_2, in order to compare w_1 to w_2, we have to compare $\{w_1, P_{w_1}, H_{w_1}\}$ to $\{w_2, P_{w_2}, H_{w_2}\}$, where P_{w_1} and P_{w_2} are the hypernyms of w_1 and w_2 in the hierarchical lexical dictionary, respectively; and H_{w_1} and H_{w_2} are the direct hyponyms of w_1 and w_2 in the same dictionary, respectively. Hence, we calculate the similarity between w_1 and w_2 Eq. (2),w_1 and P_{w_2} Eq. (3),w_1 and H_{w_2} Eq. (4),P_{w_1} and w_2 Eq. (5), P_{w_1} and P_{w_2} Eq. (6), P_{w_1} and H_{w_2} Eq. (7), H_{w_1} and w_2 Eq. (8), H_{w_1} and P_{w_2} Eq. (9), and H_{w_1} and H_{w_2} Eq.(10). Note that we consider solely non-shared hypernyms as a result $P_{w_1} \cap P_{w_2} = \emptyset$. Below, we present all nine sub-measures:

$$SM_1(w_1, w_2) = |s_{w_1} \cap s_{w_2}| + |s_{w_1} \cap (w_2 \cup Sy_{w_2})| + |s_{w_2} \cap (w_1 \cup Sy_{w_1})| \quad (2)$$

Where:

- s_{w_1} and s_{w_2} are the senses of w_1 and w_2, respectively.
- Sy_{w_1} and Sy_{w_2} are the synonyms of w_1 and w_2, respectively.

$$SM_2(w_1, P_{w_2}) = \sum_{i=1}^{|P_{w_2}|} \left|s_{w_1} \cap s_{P_{w_2}}\right| + |s_{w_1} \cap (P_{w_2} \cup Sy_{P_{w_2}})| + |s_{P_{w_2}} \cap (w_1 \cup Sy_{w_1})| \quad (3)$$

Where:

- s_{w_1} and $s_{P_{w_2}}$ are the senses of w_1 and P_{w_2}, respectively.
- Sy_{w_1} and $Sy_{P_{w_2}}$ are the synonyms of w_1 and P_{w_2}, respectively.

$$\mathrm{SM}_3(\mathrm{w}_1, \mathrm{H}_{\mathrm{w}_2}) = \sum_{\mathrm{i}=1}^{|\mathrm{H}_{\mathrm{w}_2}|} \mathrm{card}\left(s_{w_1} \cap \mathrm{s}_{\mathrm{H}_{\mathrm{w}_{2_i}}}\right) + \mathrm{card}\left(s_{\mathrm{w}_1} \cap \left(\mathrm{H}_{\mathrm{w}_{2_i}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{2_i}}}\right)\right) + \mathrm{card}\left(s_{\mathrm{H}_{\mathrm{w}_{2_i}}} \cap \left(\mathrm{w}_1 \cup \mathrm{Sy}_{\mathrm{w}_1}\right)\right) \quad (4)$$

Where:

- s_{w_1} and $s_{H_{w_2}}$ are the senses of w_1 and H_{w_2}, respectively.
- Sy_{w_1} and $Sy_{H_{w_2}}$ are the synonyms of w_1 and H_{w_2}, respectively.

$$\mathrm{SM}_4\left(\mathrm{P}_{\mathrm{w}_1}, \mathrm{w}_2\right) = \sum_{i=1}^{|\mathrm{P}_{\mathrm{w}_1}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{w}_2}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{w}_2 \cup \mathrm{Sy}_{\mathrm{w}_2}\right)\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{w}_2} \cap \left(\mathrm{P}_{\mathrm{w}_{1_i}} \cup \mathrm{Sy}_{\mathrm{P}_{\mathrm{w}_{1_i}}}\right)\right) \tag{5}$$

Where:

- s_{w_2} and $s_{P_{w_1}}$ are the senses of w_2 and P_{w_1}, respectively.
- $Sy_{w_2}$and $Sy_{P_{w_1}}$are the synonyms of $w_2$and P_{w_1}, respectively.

$$\mathrm{SM}_5\left(\mathrm{P}_{\mathrm{w}_1},\mathrm{P}_{\mathrm{w}_2}\right) = \sum_{i=1}^{|\mathrm{P}_{\mathrm{w}_1}|} \sum_{j=1}^{|\mathrm{P}_{\mathrm{w}_2}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{P}_{\mathrm{w}_{2_j}} \cup \mathrm{Sy}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right)\right) + \mathrm{card}\left(\left(\mathrm{P}_{w_{1_i}} \cup \mathrm{Sy}_{\mathrm{P}_{\mathrm{w}_{1_i}}}\right) \cap \mathrm{s}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right) \tag{6}$$

Where:

- $s_{P_{w_1}}$ and $s_{P_{w_2}}$ are the senses of P_{w_1} and P_{w_2}, respectively.
- $Sy_{P_{w_1}}$ and $Sy_{P_{w_2}}$ are the synonyms of P_{w_1} and P_{w_2}, respectively.

