xMatcher: Matching Extensible Markup Language Schemas using Semantic-based Techniques

Aola Yousfi[1], Moulay Hafid El Yazidi[2], Ahmed Zellou[3]
ENSIAS, Mohammed V University
in Rabat, Morocco

***Abstract*—Schema matching is a critical step in data integration systems. Most recent schema matching systems require a manual double-check of the matching results to add missed matches and remove incorrect matches. Manual correction is labor-intensive and time-consuming, however without it the results accuracy is significantly lower. In this paper, we present xMatcher, an approach to automatically match XML schemas. Given two schemas S_1 and S_2, xMatcher identifies semantically similar schema elements between S_1 and S_2. To obtain correct matches, xMatcher first transforms S_1 and S_2 into sets of words; then, it uses a context-based measure to identify the meanings of words in their contexts; next, it captures semantic relatedness between sets of words in different schemas; finally, it uses WordNet information to calculate the similarity values between semantically related sets and matches the pairs of sets whose similarity values are greater than or equal to 0.8. The results show that xMatcher provides superior matching accuracy compared to the state of the art matching systems. Overall, our proposal can be a stepping stone towards decreasing human assistance and overcoming the weaknesses of current matching initiatives in terms of matching accuracy.**

***Keywords*—Schema matching; matching accuracy; semantic similarity; semantic relatedness; WordNet**

I. Introduction

A. Motivation and Background

Schema matching aims at identifying semantic correspondences called matches [1], [2] in multiple schemas. It is critical for applications that manipulate data across different data sources because - if done correctly - it gives the end user a unified view over sources. We use an example to illustrate the schema matching problem. Let S_1 (Listing 1) and S_2 (Listing 2) be two XML schemas describing academic conferences. Our goal is to identify the matches in Fig. 1.

Although it is often desirable to define manually an integrated schema that represents all sources, this is often impossible for two main reasons: (1) the huge number of sources; and (2) the continuous updates. Thus, plenty of *automatic* schema matching systems have been developed (we refer the reader to [3], [4], [5], [6] for recent surveys and some state of the art matching systems). However, the term *automatic* is quite relative because even when humans do not help during the matching process, they help at the end correcting the results: adding missed matches and removing erroneous matches. Therefore, improving the accuracy of the output matches can significantly reduce humans' workload, and avoid possible mistakes humans might make. Also, it can save a considerable amount of time by leaving merely few results to correct.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="conference">
 <xs:complexType>
   <xs:element name="conference_name" type="xs:string"/>
   <xs:element name="publication">
    <xs:complexType>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="author_name" type="xs:string"/>
    </xs:complexType>
   </xs:element>
 </xs:complexType>
</xs:element>
</xs:schema>
```

Listing 1: S_1

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="conference">
 <xs:complexType>
   <xs:element name="name" type="xs:string"/>
   <xs:element name="paper">
    <xs:complexType>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="author_name" type="xs:string"/>

    </xs:complexType>
   </xs:element>
 </xs:complexType>
</xs:element>
</xs:schema>
```

Listing 2: S_2

conference.conference_name <=> conference.name
conference.publication.title <=> conference.paper.title
conference.publication.author_name <=> conference.paper.author_name

Fig. 1. Matches between S_1 and S_2

Furthermore, the state of the art schema matching systems often reach a very moderate (sometimes poor) matching accuracy [2], and require loads of manual assistance to help correct the matching results [2]. In this paper, we will introduce a new schema matching system that will overcome these limitations as it is designed to achieve a high matching accuracy without any human assistance.

B. Challenges

Valuable as it is, producing high accuracy matches is also very difficult. First, schemas often use different naming conventions, e.g. *conference_name* (see Listing 1) and *name* (see Listing 2), or totally different words, e.g. *publication* (see Listing 1) and *paper* (see Listing 2). Second, schema elements are not fully independent from each other. For example, nested

elements in XML schemas. Third, a word can have multiple meanings. Finally, given a word W, WordNet hierarchy [7] connects W to other words through a wide variety of relations (e.g. hypernyms, hyponyms, meronyms); contributing unevenly to the definition of W. For example, according to WordNet the word *conference* has a direct hypernym (*meeting*) and five direct hyponyms (*symposium, seminar, colloquium, Potsdam conference*, and *Yalta conference*), both combined provide a comprehensive definition than one of them combined with *conference*'s meronym (*conferee*).

C. Contributions

In this paper, we introduce xMatcher, an approach to automatically match XML schemas. The key idea of xMatcher is to match XML schemas based on their semantics and with the objective of obtaining high accuracy matches, which reduces considerably humans' workload and offers a reliable and unified view over a large number of data sources. In particular, we make the following contributions:

- We propose a context-based measure to determine the meanings of words according to their contexts.
- We propose an automatic strategy to capture semantic relatedness between sets of words in different schemas.
- We present a semantic similarity measure over WordNet to calculate the semantic similarity between semantically related sets of words.
- We evaluate our similarity measure on a popular dataset and show that it provides correct results and surpasses the state of the art semantic measures and distances.
- We evaluate xMatcher on different real-world domains and show that it produces high accuracy matches and outperforms the state of the art systems in terms of matching accuracy.

The rest of this paper is organized as follows. Section II first reviews the state of the art schema and ontology matching systems, then it presents the state of the art similarity measures. Section III defines the problem of schema matching. Section IV describes xMatcher. Section V evaluates both our similarity measure and xMatcher in terms of matching accuracy. Section VI concludes this paper and discusses future work.

II. Related Work

A. Schema and Ontology Matching Systems

Although it is not in its infancy, schema and ontology matching still an active research area. Indeed, the number of approaches available for schema and ontology matching increases continuously (we refer the reader to [3], [4], [5], [6], [8] for recent surveys and some existing matching systems). Also, the number of matching systems participating in the Ontology Alignment Evaluation Initiative (OAEI[1]) is increasing significantly. Before we proceed with the description of our new matching system xMatcher, we first review the state of the art matching systems that use WordNet as the matching space (e.g. ALIN [9]), and the top matching systems that participated in the 2018 edition of OAEI (e.g. Holontology [10], DOME [11], ALOD2Vec [12], and AgreementMakerLight [13]).

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

Holontology [10] is a modular holistic ontology matching system based on the Linear Program for Holistic Ontology Matching (LPHOM) system. It uses a combination of several similarity measures: Levenstein, Jaccard, and Lin to match two ontologies or multiple ontologies at once after it converts them into an internal predefined format. Then, Holontology transforms the results into alignments exported by RDF.

ALIN [9] is an interactive ontology matching system which takes as input two ontologies and deliver as output a set of alignments between them. It proceeds in two major steps. (1) It generates the initial mappings. (2) It waits for the human expert feedback and changes the mappings accordingly in order to improve the accuracy of the final results. This step is repeated until the human expert has no more mapping suggestions.

DOME (Deep Ontology MatchEr) [11] is a scalable matcher which uses doc2vec and exploits large texts that describe the concepts of the ontologies. To deal with the main issue of matching similar large texts, DOME uses topic modelling such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA).

