sensors

MDPI

Article

Optimizing Sensor Ontology Alignment through Compact co-Firefly Algorithm

Xingsi Xue [1,2,3,4,5,*] and Junfeng Chen [6]

1 Fujian Key Lab for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou 350118, China
2 Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (Guilin University of Electronic Technology), Guilin 541004, China
3 Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou 350118, China
4 Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fuzhou 350118, China
5 College of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China
6 College of IOT Engineering, Hohai University, Changzhou 213022, China; chen-1997@163.com
* Correspondence: jack8375@gmail.com; Tel.: +86-15359746498

Received: 17 February 2020; Accepted: 2 April 2020; Published: 6 April 2020

Abstract: Semantic Sensor Web (SSW) links the semantic web technique with the sensor network, which utilizes sensor ontology to describe sensor information. Annotating sensor data with different sensor ontologies can be of help to implement different sensor systems' inter-operability, which requires that the sensor ontologies themselves are inter-operable. Therefore, it is necessary to match the sensor ontologies by establishing the meaningful links between semantically related sensor information. Since the Swarm Intelligent Algorithm (SIA) represents a good methodology for addressing the ontology matching problem, we investigate a popular SIA, that is, the Firefly Algorithm (FA), to optimize the ontology alignment. To save the memory consumption and better trade off the algorithm's exploitation and exploration, in this work, we propose a general-purpose ontology matching technique based on Compact co-Firefly Algorithm (CcFA), which combines the compact encoding mechanism with the co-Evolutionary mechanism. Our proposal utilizes the Gray code to encode the solutions, two compact operators to respectively implement the exploiting strategy and exploring strategy, and two Probability Vectors (PVs) to represent the swarms that respectively focuses on the exploitation and exploration. Through the communications between two swarms in each generation, CcFA is able to efficiently improve the searching efficiency when addressing the sensor ontology matching problem. The experiment utilizes the Conference track and three pairs of real sensor ontologies to test our proposal's performance. The statistical results show that CcFA based ontology matching technique can effectively match the sensor ontologies and other general ontologies in the domain of organizing conferences.

Keywords: sensor ontology; Compact co-Firefly Algorithm; ontology matching

1. Introduction

In recent years, sensors have been used in a wide range of applications, such as urban traffic planning, flood prediction, health care, satellite imaging for earth and space observation et al. To make different types of sensors collaborate on a common task to detect and identify a multitude of observations, we need to combine the sensor data with the Internet, web services and database technologies, which is the so-called sensor web [1,2]. However, since the acquired sensor data on the sensor web lacks of semantic information and may be heterogeneous in the syntax, schema and semantic level and so forth, it is difficult to share and integrate them and implement the

communications among diverse sensor applications [3]. To address this problem, Semantic Web (SW) techniques were introduced into the sensor network to annotate the sensor data with spatial, temporal, and thematic semantic meta-data. The combination of the semantic web technology and sensor network leads to the birth of Semantic Sensor Web (SSW) [4], which dedicates to provide semantic meaning for sensor observations so as to enable the inter-operability and advanced analysis for situation awareness and other advanced applications from heterogeneous sensors. In association with semantic annotation, sensor ontologies play an important role in SSW for constructing the shared conceptual model, enhancing the sensor data semantics, and realizing the interaction of spatial, temporal, and thematic sensor data. Annotating sensor data with different sensor ontologies can be of help to implement different sensor systems' inter-operability, which requires that inter-operable sensor ontologies. To this end, we need to establish meaningful links between semantically related sensor information, which is the so-called sensor ontology matching. Since the sensor ontology can be described through its architecture graph (the nodes represent the concepts and instances, and the edges the relationships between them), the sensor ontology matching process corresponds to the determination of the largest isomorphic sub graph between two architecture graphs of two ontologies to be aligned [5]. Therefore, modeling two sensor ontologies under alignment is a complex and time-consuming task, particularly when the scale of their entities is large. For this reason, approximate methods, such as Swarm Intelligent Algorithm (SIA), represents a suitable methodology for determining the high-quality alignments [6].

With the development of SSW, the scale of sensor ontologies increase significantly, which makes the traditional SIA-based approaches unable to efficiently determine the high-quality alignment due to the huge memory consumption, long runtime and premature convergence. To reduce the memory consumption and runtime, we first investigate a new category of efficient optimization algorithm, that is, Compact Optimization Algorithm (COA), which makes use of the probability representation to approximately present the population. Since COA only needs to restore a Probability Vector (PV), it significantly saves the hardware resources to execute the evolving process. The first COA is the Compact Genetic Algorithm (CGA) [7], and later, Baraglia et al. [8] improve its performance by introducing a local search strategy. Ahn et al. [9] use two elite strategies to enhance CGA's performance. Neri et al. [10] propose a noise robust version to deal with the situation with noise context. They also propose a Compact Particle Swarm Optimization (CPSO) [11] to approximate the swarm's behaviour. Mininno et al. propose a Compact Differential Evolution algorithm (CDE) [12], whose survivor selection scheme can be straightforwardly encoded, which can efficiently perform an optimization process with a limited memory requirement.