$$\mathrm{SM}_6\left(\mathrm{P}_{\mathrm{w}_1},\mathrm{H}_{\mathrm{w}_2}\right) = \sum_{i=1}^{|\mathrm{P}_{\mathrm{w}_1}|} \sum_{j=1}^{|\mathrm{H}_{\mathrm{w}_2}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{P}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{H}_{\mathrm{w}_{2_j}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right)\right) + \mathrm{card}\left(\left(\mathrm{P}_{\mathrm{w}_{1_i}} \cup \mathrm{Sy}_{\mathrm{P}_{\mathrm{w}_{1_i}}}\right) \cap \mathrm{s}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right) \tag{7}$$

Where:

- $s_{P_{w_1}}$ and $s_{H_{w_2}}$ are the senses of P_{w_1} and H_{w_2}, respectively.
- $Sy_{P_{w_1}}$ and $Sy_{H_{w_2}}$ are the synonyms of P_{w_1} and H_{w_2}, respectively.

$$\mathrm{SM}_7\left(\mathrm{H}_{\mathrm{w}_1}, \mathrm{w}_2\right) = \sum_{i=1}^{|\mathrm{H}_{\mathrm{w}_1}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{w}_2}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{w}_2 \cup \mathrm{Sy}_{\mathrm{w}_2}\right)\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{w}_2} \cap \left(\mathrm{H}_{w_{1_i}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{1_i}}}\right)\right) \tag{8}$$

Where:

- s_{w_2} and $s_{H_{w_1}}$ are the senses of w_2 and H_{w_1}, respectively.
- $Sy_{w_2}$and $Sy_{H_{w_1}}$are the synonyms of $w_2$and H_{w_1}, respectively.

$$\mathrm{SM}_8\left(\mathrm{H}_{\mathrm{w}_1},\mathrm{P}_{\mathrm{w}_2}\right) = \sum_{i=1}^{|\mathrm{H}_{\mathrm{w}_1}|} \sum_{j=1}^{|\mathrm{P}_{\mathrm{w}_2}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{P}_{\mathrm{w}_{2_j}} \cup \mathrm{Sy}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right)\right) + \mathrm{card}\left(\left(\mathrm{H}_{\mathrm{w}_{1_i}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{1_i}}}\right) \cap \mathrm{s}_{\mathrm{P}_{\mathrm{w}_{2_j}}}\right) \tag{9}$$

Where:

- $s_{H_{w_1}}$ and $s_{P_{w_2}}$ are the senses of H_{w_1} and P_{w_2}, respectively.
- $Sy_{H_{w_1}}$ and $Sy_{P_{w_2}}$ are the synonyms of H_{w_1} and P_{w_2}, respectively.

$$\mathrm{SM}_9\left(\mathrm{H}_{\mathrm{w}_1},\mathrm{H}_{\mathrm{w}_2}\right)= \sum_{i=1}^{|\mathrm{H}_{\mathrm{w}_1}|} \sum_{j=1}^{|\mathrm{H}_{\mathrm{w}_2}|} \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \mathrm{s}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right) + \mathrm{card}\left(\mathrm{s}_{\mathrm{H}_{\mathrm{w}_{1_i}}} \cap \left(\mathrm{H}_{\mathrm{w}_{2_j}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right)\right) + \mathrm{card}\left(\left(\mathrm{H}_{\mathrm{w}_{1_i}} \cup \mathrm{Sy}_{\mathrm{H}_{\mathrm{w}_{1_i}}}\right) \cap \mathrm{s}_{\mathrm{H}_{\mathrm{w}_{2_j}}}\right) \tag{10}$$

Where:

- $s_{H_{w_1}}$ and $s_{H_{w_2}}$ are the senses of H_{w_1} and H_{w_2}, respectively.
- $Sy_{H_{w_1}}$ and $Sy_{H_{w_2}}$ are the synonyms of H_{w_1} and H_{w_2}, respectively.

We combine all nine sub-measures into one single measure Eq. (11) aimed to calculate the semantic similarity between words:

$$CSSM_{words}(w_1, w_2) = \begin{cases} 1, if\ w_1\ and\ w_2\ are\ synonyms\ or\ one\ of\ them\ is\ a\ direct\ hyponym\ of\ the\ other \\ 0, if\ \sqrt[4]{\begin{matrix}0.8 \times (SM_1 + SM_5 + SM_9) \\ +0.2 \times \sum_{\substack{i=2\\i\neq5}}^{8} SM_i\end{matrix}} \times e^{\left(\frac{\sum_{\substack{i=1\\SM_i\neq0}}^{9} 1}{9}\right)} \leq 1 \\ \left(\dfrac{\sqrt[4]{\begin{matrix}0.8\times(SM_1+SM_5+SM_9) \\ +0.2\times\sum_{\substack{i=2\\i\neq5}}^{8} SM_i\end{matrix}} \times e^{\left(\frac{\sum_{\substack{i=1\\SM_i\neq0}}^{9} 1}{9}\right)} - 1}{\sqrt[4]{\begin{matrix}0.8\times(SM_1+SM_5+SM_9) \\ +0.2\times\sum_{\substack{i=2\\i\neq5}}^{8} SM_i\end{matrix}} \times e^{\left(\frac{\sum_{\substack{i=1\\SM_i\neq0}}^{9} 1}{9}\right)} + 1}\right)^{1/2}, otherwise \end{cases} \quad (11)$$