ALOD2Vec [12] uses as external background knowledge source the WebIsALOD database of hypernym relations extracted from the Web. It also exploits element-based information and label-based information. In order to determine the similarity score between nodes of the knowledge graph (WebIsALOD is viewed as a knowledge graph), ALOD2Vec applies RDF2Vec.

AgreementMakerLight (AML) [13] is an ontology matching system which derives from AgreementMaker [14]. AML consists of two main modules: the ontology loading module and the ontology matching module. The ontology loading module loads the ontology files along with the external resources and then generates the ontology objects. The ontology matching module main goal is to align the ontology objects generated. The ontology loading module is extensible as it allows the virtual integration of new matching algorithms.

The matching systems presented above achieve acceptable results. The goal of this paper is to surpass the aforementioned systems in terms of *Precision*, *Recall*, *Overall*, and *F-Measure* (we refer the reader to subsection V-A for a definition of these quality metrics).

B. Semantic Similarity

1) Similarity Measures and Distances: One of the many possible approaches to discover matches is to compute the semantic similarity values between schema elements which is the approach we adopted for our matching system. Semantic similarity measures are one of the biggest pressing challenges facing the improvement of schema matching. According to [15], [16], [17], [18], [19], [20], [21], [22], semantic similarity measures are grouped into four categories: edge-based measures, information content-based measures, feature-based measures, and hybrid-based measures.

- **Edge-based measures (also known as path-based measures).** They determine the similarity between two concepts by considering both the length of the path that links the concepts in the taxonomy and the position of the

concepts in the taxonomy [15], [16], [18], [20]. Examples include the shortest path-based measure [15].
- **Information content-based measures.** The main idea of these measures is that the more information two concepts have in common, the more semantically similar the concepts are [15], [23]. Examples include Resnik [24], Jiang & Conrath [25], Lin [26], and Nababteh [27].
- **Feature-based measures.** They use the properties of the concepts in a way that the more common features two concepts have and the less non-common features they have, the more semantically similar the two concepts are [15], [16], [18], [20], e.g. Tversky [28].
- **Hybrid-based measures.** They combine all the three aforementioned categories [16], [18], [20]. Zhou's measure is an example of hybrid-based measures [29].

But since the information content-based measures perform better than other categories (information content-based measures have the highest correlation coefficients when compared to the matching results provided by human experts) [15], we decided to direct our attention to the aforementioned information content-based measures that we will compare later to our semantic similarity measure.

WordNet [7] is a lexical database for the English language created by a research team at Princeton University. It groups words into sets of synonyms called *synsets*, which are interlinked by means of semantic relationships, for instance, ***is-a*** relationship which connects a hyponym to a hypernym. And it is commonly used by semantic similarity measures. Indeed, the following measures all use WordNet as an external resource.

Resnik's measure [24] computes the Information Content (IC) of the Least Common Subsumer (LCS) of two concepts denoted by a and b as follows:

$$Sim_{Resnik}(a,b) = IC(LCS(a,b)) \tag{1}$$

Where:

- Given a concept C, we have IC(C)= - log(p(C)).
- $p(C) = \frac{frequency(C)}{N}$ refers to the probability of C.
- N refers to the total number of nouns.

The main issue of Resnik's measure is the following: any pair of concepts having the same LCS will definitely have the same semantic similarity value [15]. Luckily, Jiang & Conrath (J&C) and Lin found out a way to overcome Resnik's problem [25], [26]. In addition to the IC of the LCS, both J&C and Lin consider the IC of each concept [25], [26]. J&C define the distance between two concepts as follows [25]:

$$Dis_{J\&C}(a,b) = IC(a) + IC(b) - 2 \times IC(LCS(a,b)) \tag{2}$$

It differs from similarity measures in a way that the higher it gets, the less similar the two compared concepts are. Typically, given J&C's distance, one can revert it to serve as a similarity measure and vice versa. Conversions are made using equation 3. In this paper, we are going to use the similarity measure.

$$Sim_{J\&C}(a,b) = \begin{cases} 1, & \text{if } Dis_{J\&C}(a,b) = 0 \\ \frac{1}{Dis_{J\&C}(a,b)}, & \text{otherwise} \end{cases} \tag{3}$$

Lin describes the semantic similarity between two concepts as follows [26]:

$$Sim_{Lin}(a,b) = \frac{2 \times IC(LCS(a,b))}{IC(a) + IC(b)} \tag{4}$$

The main issue with Lin's measure is the following: if the IC of LCS, a, or b is equal to 0 then the semantic similarity value is equal to 0 as well [27].

In order to deal with Lin's problem, Nababteh suggests to divide 2 times the IC of the LCS of the two compared concepts by the sum of the IC of the direct hypernym of the first concept and the IC of the direct hypernym of the second concept [27].

$$Sim_{Nababteh}(a,b) = \frac{2 \times IC(LCS(a,b))}{IC(P(a)) + IC(P(b))} \tag{5}$$

For the time being, the aforementioned semantic similarity measures are quite successful, they remain, however, some issues that require more attention. Indeed, according to [24], [25], [26], [27], the aforementioned measures might not provide the correct results all the time since when compared to the reference similarity values on Miller and Charles' (M&C) benchmark dataset the results were not promising.

2) Schema-Based Information and Instance-based Information: The Rivalry to Dominate Schema Matching: One of the most important choices that impacts the accuracy of the results returned by a similarity measure used by a schema matching system is the information used to find out semantic correspondences between schemas. Besides the external resources, a similarity measure may utilize either schema-based information, instance-based information, or both. In Table I, we present the advantages and disadvantages of each approach.

TABLE I. PROS AND CONS OF SCHEMA- AND INSTANCE-BASED PRACTICES

	Advantages	**Disadvantages**
Schema-based approach	- It uses the properties of the schema elements (e.g. labels, data types, integrity constraints). - Easy to implement. - They are fast.	- It does not produce good results when the properties of the schema elements are not available.
Instance-based approach	- It exploits the data stored at a given time which provides more details about the schema elements and hence improves the accuracy of the final results.	- Unavailable data may cause the matching system to stop functioning properly and exit. - Incorrect data may lead to false matches or miss true matches. - They operate slowly. - More complicated to implement than schema-based approaches.

Based on the information presented in Table I, we decided to use schema-based information to define our solution.

III. PROBLEM STATEMENT

In this section, we present definitions related to the schema matching problem. In this paper, we consider only XML schemas and leave other data representations for future work.

Definition 1 (Entity). *Let S be an XML schema. An entity e is used interchangeably to refer to a complex type element, a simple type element, or an attribute.*

Definition 2 (Set of Words). *Let S be an XML schema and n be the number of entities ($e_1, e_2, \ldots, e_n$) it contains. Given an entity $e_1 \in S$, the set of words generated from e_1 is defined as follows $set_{e_1} = \{W_{1,1}, W_{1,2}, \ldots, W_{1,card(set_{e_1})}\}$, where $W_{1,1}, W_{1,2}, \ldots, W_{1,card(set_{e_1})}$ are words extracted from e_1.*

Remark: All the sets of words generated from S are defined as follows $SETS = \{set_{e_1}, set_{e_2}, \ldots, set_{e_n}\}$.

Definition 3 (Semantic Relatedness). *Let S_1 and S_2 be two schemas, and $SETS_1$ and $SETS_2$ be their respective sets of words. $set_1 \in SETS_1$ and $set_2 \in SETS_2$ are semantically related if they can be used together in the same schema. For example {conference, paper, title} and {conference, paper, author} from Listing 2 are semantically related.*

Definition 4 (Semantic Similarity). *Let S_1 and S_2 be two schemas, and $SETS_1$ and $SETS_2$ be their respective sets of words. $set_1 \in SETS_1$ and $set_2 \in SETS_2$ are semantically similar if they share the same meaning. Also, semantically similar sets cannot be used together in the same schema. For example {conference, publication, title} in Listing 1 and {conference, paper, title} in Listing 2 are semantically similar.*

Remark: Let S_1 and S_2 be two schemas, and $SETS_1$ and $SETS_2$ be their respective sets of words. If $set_1 \in SETS_1$ and $set_2 \in SETS_2$ are semantically similar then they are semantically related as well, e.g. {*conference, publication, title*} and {*conference, paper, title*}. However $set_1 \in SETS_1$ and $set_2 \in SETS_2$ are semantically related does not necessarily imply that they are similar, e.g. {*conference, paper, title*} and {*conference, paper, author*}.

Definition 5 (Problem Statement). *Given n schemas S_1, S_2, ..., S_n. Our goal is to maximize the accuracy of the matches discovered between S_1, S_2, ..., S_n and minimize humans' workload traditionally used to correct the matching results.*

Table II lists the notations used throughout this paper.

TABLE II. SUMMARY OF SYMBOL NOTATIONS

Notation	Description
S, e, c	XML schema, *entity*, complex type element
W, DB_{abbr}	Word from WordNet entries, abbreviations database
set, $SETS$, $SETS'$	set of words, sets of words, semantically related pairs of word sets
$set_{e_{WordNet}}$, $set_{e_{abbreviations}}$, $set_{e_{expression}}$	set of WordNet entries that correspond to words in e, set of abbreviations database entries that correspond to words in e, set of full expressions of abbreviations contained in $set_{e_{abbreviations}}$
SM, Sim, Dis	sub-measure, similarity measure, distance
F, M	relatedness matrix, similarity matrix
$card(set)$	cardinality of set

In the next section, we introduce xMatcher the solution to the schema matching problem described in Definition 5.

IV. THE XMATCHER APPROACH

The xMatcher architecture (see Fig. 2) consists of three main modules: pre-matching, matching, and post-matching. Given two XML schemas S_1 and S_2, the *pre-matching module* ($\mu : S_1 \times S_2 \rightarrow SETS_1 \times SETS_2$) uses WordNet along with a database of abbreviations and applies fuzzy string matching to generate, from each *entity* in S_1 and S_2, a set of words. The *matching module* ($\phi : SETS_1 \times SETS_2 \rightarrow [0, 1]$) then identifies semantically related sets, for which it calculates the similarity values. Finally, the *post-matching module* ($\theta : [0, 1] \rightarrow Matches$) matches the *entities* whose similarity values are greater than or equal to 0.8. It is important to note that all three modules take place prior to any user request.

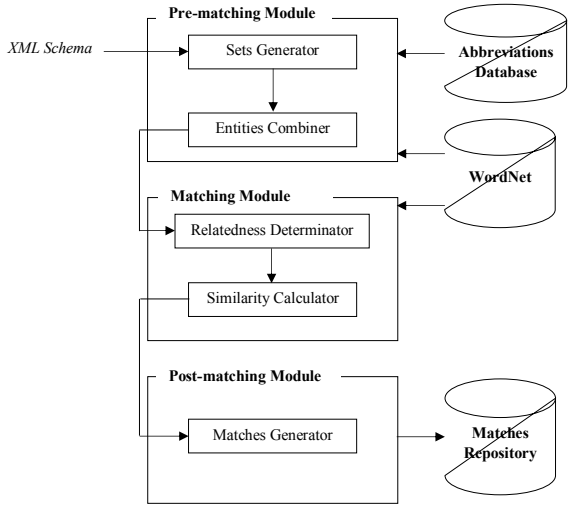

Fig. 2. The xMatcher Architecture

The rest of this section describes the pre-matching module (see subsection IV-A), the matching module (see subsection IV-B), and the post-matching module (see subsection IV-C).

A. The Pre-Matching Module

Before we proceed with the matching module, a pre-matching step is required since schemas use different naming conventions. The *entity* name might be an expression that does not belong to WordNet. Examples of such non-WordNet *entities* include abbreviations, concatenation of words, and words separated by underscores. Thus, we use two components, the *sets generator* and the *entities combiner*, to produce, for each *entity*, a set of words that help clarify its meaning.

Sets generator. Given a non-WordNet *entity* e, the sets generator proceeds in three steps (see Algorithm 1). (1) It uses fuzzy string matching to extract from e words $set_{e_{WordNet}}$ that syntactically correspond exactly or approximately to WordNet entries. (2) The sets generator then uses fuzzy string matching to see if $e \setminus set_{e_{WordNet}}$ includes abbreviations $set_{e_{abbreviations}}$ that correspond exactly or approximately to DB_{abbr}, in which case it substitutes $set_{e_{abbreviations}}$ for their full expression $set_{e_{expression}}$ available in DB_{abbr}. (3) It assigns a set of words to e, such that $set_e = set_{e_{WordNet}} \cup set_{e_{expression}}$.

Entities combiner. Let c be a complex type element, e be a non-complex type element included in c, and $set_c = set_{c_{WordNet}} \cup set_{c_{expression}}$ and $set_e = set_{e_{WordNet}} \cup set_{e_{expression}}$ be their respective sets of words. We made the following observation: the more words set_e contains, the more meaning e conveys. Therefore, we decided to utilize the context of e, which is the complex elements e belongs to, as follows $set_e \leftarrow set_e \cup set_c$. Algorithm 2 summarizes this.

Next, we use the sets of words to match schemas using relatedness matrices and a semantic similarity measure.