We applied CSSM on the words sets generated from the conference schemas (see section 4). The findings formed a set of similarity values, each is the similarity between two sets. The selection of the threshold value was based on the reference matches, identified manually by a group of thirty-five Ph.D. students from our university. We noticed that almost all matched sets have a similarity value ≥ 0.8. Thus, CSSM has a threshold of 0.8 which means that the pair whose similarity value is greater than or equal to 0.8 are considered matched; and the pair whose similarity value is inferior to 0.8 are not matched.

b. Determine frequent schema elements

Given a schema $S \in \mathbb{S}_{Learning}$, let e be an element from S. We use Element Frequency-Schema Frequency (EF-SF) defined in Eq. (12) which is inspired by the popularity and the success of the Term Frequency-Inverse Document Frequency (TF-IDF) [36] in order to determine the degree of frequency of e.

$$EF - SF_{e\in S,\mathbb{S}_{Learning}} = EF_{e,S} \times SF_{e,\mathbb{S}_{Learning}} = e^{ef_{e,S}} \times e^{\left(\frac{sf_e}{|\mathbb{S}_{Learning}|}\right)} \quad (12)$$

Where:

- $ef_{e,S}$ is the frequency of e in S, such that $ef_{e,S} = \frac{count\ of\ e\ in\ S}{count\ of\ elements\ in\ S}$
- sf_e is the number of schemas containing e.
- $|\mathbb{S}_{Learning}| = p$ is the cardinality of $\mathbb{S}_{Learning}$.

We say that an element e is frequent if and only if its degree of frequency satisfies the following:

$$\lfloor EF - SF_{e\in S,\mathbb{S}_{Learning}} \rfloor \geq log\left(\frac{\sqrt[2]{m^4+1}}{\sqrt[2]{|\mathbb{S}_{Learning}|^2 - 1}}\right) (13)$$

Where:

- m is the number of elements in $\mathbb{S}_{Learning}$.
- $|\mathbb{S}_{Learning}| = p$ is the cardinality of $\mathbb{S}_{Learning}$.

c. See if there are other frequent elements

If we extend the number of learning schemas $\mathbb{S}_{Learning}$ but we end up having $\mathbb{F} = constant$, then the frequent elements generator stops. Otherwise, it repeats step and step b for more schemas.

Algorithm 1 Summarizes these steps:

```
Algorithm 1: FrequentElementsGenerator ($\boldsymbol{\Theta_1, \Theta_2, \ldots, \Theta_p}$)
Input:
$\boldsymbol{\Theta_1, \Theta_2, \ldots, \Theta_p}$: The words sets generated from $\boldsymbol{S_1, S_2, \ldots, S_p}$
Output:
$\mathbb{F}$: The frequent schema elements

$\mathbb{F} \leftarrow \emptyset$
Generate the matches $\boldsymbol{\Phi}$ between $\boldsymbol{\Theta_1, \Theta_2, \ldots, \Theta_p}$ according to CSSM
For each $\boldsymbol{\varphi}$ in $\boldsymbol{\Phi}$
If ( $\boldsymbol{e \in \varphi}$ and $\lfloor \boldsymbol{EF - SF_{e\in S,\mathbb{S}_{Learning}}} \rfloor \geq \boldsymbol{log}\left(\frac{\sqrt[2]{\boldsymbol{m^4+1}}}{\sqrt[2]{|\mathbb{S}_{\boldsymbol{Learning}}|^2-\boldsymbol{1}}}\right)$ ) Then $\mathbb{F} \leftarrow \mathbb{F} \cup \boldsymbol{e}$ /* $\mathbb{F}$ stores one element e in $\boldsymbol{\varphi}$*/
End if
End for
Return $\mathbb{F}$
```

3.2 The schema matcher

Let $\mathbb{S}_{Testing} = \{S_{p+1}, S_{p+2}, \ldots, S_n\}$ be the testing schemas, and let $\Theta_{p+1}, \Theta_{p+2}, \ldots, \Theta_n$be the words sets generated from $S_{p+1}, S_{p+2}, \ldots, S_n$, respectively. The schema matcher uses $\mathbb{F}$ to generate the matches Φ. To do so, it proceeds in two key steps:

a. Calculate the semantic similarity values

It uses CSSM (Eq. (1) and Eq. (11)) to compare the words sets $\Theta_{p+1}, \Theta_{p+2}, \dots, \Theta_n$ to the frequent schema elements $\mathbb{F}$.

b. Capture new matches

Every words set $\theta_i \in \{\Theta_{p+1}, \Theta_{p+2}, \dots, \Theta_n\}$ that has a semantically corresponding element $f_i \in \mathbb{F}$, its associated element e_i will be added to the matches list Φ such that $\varphi \leftarrow \varphi \cup e_i$, where $f_i \in \varphi$ and $\varphi \in \Phi$.

Algorithm 2 summarizes these steps:

```
Algorithm 2: SchemaMatcher ($(\boldsymbol{\Theta_{p+1}}, \boldsymbol{\Theta_{p+2}}, \dots, \boldsymbol{\Theta_n})$)
Input:
$\boldsymbol{\Theta_{p+1}}, \boldsymbol{\Theta_{p+2}}, \dots, \boldsymbol{\Theta_n}$: The words sets generated from $\mathbf{S_{p+1}}, \mathbf{S_{p+2}}, \dots, \mathbf{S_n}$
Output:
$\boldsymbol{\Phi}$: The matches

For each $\boldsymbol{\Theta}$ in $\{\boldsymbol{\Theta_{p+1}}, \boldsymbol{\Theta_{p+2}}, \dots, \boldsymbol{\Theta_n}\}$
    Generate the matches $\boldsymbol{\Phi}$ between $\boldsymbol{\Theta}$ and $\mathbb{F}$ according to CSSM
End for
Return $\boldsymbol{\Phi}$
```

3.3 The rare elements matcher

The rare elements matcher uses the transitivity principle (see theorem 1) to match the rare schema elements.

Theorem 1. (Transitive relation). A binary relation $\mathfrak{R}$ is transitive over a set B if and only if it satisfies the following:

$$\forall\, x, y, z \in B, (x\mathfrak{R}y \wedge y\mathfrak{R}z) \Rightarrow x\mathfrak{R}z \tag{14}$$

The rare elements matcher applies the transitivity principle as follows:

Let S_1 and S_2 be two schemas, let $r_1 \in \mathbb{R}$ and $r_2 \in \mathbb{R}$ be two rare schema elements from S_1 and S_2, respectively; and let $\mathbb{F} = \{f_1, f_2, \dots f_q\}$ such that $q \in \mathbb{N}^*$ be the set of frequent schema elements. We have the following:

$$\forall\, i \in \{1,2,\dots,q\}, CSSM(r_1, f_i) = CSSM(r_2, f_i) \pm 0.05 \Rightarrow r_1 \; and \; r_2 \; are \; matched \tag{15}$$

If r_1 (or r_2) satisfies Eq. (13), then the set of frequent schema elements is updated as follows:

$$\mathbb{F} \leftarrow \mathbb{F} \cup r_1 \text{ OR } \mathbb{F} \leftarrow \mathbb{F} \cup r_2 \text{ (not both)}$$

And the rare schema elements list is updated as follows:

$$\mathbb{R} \leftarrow \mathbb{R} \cup r_1 \text{ AND } \mathbb{R} \leftarrow \mathbb{R} \cup r_2$$

Algorithm 3 summarizes this:

```
Algorithm 3: RareElementsMatcher ($\mathbb{F}, \mathbb{R}$)
Input:
$\mathbb{F}$: Frequent schema elements
$\mathbb{R}$: Rare schema elements
Output:
$\boldsymbol{\Phi}$; $\mathbb{F}$; $\mathbb{R}$: The matches; Frequent schema elements; Rare schema elements

For each $\mathbf{r_1}, \boldsymbol{r_2} \in \mathbb{R}$
    If ( $\forall\, \boldsymbol{f} \in \mathbb{F}, \boldsymbol{CSSM(r_1, f_i) = CSSM(r_2, f_i) \pm 0.05}$) Then
        $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} \cup \boldsymbol{r_1}$ /* $\boldsymbol{\varphi} \in \boldsymbol{\Phi}$ contains the matches of $\boldsymbol{r_1}$*/
        $\mathbb{F} \leftarrow \mathbb{F} \cup \boldsymbol{r_1}$ OR $\mathbb{F} \leftarrow \mathbb{F} \cup \boldsymbol{r_2}$ (not both)
        $\mathbb{R} \leftarrow \mathbb{R} \cup \boldsymbol{r_1}$ AND $\mathbb{R} \leftarrow \mathbb{R} \cup \boldsymbol{r_2}$
    End if
End for
Return ($\mathbb{F}, \boldsymbol{\Phi}, \mathbb{R}$)
```

4. Experiments and evaluations

In this section, we first evaluate hMatcher in terms of matching accuracy, and then compare the findings to the state of the art matching systems.

4.1 Datasets

We evaluated hMatcher on the *Conference* dataset which includes sixteen ontologies all describing the domain of organizing academic conferences. The ontologies were used in OAEI 2019 and are available for free download on the Web[1]. It has twenty-one reference alignments formed from seven out of sixteen ontologies.

4.2 Measures

We use the following measures to evaluate the matches generated by hMatcher on the *Conference*.