B. The Matching Module

The matching module consists of two major components: *relatedness determinator* and *similarity calculator*. The relatedness determinator uses relatedness matrices to capture

TABLE III. RELATEDNESS MATRIX

	W_{1_1}	W_{1_2}	...	$W_{1_{card(set_{e_1})}}$
W_{2_1}	$f_{1,1}$	$f_{1,2}$	...	$f_{1,card(set_{e_1})}$
W_{2_2}	$f_{2,1}$	$f_{2,2}$	...	$f_{2,card(set_{e_1})}$
...	...	...	...	...
$W_{2_{card(set_{e_2})}}$	$f_{card(set_{e_2}),1}$	$f_{card(set_{e_2}),2}$	...	$f_{card(set_{e_2}),card(set_{e_1})}$

Algorithm 1 SetsGenerator(S)

Input:
S
Output:
$SETS$

```
for each e in S do
    for each W ∈ WordNet in e do
        set_e ← W
    end for
    for each abbr ∈ DB_abbr in e do
        Substitute abbr for its full expression
        Add its full expression to set_e
    end for
end for
return SETS
```

Algorithm 2 EntitiesCombiner(S, $SETS$)

Input:
S
$SETS$
Output:
$SETS$

```
for each e in S do
    for each c containing e do
        set_e ← set_e ∪ set_c
    end for
end for
return SETS
```

semantic relatedness between different sets of words. Then, the similarity calculator exploits WordNet hierarchy to calculate the similarity between every semantically related sets.

1) Generating relatedness matrices: Prior to computing the semantic similarity values between different sets of words, we first must identify semantically related sets. This is very important for two main reasons. First, it narrows down the total number of computations, since we will only calculate the semantic similarity values between related sets. Second, let $e_1 \in S_1$ and $e_2 \in S_2$ be two *entities*, and set_{e_1} and set_{e_2} be their respective sets of words. Let's suppose that both e_1 and e_2 are not contained in any complex type element. Missing contexts implies that set_{e_1} and set_{e_2} convey poor meanings. Thus, identifying whether they are semantically related or not will help improve considerably their meanings. To this end, the relatedness determinator proceeds in two steps (Algorithm 3 summarizes this). First, it uses equation (6) to determine the meaning of a word according to the other words in the same set. Second, it employs fuzzy string matching and words synonyms available in WordNet to identify semantically related sets. In the following, we explain these steps in more details.

Step 1: Identifying meanings of words. Let e be an *entity* and set_e be its set of words. Given that a word $W \in set_e$ may have more than one meaning, we use $set_e \setminus W$ to identify the meanings of W.

$$\forall\ W \in set_e,\ Sense(W) = \max_{1 \le i \le n} \sum_{j=1}^{card(set_e \setminus W)} \sum_{k=1}^{n_j} relatedness(s_i, s_{j,k}) \quad (6)$$

Where:

- s_i and $s_{j,k}$ are the i^{th} sense of W (meaning of W in WordNet) and the k^{th} sense of the j^{th} word in $set_e \setminus W$, respectively.
- n and n_j are the total number of senses of W and the total number of senses of the j^{th} word in $set_e \setminus W$, respectively.
- $relatedness$ returns the number of overlapping phrases or words between s_i and $s_{j,k}$.

Step 2: Identifying semantically related sets of words. Let $e_1 \in S_1$ and $e_2 \in S_2$ be two *entities* and $set_{e_1} = \{W_{1,1}, W_{1,2}, \ldots, W_{1,card(set_{e_1})}\} \in SETS_1$ and $set_{e_2} = \{W_{2,1}, W_{2,2}, \ldots, W_{2,card(set_{e_2})}\} \in SETS_2$ be their respective sets of words. We use fuzzy string matching to determine the words contained in both set_{e_1} and set_{e_2}. We display the results in a relatedness matrix $F = (f_{i,j})_{\substack{1 \le i \le card(set_{e_1}) \\ 1 \le j \le card(set_{e_2})}}$ (see Table III) whose individual items are defined as follows: $f_{i,j} = (o_{1_{i,j}}, o_{2_{i,j}})$, where $o_{1_{i,j}}$ is equal to 1 if W_{1_j} or one of its synonyms and W_{2_i} or one of its synonyms appear together in set_{e_1}, and 0 otherwise. Similarly, $o_{2_{i,j}}$ is equal to 1 if W_{1_j} or one of its synonyms and W_{2_i} or one of its synonyms appear together in set_{e_2}, and 0 otherwise.

Remark: $\forall \{i,j\} \in [\![1, card(set_{e_1})]\!] \times [\![1, card(set_{e_2})]\!]$. If W_{1_j} and W_{2_i} refer to the same word then $o_{1_{i,j}} = o_{2_{i,j}} = 1$.

We generated relatedness matrices for different real-world schemas (*Airfare*, *Automobiles*, *Books*, *Car Rentals*, *Hotels*, *Jobs*, *Movies*, and *Music Records*) extracted from the Web interfaces in the TEL dataset of the UIUC Web Integration Repository[2]. We noticed that semantically related sets (provided manually) are assigned matrices that contain more ones than zeros. Thus, we made the following conclusion: we say that two sets are semantically related if and only if the occurrence of 1 in F is greater than the occurrence of 0.

[2]http://metaquerier.cs.uiuc.edu/repository

Algorithm 3 RelatednessDeterminator($SETS_1$, $SETS_2$)

Input:
$SETS_1$, $SETS_2$
Output:
$SETS'$

1: **for** each W in $set_1 \in SETS_1$ **do**
2: Identify the meaning of W using equation (6) */*Similarly, we identify the meanings of words in $SETS_2$*/*
3: **end for**
4: **for** each set_1 in $SETS_1$ **do**
5: **for** each set_2 in $SETS_2$ **do**
6: Determine semantically related sets based on their relatedness matrix
7: Add semantically related sets to $SETS'$
8: **end for**
9: **end for**
10: **return** $SETS'$

Next, we calculate the similarity between semantically related sets of words.

2) Calculating similarity values between entities: The similarity calculator operates in two steps (see Algorithm 4). First, it calculates the similarity between words. Then, it uses the results to calculate the similarity between sets of words.

Algorithm 4 SimilarityCalculator($SETS'_1$, $SETS'_2$)

Input:
$SETS'_1$
$SETS'_2$
Output:
V */*Similarity values between sets of $SETS'_1$ and sets of $SETS'_2$*/*

1: **for** each set_1 in $SETS'_1$ **do**
2: **for** each set_2 in $SETS'_2$ **do**
3: Calculate the similarity v between set_1 and set_2 using equation (17)
4: $V \leftarrow V \cup v$
5: **end for**
6: **end for**
7: **return** V

Step 1: Calculating the semantic similarity between words. Given a word $W \in WordNet$, we noticed that both its hypernyms and its direct hyponyms can be used together to define it. Hence, we decided to utilize this information to determine how similar two words are. Given two words $a, b \in WordNet$, comparing a to b is equivalent to comparing $\{a,\ P_a,\ H_a\}$ to $\{b,\ P_b,\ H_b\}$. Thus, the similarity calculator calculates the similarity between a and b (7), a and P_b (8), a and H_b (9), P_a and b (10), P_a and P_b (11), P_a and H_b (12), H_a and b (13), H_a and P_b (14), and H_a and H_b (15). P_a and P_b refer to the hypernyms of a and b, respectively. H_a and H_b refer to the direct hyponyms of a and b, respectively. Note that we consider only non-shared hypernyms hence $P_a \cap P_b = \phi$.