[1] http://oaei.ontologymatching.org/2019/

Received: July 2, 2020. Revised: July 28, 2020.

$$Precision = \frac{Accurate\ matches}{Accurate\ matches + Inaccurate\ matches} \quad (16)$$

Eq. (16) is the probability of correct matches among all matches returned by the matching system.

$$Recall = \frac{Accurate\ matches}{Missed\ matches + Accurate\ matches} \quad (17)$$

Eq. (17) is the probability of correct matches returned by a matching system among the reference matches.

$$Overall = Recall \times \left(2 - \frac{1}{Precision}\right) \quad (18)$$

Eq. (18) measures the amount of manual post-effort required to add missed matches and remove false matches.

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (19)$$

Eq. (19) is the harmonic mean of *Precision* and *Recall*.

The ideal case scenario is when all four metrics reach their largest values:

$$\text{Precision} = \text{Recall} = \text{Overall} = \text{F} - \text{Measure} = 1$$

We then compare the findings obtained by hMatcher on the *Conference* dataset against previously published results of twelve well-known ontology matching systems (Holontology [11], AML [37], DOME [17], LogMap [18], XMap [29], KEPLER [20], ALIN [13], SANOM [21], FCAMapX [19], LogMapLt [18], ALOD2Vec [14] and Lily [23]). The evaluations are based on nine combinations of crisp reference alignments: *ra1-M1*, *ra1-M2*, *ra1-M3*, *ra2-M1*, *ra2-M2*, *ra2-M3*, *rar2-M1*, *rar2-M2* and *rar2-M3* (*ra1* is the original reference alignment; *ra2* is an extension of *ra1*; and *rar2* is an updated version of *ra2* that deals with the violations of conservativity). *ra1-M1*, *ra2-M1* and *rar2-M1* are used to evaluate alignments between classes; *ra1-M2*, *ra2-M2* and *rar2-M2* are used to evaluate alignments between properties; *ra1-M3*, ra2-M3, and rar2-M3 are used to evaluate alignments between both classes and properties.

4.3 Results and discussions

Fig. 2, Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10 show the new and previously published results on the *Conference* dataset.