$$SM_1(a,b) = card(s_a \cap s_b) + card(s_a \cap (b \cup Sy_b)) + card(s_b \cap (a \cup Sy_a)) \quad (7)$$

$$SM_2(a,P_b) = \sum_{i=1}^{|P_b|} card(s_a \cap s_{P_{b_i}}) + card(s_a \cap (P_{b_i} \cup Sy_{P_{b_i}})) + card(s_{P_{b_i}} \cap (a \cup Sy_a)) \quad (8)$$

$$SM_3(a,H_b) = \sum_{i=1}^{|H_b|} card(s_a \cap s_{H_{b_i}}) + card(s_a \cap (H_{b_i} \cup Sy_{H_{b_i}})) + card(s_{H_{b_i}} \cap (a \cup Sy_a)) \quad (9)$$

$$SM_4(P_a,b) = \sum_{i=1}^{|P_a|} card(s_{P_{a_i}} \cap s_b) + card(s_{P_{a_i}} \cap (b \cup Sy_b)) + card(s_b \cap (P_{a_i} \cup Sy_{P_{a_i}})) \quad (10)$$

$$SM_5(P_a,P_b) = \sum_{i=1}^{|P_a|}\sum_{j=1}^{|P_b|} card(s_{P_{a_i}} \cap s_{P_{b_j}}) + card(s_{P_{a_i}} \cap (P_{b_j} \cup Sy_{P_{b_j}})) + card((P_{a_i} \cup Sy_{P_{a_i}}) \cap s_{P_{b_j}}) \quad (11)$$

$$SM_6(P_a,H_b) = \sum_{i=1}^{|P_a|}\sum_{j=1}^{|H_b|} card(s_{P_{a_i}} \cap s_{H_{b_j}}) + card(s_{P_{a_i}} \cap (H_{b_j} \cup Sy_{H_{b_j}})) + card(P_{a_i} \cup Sy_{P_{a_i}}) \cap s_{H_{b_j}}) \quad (12)$$

$$SM_7(H_a,b) = \sum_{i=1}^{|H_a|} card(s_{H_{a_i}} \cap s_b) + card(s_{H_{a_i}} \cap (b \cup Sy_b)) + card(s_b \cap (H_{a_i} \cup Sy_{H_{a_i}})) \quad (13)$$

$$SM_8(H_a,P_b) = \sum_{i=1}^{|H_a|}\sum_{j=1}^{|P_b|} card(s_{H_{a_i}} \cap s_{P_{b_j}}) + card(s_{H_{a_i}} \cap (P_{b_j} \cup Sy_{P_{b_j}})) + card((H_{a_i} \cup Sy_{H_{a_i}}) \cap s_{P_{b_j}}) \quad (14)$$

$$SM_9(H_a,H_b) = \sum_{i=1}^{|H_a|}\sum_{j=1}^{|H_b|} card(s_{H_{a_i}} \cap s_{H_{b_j}}) + card(s_{H_{a_i}} \cap (H_{b_j} \cup Sy_{H_{b_j}})) + card((H_{a_i} \cup Sy_{H_{a_i}}) \cap s_{H_{b_j}}) \quad (15)$$

Where s_a refers to the sense of a and Sy_a refers to the *synset* (set of synonyms) of a.

We applied our measure (16) on M&C benchmark dataset several times, each time with a different combination of $SM_{1 \leq i \leq 9}$ (Given two parameters $\alpha, \beta \in [0,1]$, $[\alpha \times \sum_{i=1}^{9} SM_i = \beta \times \sum_{i=1}^{9} SM_i = \sum_{i=1}^{9} SM_i$, (Where $\alpha = \beta = 1$)], $[\alpha \times SM_1 = 0.1 \times SM_1$ and $\beta \times \sum_{i=2}^{9} SM_i = 0.9 \times \sum_{i=2}^{9} SM_i]$, $[\alpha \times SM_2 = 0.1 \times SM_2$ and $\beta \times \sum_{\substack{i=1 \\ i \neq 2}}^{9} SM_i = 0.9 \times \sum_{i=1}^{9} SM_i]$ etc.). We then calculated, for each combination, the correlation coefficients between the reference results in M&C's experiment [24] and our similarity values. The process of selecting the most promising combination was based on the correlation r: eliminating combinations with weak correlation ($|r| < 0.5$), and keeping combinations with strong correlation ($0.5 \leq |r| \leq 1$).

$0.8 \times (SM_1 + SM_5 + SM_9) + 0.2 \times \sum_{\substack{i=2 \\ i \neq 5}}^{8} SM_i$ is the combination we decided to keep because its correlation was

the highest almost every time (in the range of $0.88 - 1$). This is due to the fact that given a and b are semantically similar, they satisfy that similar relations (a with b (SM_1), hypernyms of a with hypernyms of b (SM_5), and hyponyms of a with hyponyms of b (SM_9)) are more likely to be similar than different relations (a with hypernyms of b (SM_2), a with hyponyms of b (SM_3), hypernyms of a with b (SM_4), hypernyms of a with hyponyms of b (SM_6), hyponyms of a with b (SM_7), and hyponyms of a with hypernyms of b (SM_8)). Thus, the similarity value between a and b is calculated as follows:

$$Sim_{words}(a,b) = \begin{cases} 1, \text{ if } a \text{ and } b \text{ are synonyms or one of them is a direct hyponym of the other} \\ 0, \text{ if } [0.8 \times (SM_1 + SM_5 + SM_9) + 0.2 \times \sum_{\substack{i=2 \\ i \neq 5}}^{8} SM_i] \times \exp \frac{\sum_{\substack{i=1 \\ SM_i \neq 0}}^{9} 1}{9} \leq 1 \\ \frac{[0.8 \times (SM_1 + SM_5 + SM_9) + 0.2 \times \sum_{\substack{i=2 \\ i \neq 5}}^{8} SM_i] \times \exp \frac{\sum_{\substack{i=1 \\ SM_i \neq 0}}^{9} 1}{9} - 1}{[0.8 \times (SM_1 + SM_5 + SM_9) + 0.2 \times \sum_{\substack{i=2 \\ i \neq 5}}^{8} SM_i] \times \exp \frac{\sum_{\substack{i=1 \\ SM_i \neq 0}}^{9} 1}{9} + 1}, \text{ otherwise} \end{cases} \quad (16)$$

Step 2: Calculating the semantic similarity between sets of words. The similarity calculator uses the similarity measure between words (16) to compute the similarity between sets of words. Given two *entities* $e_1 \in S_1$ and $e_2 \in S_2$. Let $set_{e_1} = \{W_{1,1}, W_{1,2}, \ldots, W_{1,card(set_{e_1})}\}$ and $set_{e_2} = \{W_{2,1}, W_{2,2}, \ldots, W_{2,card(set_{e_2})}\}$ be their respective sets of words. The similarity calculator uses equation (17) to calculate the similarity between set_{e_1} and set_{e_2}.

$$Sim_{sets}(set_{e_1}, set_{e_2}) = \frac{1}{min(card(set_{e_1}), card(set_{e_2}))} \times \Big(\sum_{i=1}^{card(set_{e_1})} max(m_{i,j})_{1 \leq j \leq card(set_{e_2})} \quad (17)$$

Where $M = (m_{i,j})_{\substack{1 \leq i \leq card(set_{e_1}) \\ 1 \leq j \leq card(set_{e_2})}}$ is the similarity matrix. Its individual items are defined as follows $m_{i,j} = Sim_{words}(W_{1_i}, W_{2_j})$.

Next, we define the matches based on the similarity values.

C. The Post-matching Module

We applied our similarity measure (17) on the semantically related sets of words from the TEL schemas. The results formed a set of similarity values, each represents the similarity between two sets. The process of selecting the threshold value was based on reference matches we defined manually in order to identify the range of similarity values generated for semantically similar sets. We noticed that most matching sets have a similarity value greater than or equal to 0.8. Hence, we defined the threshold value 0.8 beyond which the pair of *entities* must be matched.

The post-matching module consists mainly of one major component, namely the *matches generator*, which uses the threshold value to eliminate *entity* pairs with very low similarity values, and match only pairs with high similarity values (≥ 0.8). Algorithm 5 summarizes this.

Algorithm 5 MatchesGenerator($SETS'_1$, $SETS'_2$, V)

Input:
$SETS'_1$
$SETS'_2$
V
Output:
$Matches$

```
for each v in V do
    if (v ≥ 0.8) then
        Matches ← Matches ∪ (set_1, set_2)
    end if
end for
return Matches
```

V. Experimental Results

We conducted extensive experiments to evaluate xMatcher based on a real implementation. We focused on evaluating two major issues. (1) We verified the accuracy of the results of our similarity measure, by evaluating the correlation coefficient and the Mean Square Error. (2) We examined the accuracy of the matches generated by xMatcher, by evaluating *Precision*, *Recall*, *Overall*, and *F-Measure*.

A. Experimental Setup

Datasets: First, we experimented our measure on M&C dataset [24], which contains thirty word pairs (see Table IV). We then experimented xMatcher over the *Conference Track* used in OAEI 2018 and available on the Web[3]. The *Conference Track* involves 16 ontologies describing the domain of organizing academic conferences. It has been used by the research community for over 13 years. It has 21 reference alignments composed from 7 out of 16 real domain ontologies.

Implementation: In addition to our measure, we implemented four measures and distances Resnik, J&C, Lin, and Nababteh over WordNet. Then, we implemented xMatcher. Finally, since xMatcher was initially developed to take as input XML schemas and since the *Conference Track* includes ontologies, we implemented the converting process presented in [30] to transform ontologies into XML schemas.

Measures: For semantic similarity values (produced by all five measures), we used the correlation coefficient and Mean Square Error (MSE) to compare the returned results with the reference results [24]. The correlation coefficient measures how strong the relationship is between the returned values and the reference results. MSE measures the average of the squares of the errors between the returned values and the reference results. The lower the MSE is, the better.

For matching results, we used the previously published results produced by twelve ontology matching systems (SANOM [31], AML [13], LogMap [32], XMap [33], KEPLER [34], ALIN [9], DOME [11], Holontology [10], FCAMapX [35], [36], LogMapLt [32], ALOD2Vec [12], and Lily [37]) that

[3] http://oaei.ontologymatching.org/2018/

TABLE IV. SEMANTIC SIMILARITY VALUES BY WORD PAIR

Word pair	M&C	Resnik	J&C	Lin	Nababteh	Our measure
Automobile / Car	0.98	0.9962	1	1	1	1
Journey / Voyage	0.96	0.9907	0.9165	0.8277	0.857335	1
Gem / Jewel	0.96	1	1	0.2434	0.31453	1
Boy / Lad	0.94	0.9971	0.8613	0.6433	1	1
Coast / Shore	0.925	0.9994	0.9567	0.96	1	1
Asylum / Madhouse	0.9025	1	0.9379	0.769	0.879	1
Magician / Wizard	0.875	0.9999	1	0.1958	0.28158	1
Midday / Noon	0.855	0.9998	1	1	1	1
Furnace / Stove	0.7775	0.6951	0.593	0.2294	0.26674	0.79
Food / Fruit	0.77	0.9689	0.7925	0.0956	0.103839	0.98
Bird / Cock	0.7625	0.9984	0.8767	0.7881	0.930014	1
Bird / Crane	0.7425	0.9984	0.815	0	0.850943	0.95
Implement / Tool	0.7375	0.9852	0.977	0.914	1	1
Brother / Monk	0.705	0.8722	0.6656	0	1	1
Crane / Implement	0.42	0.8722	0.6526	0	0.513459	0.73
Brother / Lad	0.415	0.8693	0.6775	0.24	0.29735	0.62
Car / Journey	0.29	0	0.5883	0	0	0
Monk / Oracle	0.275	0.8722	0.6203	0.1828	0.191595	0.75
Food / Rooster	0.2225	0.5036	0.5885	0.0762	0.095302	0.66
Coast / Hill	0.2175	0.9867	0.8487	0.127	0.19414	0.49
Forest / Graveyard	0.21	0	0.484	0.1119	0.1706	0.61
Monk / Slave	0.1375	0.8722	0.6962	0.2011	0.34281	0.25
Coast / Forest	0.105	0	0.5179	0	0	0.2
Lad / Wizard	0.105	0.8722	0.6905	0.2241	0.34155	0.2
Cord / Smile	0.0325	0.8044	0.5845	0	0	0.11
Glass / Magician	0.0275	0.5036	0.5699	0.0663	0.09335	0
Rooster / Voyage	0.02	0	0.4168	0	0	0
Noon / String	0.02	0	0.4329	0	0	0

participated in OAEI 2018 over the *Conference Track*. We used $Precision$ (18), $Recall$ (19), $Overall$ (20), and $F-Measure$ (21) [38] to evaluate the returned matches based on nine combinations of evaluation variants with 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 violations of conservativity). *ra1-M1*, *ra2-M1*, and *rar2-M1* are used to evaluate only alignments between classes; *ra1-M2*, *ra2-M2*, and *rar2-M2* are used to evaluate only alignments between properties; and *ra1-M3*, *ra2-M3*, and *rar2-M3* are used to evaluate both alignments between classes and properties.

$$Precision = \frac{Correct\ Matches}{Correct\ Matches + Incorrect\ Matches} \quad (18)$$

(18) is the probability of correct matches among the matches returned by a matching system.

$$Recall = \frac{Correct\ Matches}{Missed\ Matches + Correct\ Matches} \quad (19)$$

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

$$Overall = Recall \times (2 - \frac{1}{Precision}) \quad (20)$$

(20) quantifies the amount of manual post-effort necessary to remove false matches and add missed matches.

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (21)$$

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

B. Results and Discussion

1) Semantic Similarity Measure: Experiment Results: We first applied our measure, Resnik, J&C, Lin, and Nababteh on M&C dataset (see the results in Table IV). We then used the results to calculate the correlation coefficient and MSE (see the overall results in Table V and the details about correlations in Fig. 3(a), Fig. 3(b), Fig. 3(c), Fig. 3(d), and Fig. 3(e)).