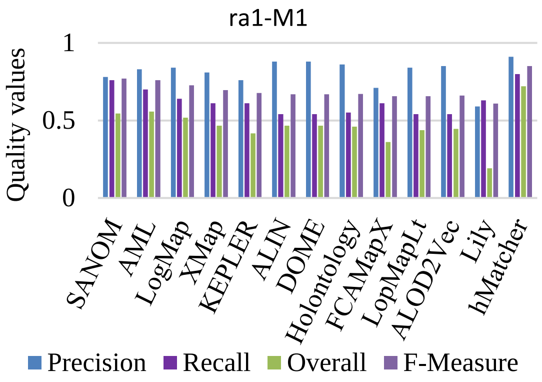

Figure. 2 ra1-M1: matching accuracy and human assistance

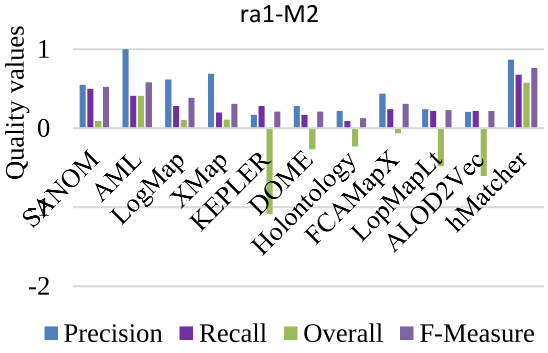

Figure. 3 ra1-M2: matching accuracy and human assistance

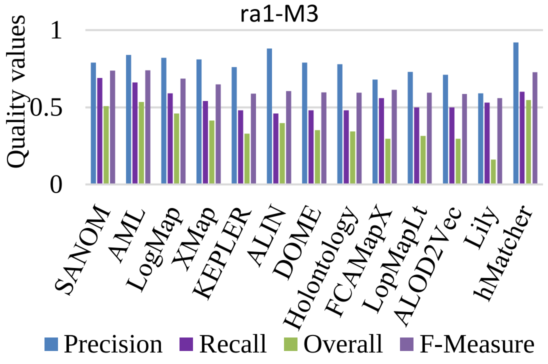

Figure. 4 ra1-M3: matching accuracy and human assistance

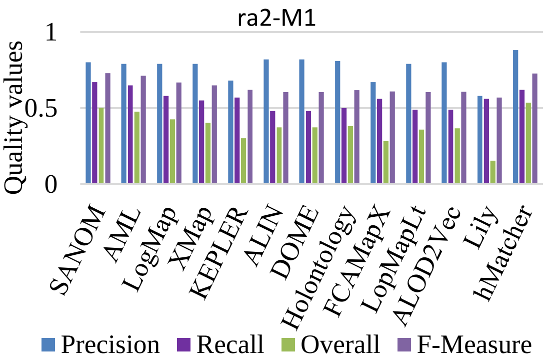

Figure. 5 ra2-M1: matching accuracy and human assistance

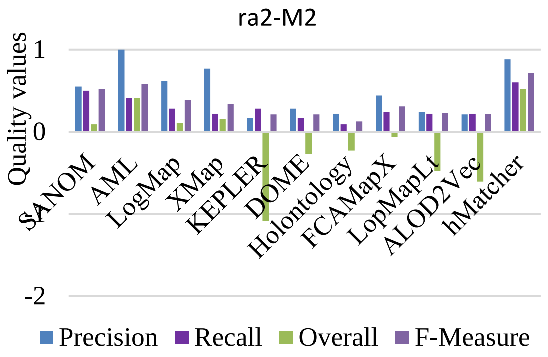

Figure. 6 ra2-M2: matching accuracy and human assistance

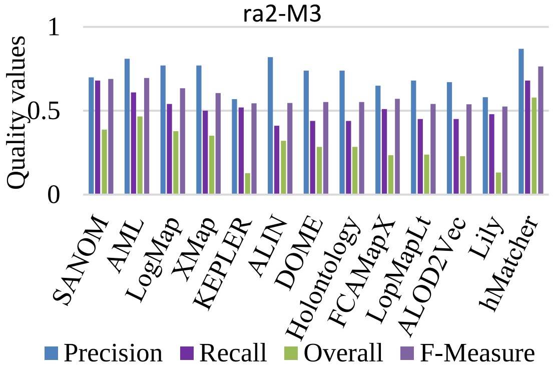

Figure. 7 ra2-M3: matching accuracy and human assistance

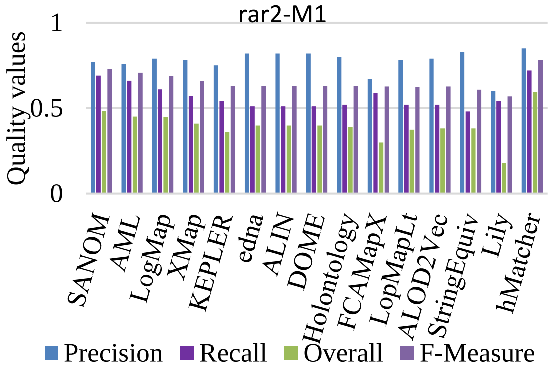

Figure. 8 rar2-M1: matching accuracy and human assistance

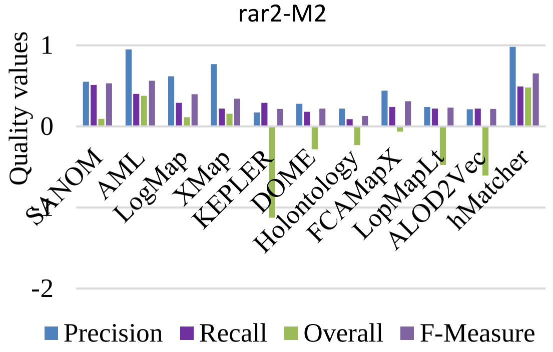

Figure. 9 rar2-M2: matching accuracy and human assistance

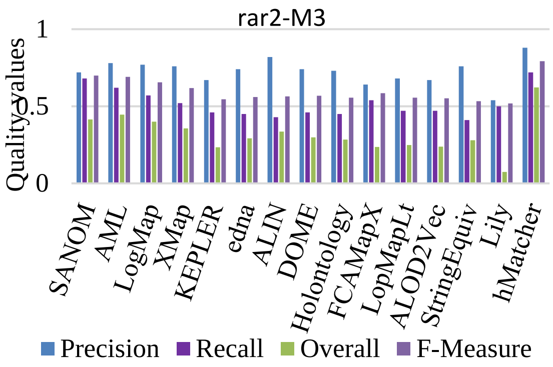

Figure. 10 rar2-M3: matching accuracy and human assistance

First, the previously published results indicate visible changes for *Precision*, *Recall*, *Overall* and *F-Measure*. They reached a high matching accuracy when evaluated based on *ra1-M1*, *ra1-M3*, *ra2-M1*, *ra2-M3*, *rar2-M1*, and *rar2-M3*; and low matching accuracy (null in some cases for example ALIN and Lily) with *ra1-M2*, *ra2-M2*, and *rar2-M2*. Second, hMatcher achieved superior matching accuracy compared to other matching approaches as it surpassed them almost every time except from *ra1-M2* and *ra2-M2* where AML surpassed it slightly (Precision = 1).

ALIN and Lily match merely classes. Because of that, they did not achieve a high matching accuracy with *ra1-M2*, *ra2-M2*, and *rar2-M2*; SANOM, XMap, AML and LogMap match few properties which justifies their negative *Overall* with *ra1-M2*, *ra2-M2*, and *rar2-M2*; FCAMapX, KEPLER, LogMapLt, DOME, ALOD2Vec and Holontology match little properties which explains their low *Precision*, *Recall* and *F-Measure* with *ra1-M2*, *ra2-M2*, and *rar2-M2*, *and* negative *Overall*; and hMatcher matches both classes and properties which explains its high *Precision*, *Recall* and *F-Measure* with *ra1-M1*, *ra1-M2*, *ra1-M3*, *ra2-M1*, *ra2-M2*, *ra2-M3*, *rar2-M1*, *rar2-M2*, and *rar2-M3*, and positive *Overall*.