TABLE V. COMPARISON BETWEEN SOME STATE OF THE ART SIMILARITY MEASURES AND OUR MEASURE

Measure	Correlation coefficient	MSE
Resnik	0.6671	0.1373
J&C	0.8363	0.1018
Lin	0.6852	0.1188
Nababteh	0.7654	0.0699
Our measure	0.9102	0.0453

The findings indicate a strong positive correlation ($+0.9102$) between our measure and the reference results. They also indicate that our measure obtained the smallest MSE (0.0453) compared to the other measures. Thus, our measure outperforms the state of the art measures, showing that on the one hand information content-based measures cannot provide high accuracy results; on the other hand combining different WordNet information (hypernyms, direct hyponyms, senses, and *synsets*) is a good plus to obtain high accuracy results.

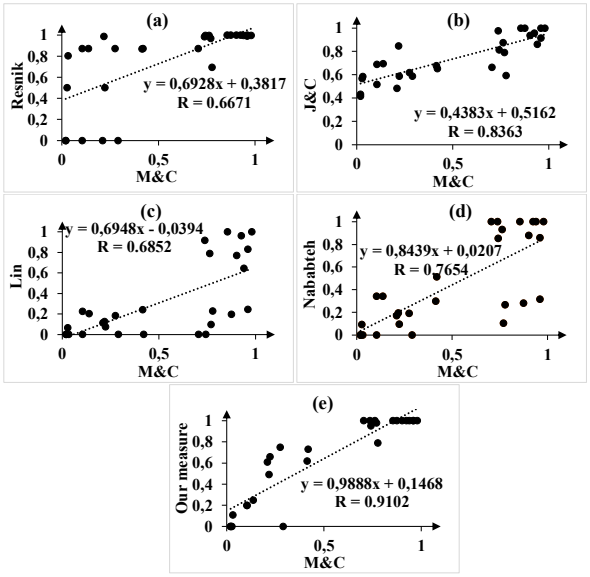

Fig. 3. Regression lines for (a) Resnik vs. M&C; (b) J&C vs. M&C; (c) Lin vs. M&C; (d) Nababteh vs. M&C; and (e) Our measure vs. M&C

2) xMatcher: Experiment Results: We first generated the matches using xMatcher. We then calculated, for all matches for which there is a reference alignment, *Precision*, *Recall*, *Overall*, and *F-Measure* nine times, each time with a different reference alignment. Fig. 4(a), Fig. 4(b), Fig. 4(c), Fig. 4(d), Fig. 4(e), Fig. 4(f), Fig. 4(g), Fig. 4(h), and Fig. 4(i) present the new and previously published results.

On the one hand, the previously published results indicate noticeable changes in *Precision*, *Recall*, *Overall*, and *F-Measure*: overall, they achieved good matching accuracy when evaluated based on *ra1-M1*, *ra1-M3*, *ra2-M1*, *ra2-M3*, *rar2-M1*, and *rar2-M3*; and low accuracy even null sometimes (Lily and ALIN) with *ra1-M2*, *ra2-M2*, and *rar2-M2*. On the other hand, xMatcher obtained high accuracy matches, outperforming all systems almost every time except from *ra1-M2* and *ra2-M2* where AML surpassed it slightly ($Precision = 1$).

While xMatcher matches both classes and properties, Lily and ALIN match only classes the reason why they failed to produce high accuracy matches with *ra1-M2*, *ra2-M2*, and *rar2-M2*; SANOM, AML, LogMap, and XMap match some but not all properties which explain their negative *Overall* with *ra1-M2*, *ra2-M2*, and *rar2-M2*; and KEPLER, DOME, Holontology, FCAMapX, LogMapLt, and ALOD2Vec match very few properties which justify their negative *Overall* and low *Precision*, *Recall*, and *F-Measure* with *ra1-M2*, *ra2-M2*, and *rar2-M2*. We can conclude that (1) SANOM, AML, LogMap, XMap, KEPLER, ALIN, DOME, Holontology, FCAMapX, LogMapLt, ALOD2Vec, and Lily work well with the reference alignments that consider classes or both classes and properties. However, they fail to match correctly with the reference alignments that consider only properties; and (2) xMatcher succeeds to achieve superior accuracy matches regardless of the reference alignment it is compared to.

Overall, xMatcher obtained the highest accuracy matches (see Fig. 4.j which displays the average matching accuracy): $Precision = 0.89$ suggests that most matches are correct;

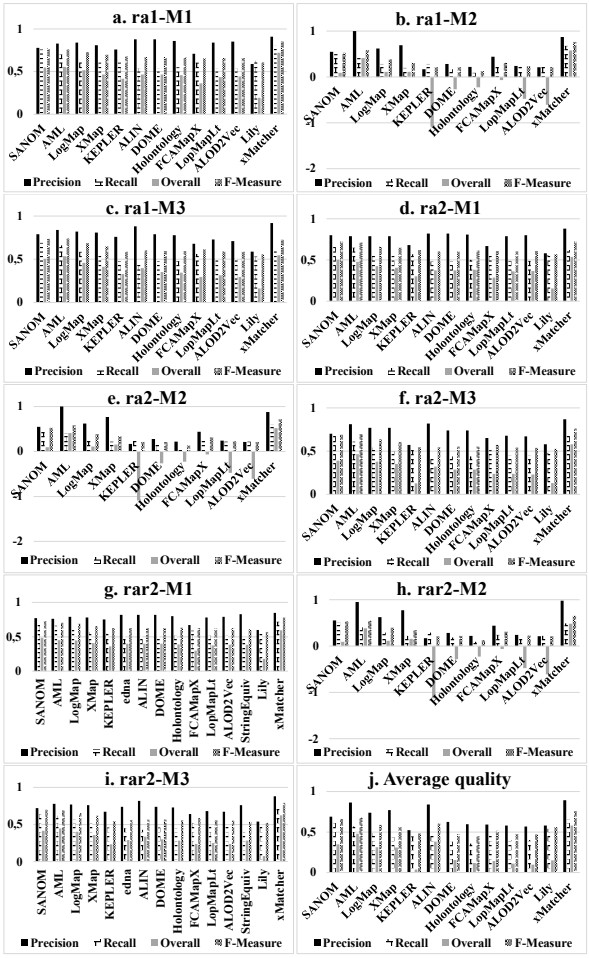

Fig. 4. Accuracy of the Matches

$Recall = 0.66$ suggests that xMatcher missed only few matches; and $Overall = 0.57$ implies that xMatcher needs only a small amount of manual post-effort to correct the results.

To prove scalability of xMatcher (note that due to space limitation, we do not display the results in figures in this paper), we applied xMatcher on more datasets, for instance the TEL (Travel, Entertainment and Living) datasets which contain five different datasets that are publicly available on the Web. The Travel group includes two various domains: *Car Rentals* and *Airfare*; the Entertainment group contains two different domains as well: *Movies* and *Books*; and, the Living group involves mainly one single domain: *Jobs*. The results show once again the capability of xMatcher to reach a high matching accuracy, which proves that xMatcher is scalable.