Hence, we conclude that hMatcher obtains better matching results than existing matching approaches because unlike the latter that often do not match properties, hMatcher matches all elements (classes and properties). Indeed, on the one hand, we have XMap, DOME, Holontology, FCAMapX, LogMapLt, Lily, SANOM, AML, LogMap, KEPLER, ALOD2Vec and ALIN that reach good results with the reference alignments that take into consideration either classes or both classes and properties. However, they obtain lots of matching errors with the reference alignments that take into consideration merely properties. On the other hand, we have hMatcher that reaches a superior matching accuracy regardless of the reference alignments it is used with. This implies that hMatcher requires less human-assistance compared to existing matching systems that need loads of human assistance in order to correct the final matching results.

5. Conclusions and future work

We demonstrated that defining a holistic schema matching approach is critical to match multiple schemas simultaneously and generate accurate matches. The state of the art matching approaches often obtain a low matching accuracy and hence need human assistance to correct the matches; furthermore, current matching approaches often match two schemas at a time.

Let $\mathbb{S}_{Learning}$ be the learning schemas and $\mathbb{S}_{Testing}$ be the testing schemas, hMatcher first generates frequent schema elements $\mathbb{F}$ from $\mathbb{S}_{Learning}$. It then uses $\mathbb{F}$ to identify new matches Φ in $\mathbb{S}_{Testing}$. Next, hMatcher reuses previous results to determine new matches in the rare schema elements list.

We evaluated hMatcher on a real-world domain dataset, the results show a high matching accuracy achieved by hMatcher (the average metrics on nine reference alignments are: $Precision = 0.89; Recall = 0.66; Overall = 0.57$), and an inferior (compared to hMatcher) matching accuracy obtained by the state of the art matching systems (the average metrics on nine reference alignments are defined as follows: $0.69 \leq Precision \leq 0.83; 0.57 \leq Recall \leq 0.62; 0.37 \leq Overall \leq 0.49$).

Future interesting research directions include mainly the following:

- Study the impact of hMatcher on data source selection and ordering. Before the system answers users queries, it selects a subset of data sources that contain a piece of the answer or ideally the whole answer to the query, this is called source selection; then, the system orders data sources in a decreasing order of their coverage (source coverage refers to the amount of answers to a particular query included in the data source), this is called source ordering. So, in the future, we will study the impact of hMatcher on data source selection and ordering.
- Take into consideration cases where schemas are expressed using different lexical languages. We focused on schemas that use the same lexical language. In the future, we will improve our approach to match schemas regardless of the lexical language they are expressed in.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Aola Yousfi; methodology, Aola Yousfi, Moulay Hafid El Yazidi and Ahmed Zellou; software, Aola Yousfi; validation, Moulay Hafid El Yazidi and Ahmed Zellou; formal analysis, Aola Yousfi, Moulay Hafid El Yazidi and Ahmed Zellou; resources, Aola Yousfi, Moulay Hafid El Yazidi and Ahmed Zellou; data curation, Aola Yousfi, Moulay Hafid El Yazidi; writing—original draft preparation, Aola Yousfi; writing—review and editing, Aola Yousfi, Moulay Hafid El Yazidi and Ahmed Zellou; visualization, Aola Yousfi; supervision, Moulay Hafid El Yazidi and Ahmed Zellou; project administration, Aola Yousfi, Moulay Hafid El Yazidi and Ahmed Zellou.

References

[1] Y. Lee, M. Sayyadian, A. Doan, and A. Rosenthal, "etuner: tuning schema matching software using synthetic scenarios", *VLDB Journal*, Vol. 16, No. 1, pp. 97–122, 2007.

[2] E. Rahm and E. Peukert, "Holistic schema matching", *Encyclopedia of Big Data Technologies*, 2019.

[3] E. Rahm and E. Peukert, "Large-scale schema matching", *Encyclopedia of Big Data Technologies*, 2019.

[4] M. H. El Yazidi, A. Zellou, and A. Idri, "FMAMS: fuzzy mapping approach for mediation systems", *IJAEC*, Vol. 4, No. 3, pp. 34–46, 2013.

[5] M. H. El Yazidi, A. Zellou, and A. Idri, "Fgav (fuzzy global as views)", In: *Proc. of AIP Conf.*, Vol. 1644, No. 1, pp. 236–243, 2015.

[6] E. Rahm and P. A. Bernstein, "On matching schemas automatically", *VLDB Journal*, Vol. 10, No. 4, pp. 334–350, 2001.

[7] E. Sutanta, R. Wardoyo, K. Mustofa, and E. Winarko, "Survey: Models and prototypes of schema matching", *International Journal of Electrical and Computer Engineering*, Vol. 6, No. 3, pp. 2088-8708, 2016.

[8] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches", *Journal on data semantics*, Vol. 4, pp. 146–171, 2005.

[9] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching", *VLDB Journal*, Vol. 10, No. 4, pp. 334–350, 2001.

[10] J. Köpke, "Annotation paths for matching xml-schemas", *Data Knowledge Engineering*, Vol. 122, pp. 25–54, 2019.