VI. Conclusion

We have demonstrated that the use of WordNet combined with our semantic similarity measure is an effective way to capture semantic correspondences in XML schemas. Current matching systems are error-prone and human-dependent. Thus, we have developed xMatcher, an approach to automatically match XML schemas and provide accurate matches.

Given two XML schemas S_1 and S_2, our main idea is to first generate sets of words from S_1 and S_2, then determine

semantically related sets, and finally identify semantic correspondences between related sets. We evaluated xMatcher over the *Conference Track*. The results show that xMatcher achieves better accuracy than twelve state of the art matching systems. Future research includes the following:

- **Improving the accuracy of the matches.** An interesting direction is to achieve better correlation, MSE, *Precision*, *Recall*, *Overall*, and *F-Measure*.
- **Considering other matching quality factors.** In this paper, we focused on achieving high matching accuracy. A future direction is to propose techniques that consider other quality factors.
- **Matching other data representations.** xMatcher takes as input XML schemas. An interesting direction is to match different data representations.

REFERENCES

[1] C. Zhang, L. Chen, H. Jagadish, M. Zhang, and Y. Tong, "Reducing uncertainty of schema matching via crowdsourcing with accuracy rates," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[2] Y. Lee, M. Sayyadian, A. Doan, and A. S. Rosenthal, "etuner: tuning schema matching software using synthetic scenarios," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 16, no. 1, pp. 97–122, 2007.

[3] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: A literature review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015.

[4] F. Ardjani, D. Bouchiha, and M. Malki, "Ontology-alignment techniques: survey and analysis," *International Journal of Modern Education and Computer Science*, vol. 7, no. 11, p. 67, 2015.

[5] L. Mukkala, J. Arvo, T. Lehtonen, T. Knuutila, *et al.*, "Current state of ontology matching. a survey of ontology and schema matching," 2015.

[6] S. Anam, Y. S. Kim, B. H. Kang, and Q. Liu, "Review of ontology matching approaches and challenges," *International journal of Computer Science and Network Solutions*, vol. 3, no. 3, pp. 1–27, 2015.

[7] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[8] 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 *OM@ ISWC*, pp. 125–131, 2018.

[9] J. da Silva, K. Revoredo, and F. A. Baião, "ALIN results for OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 117–124, 2018.

[10] P. Roussille, I. Megdiche Bousarsar, O. Teste, and C. Trojahn, "Holontology: results of the 2018 oaei evaluation campaign," CEUR-WS: Workshop proceedings, 2018.

[11] S. Hertling and H. Paulheim, "Dome results for oaei 2018.," in *OM@ ISWC*, pp. 144–151, 2018.

[12] J. Portisch and H. Paulheim, "Alod2vec matcher.," in *OM@ ISWC*, pp. 132–137, 2018.

[13] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 527–541, Springer, 2013.

[14] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: efficient matching for large real-world schemas and ontologies," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1586–1589, 2009.

[15] L. Meng, R. Huang, and J. Gu, "A review of semantic similarity measures in wordnet," *International Journal of Hybrid Information Technology*, vol. 6, no. 1, pp. 1–12, 2013.

[16] B. Poorna and A. S. Ramkumar, "Semantic similarity measures: an overview and comparison," *International Journal of Advanced Research in Computer Science*, vol. 9, no. Special Issue 1, p. 100, 2018.

[17] M. H. El Yazidi, A. Zellou, and A. Idri, "Towards a fuzzy mapping for mediation systems," in *2012 IEEE International Conference on Complex Systems (ICCS)*, pp. 1–4, IEEE, 2012.

[18] A. Gupta, A. Kumar, J. Gautam, A. Gupta, M. A. Kumar, and J. Gautam, "A survey on semantic similarity measures," *IJIRST-International Journal for Innovative Research in Science & Technology*, vol. 3, p. 12, 2017.

[19] A. Yousfi, M. H. Elyazidi, and A. Zellou, "Assessing the performance of a new semantic similarity measure designed for schema matching for mediation systems," in *International Conference on Computational Collective Intelligence*, pp. 64–74, Springer, 2018.

[20] A. M. Abdelrahman and A. Kayed, "A survey on semantic similarity measures between concepts in health domain," *American Journal of Computational Mathematics*, vol. 5, no. 02, p. 204, 2015.

[21] M. H. E. Yazidi, A. Zellou, and A. Idri, "FMAMS: fuzzy mapping approach for mediation systems," *Int. J. Appl. Evol. Comput.*, vol. 4, no. 3, pp. 34–46, 2013.

[22] M. H. E. Yazidi, A. Zellou, and A. Idri, "Mapping in GAV context," in *10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015, Rabat, Morocco, October 20-21, 2015*, pp. 1–5, 2015.

[23] F. Couto and A. Lamurias, "Semantic similarity definition," *Encyclopedia of bioinformatics and computational biology*, vol. 1, 2019.

[24] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *arXiv preprint cmp-lg/9511007*, 1995.

[25] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *arXiv preprint cmp-lg/9709008*, 1997.

[26] D. Lin, "Principle-based parsing without overgeneration," in *31st annual meeting of the association for computational linguistics*, pp. 112–120, 1993.

[27] N. Mohammed and D. Mohammed, "New modified semantic similarity measure based on information content approach," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 3, p. 73, 2017.

[28] A. Tversky, "Features of similarity.," *Psychological review*, vol. 84, no. 4, p. 327, 1977.

[29] Z. Zhou, Y. Wang, and J. Gu, "New model of semantic similarity measuring in wordnet," in *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, vol. 1, pp. 256–261, IEEE, 2008.

[30] L. Mukkala, J. Arvo, T. Lehtonen, and T. Knuutila, "Trc-matcher and enhanced trc-matcher. new tools for automatic xml schema matching," 2017.

[31] M. Mohammadi, W. Hofman, and Y. Tan, "SANOM results for OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 205–209, 2018.

[32] E. Jiménez-Ruiz, B. C. Grau, and V. Cross, "Logmap family participation in the OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 187–191, 2018.

[33] W. E. Djeddi, S. B. Yahia, and M. T. Khadir, "Xmap: results for OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 210–215, 2018.

[34] M. Kachroudi, G. Diallo, and S. B. Yahia, "KEPLER at OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 173–178, 2018.

[35] M. Zhao and S. Zhang, "Fca-map results for oaei 2016.," in *OM@ ISWC*, pp. 172–177, 2016.

[36] G. Chen and S. Zhang, "Fcamapx results for OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 160–166, 2018.

[37] Y. Tang, P. Wang, Z. Pan, and H. Liu, "Lily results for OAEI 2018," in *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pp. 179–186, 2018.

[38] I. Kastner and F. Adriaans, "Linguistic constraints on statistical word segmentation: The role of consonants in arabic and english," *Cognitive science*, vol. 42, pp. 494–518, 2018.