[11] P. Roussille, I. Megdiche, O. Teste, and C. Trojahn, "Holontology: results of the 2018 OAEI evaluation campaign", In: *Proc. of the 13th International Workshop on Ontology Matching Co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 167–172, 2018.

[12] K. Saleem, Z. Bellahsene, and E. Hunt, "PORSCHE: performance oriented schema mediation", *Information Systems*, Vol. 33, No. 7-8, pp. 637–657, 2008.

[13] J. da Silva, K. Revoredo, and F. A. Baião, "ALIN results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 117–124, 2018.

[14] J. Portisch and H. Paulheim, "Alod2vec matcher", In: *Proc. of the 13th International Workshop on Ontology Matching Co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 132–137, 2018.

[15] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system", In: *Proc. of On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conf.: CoopIS, DOA-Trusted Cloud, and ODBASE*, Graz, Austria, pp. 527–541, 2013.

[16] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: Efficient matching for large real-world schemas and ontologies", *PVLDB*, Vol. 2, No. 2, pp. 1586–1589, 2009.

[17] S. Hertling and H. Paulheim, "DOME results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 144–151, 2018.

[18] E. Jiménez-Ruiz, B. C. Grau, and V. Cross. "Logmap: family participation in the OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 187–191, 2018.

[19] G. Chen and S. Zhang, "Fcamapx: results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 160–166, 2018.

[20] M. Kachroudi, G. Diallo, and S. Ben Yahia, "KEPLER at OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 173–178, 2018.

[21] M. Mohammadi, W. Hofman, and Y.-H. Tan, "SANOM results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching Co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 205–209, 2018.

[22] M. Mohammadi, W. Hofman, and Y.-H. Tan, "Simulated annealing-based ontology matching", *ACM Trans. Management Inf. Syst.*, Vol. 10, No. 1, pp. 1-24, 2019.

[23] Y. Tang, P. Wang, Z. Pan, and H. Liu, "Lily results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 179–186, 2018.

[24] P. Wang and B. Xu, "An effective similarity propagation model for matching ontologies without sufficient or regular linguistic information", In: *Proc. of the 4th Asian Semantic Web Conf*. (ASWC2009), Shanghai, China, 2009.

[25] P. Wang and B. Xu, "Debugging ontology mappings: A static approach", *Computing and Informatics*, Vol. 27, No. 1, pp. 21–36, 2008.

[26] P. Yang, P. Wang, L. Ji, X. Chen, K. Huang, and B. Yu, "Ontology matching tuning based on particle swarm optimization: Preliminary results", In: *Proc. of the Semantic Web and Web Science - 8th Chinese Conf.*, Wuhan, China, pp. 146–155, 2014.

[27] I. Megdiche, O. Teste, and C. T. dos Santos, "LPHOM results for OAEI 2016", In: *Proc. of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conf.*, Kobe, Japan, pp. 190–195, 2016.

[28] Megdiche, O. Teste, and C. T. dos Santos, "An extensible linear approach for holistic ontology matching", In: *Proc. of The Semantic Web - ISWC 2016 - 15th International Semantic Web Conf.*, Kobe, Japan, Part I, pp. 393–410, 2016.

[29] W. E. Djeddi, S. Ben Yahia, and M. T. Khadir, "Xmap: results for OAEI 2018", In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 210–215, 2018.

[30] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification", *WordNet: An electronic*

Lexical Database, Vol. 49, No. 2, pp. 265–283, 1998.

[31] Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology”, *Nucleic Acids Research*, Vol. 32, No. 1, pp. 267–270, 2004.

[32] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao, “Reducing uncertainty of schema matching via crowdsourcing”, *PVLDB*, Vol. 6, No. 9, pp.757–768, 2013.

[33] C. J. Zhang, L. Chen, H. V. Jagadish, M. Zhang, and Y. Tong, “Reducing uncertainty of schema matching via crowdsourcing with accuracy rates”, *IEEE Trans. Knowledge Data Engineering*, Vol. 32, No. 1, pp. 135–151, 2020.

[34] M. H. El Yazidi, A. Zellou, and A. Idri, “Towards a fuzzy mapping for systems”, In: *Proc. of IEEE International Conf. on Complex Systems (ICCS),* pp. 1-4, 2012.

[35] A. Yousfi, M. H. El Yazidi, and A. Zellou, “Assessing the performance of a new semantic similarity measure designed for schema matching for mediation systems”, In: *Proc. of the Computational Collective Intelligence - 10th International Conf.*, Bristol, UK, Part I, pp. 64–74, 2018.

[36] L. Havrlant and V. Kreinovich, “A simple probabilistic explanation of term frequency inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)”, *International Journal General Systems*, Vol. 46, No. 1, pp. 27–36, 2017.

[37] D. Faria, C. Pesquita, B. S. Balasubramani, T. Tervo, D. Carriço, R. Garrilha, F. M. Couto, and I. F. Cruz, “Results of AML participation in OAEI 2018”, In: *Proc. of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conf.*, Monterey, CA, USA, pp. 125–131, 2018.