In addition, to overcome the algorithm's premature convergence, we also investigate another kind of optimization algorithm, that is, co-Evolutionary Algorithm (cEA), which makes multiple sub-swarms evolves independently and exchanges the information of each sub-swarms at particular time to improve the searching efficiency in the large search space. Tan et al. [13] first decomposes the problem's vector dimensions, and then tries to make each sub-swarm addresses one sub-problem. Mu et al. [14] use the multiple elite strategies to execute the evolving process, and they assign the centered sub-swarm with higher priority. Zhou et al. propose a Parallel cEA (PcEA), which divides the population into three sub-swarms and they evolve with the same evolving strategy. To improve the algorithm's search ability, Liu et al. [15] propose two elite strategies, that is, a better elite strategy and a worse elite strategy. They utilize two sub-swarms with different evolving strategies to respectively update two kinds of elites. EI-Abd [16] propose a Cooperative Co-evolutionary Brain Storm Optimization algorithm (CCBSO), which is based on the explicit space decomposition approach. In this work, we combines the COA with cEA to have their complementary advantages, and being inspired by the success of Firefly Algorithm (FA) in many domains [17,18], we further propose a general-purpose ontology matching technique based on Compact co-Firefly Algorithm (CcFA) to efficiently optimize the ontology alignments.

The rest of the paper is organized as follows: Section 3 presents the basic concepts on sensor ontology matching; Section 4 shows the CcFA-based sensor ontology matching technique in details; Section 5 presents the greedy strategy to filtering the final alignment; Section 6 shows the statistical experiment; and finally, Section 7 draws the conclusion and presents the future work.

2. Swarm Intelligence Algorithm Based Ontology Alignment

Many Machine Learning (ML) techniques have been applied to match ontologies to determine high-quality alignment, such as Logistic Regression (LR) [19], Neural Network (NN) [20], Word Embedding (WE) [21], Graph Embedding (GE) [22], Support Vector Machine (SVM) [23], Clustering Algorithm [24], Decision Tree (DT) [25] and so forth. Researchers also try to improve the matching efficiency through the high performance computing techniques such as Parallel Computing (PC) [26] and Cloud Computing (CC) [27]. Due to the complexity of the ontology matching process, recently, SIA-based technique has become an efficient approach for determining high-quality ontology alignment.

The first generation of SI-based matchers aimed at solving the ontology meta-matching problem, that is, how to determine the optimal parameters to aggregate different matchers and optimize the quality of obtained ontology alignment. Genetics for Ontology ALignments (GOAL) [28] was the first SI-based ontology meta-matcher, which used Evolutionary Algorithm (EA) to optimize the aggregating weight set of different ontology matchers. Ginsca et al. [29] proposed to use EA to optimize the all the parameters in the whole meta-matching process, which included the aggregating weight set and a threshold for filtering the final alignment. Xue et al. [30] introduced a new metric to measure the ontology alignment's quality, which did not require the utilization of golden standard alignment, and formally defined ontology meta-matching problem. Their approach was able to match multiple ontology pairs at a time and overcame three drawbacks of the EA-based meta-matchers. More recently, He et al. [31] used Artificial Bee Colony (ABC) algorithm to address the ontology meta-matching problem, whose results were better than the EA-based matchers. Recently, Xue et al. [32] propose a Multi-objective CFA (MCFA) to optimize the ontology alignments. Their proposal borrows the idea of MOEA/D [33] by first decomposing the original problem into three sub-problems, and then use three PVs to respectively address three sub-problems. Later on, Xue [34] further construct a single objective model for the biomedical ontology matching problem, and proposes a Compact Firefly Algorithm (CFA) to address it. CFA utilizes the compact encoding mechanism to represent the swarm of fireflies with a probabilistic model, instead of storing each one, but it may result in premature convergence. To overcome this drawback, in this work, CcFA combines the compact encoding mechanism with the co-Evolutionary mechanism to further enhance the algorithm's performance.

3. Sensor Ontology Matching Problem

An ontology can be defined as a 3-tuples (C, DP, OP), where C is the set of classes, that is, the set of concepts that populate the domain of interest; DP is the set of data properties, that is, the set of features describing the classes; OP is a set of object properties, that is, the set of relations existing between the concepts. In particular, concept, datatype property and object property are called ontology entities. Sensor ontology formally defines the shared concepts and their relationships in the sensor domain. However, since the sensor ontologies is developed and maintained by different organizations with various requirements, they may define the same observation with different ways, which yields the sensor ontology heterogeneity problem. It is necessary to find the identical sensor entities to bridge the semantic gap between two ontologies, which is the so-called sensor ontology matching. The obtained sensor entity mapping set is called sensor ontology alignment.

It is obvious that how to calculate two sensor entity's similarity value is the prerequisite technique for matching sensor ontologies [35]. Since none of the similarity measure can ensure their effectiveness in any context, usually, a sensor ontology system aggregates several similarity measures to improve the result's confidence. In this work, we combine three broad categories of similarity measures to measure two sensor entities' similarity, that is, structure-based, linguistic-based and syntax-based

similarity measures. To be specific, we construct a context profile for each sensor entity by collecting the information from its direct ascendant and descendant entities. Then, the similarity of two entities e_1 and e_2 is calculated as follows:

$$sim(e_1, e_2) = \frac{2 \times |p_1 \bigcap p_2|}{|p_1| + |p_2|}, \tag{1}$$

where p_1 and p_2 are respectively e_1 and e_2's profiles. With respect to the similarity measure on two elements w_1 and w_2 in two profiles, we aggregate the Wordnet [36] based distance, a linguistic-based similarity measure, and the N-gram distance [37], a syntax-based similarity measure, whose equation is as follows:

$$sim(w_1, w_2) = \begin{cases} 1, & \text{two words are synonymous} \\ N - gram(w_1, w_2), & \text{otherwise} \end{cases}. \tag{2}$$

Since the alignment with more correspondences and higher mean similarity value should have better quality, the quality of a sensor ontology alignment is calculated as follows:

$$f(A) = 2 \times \frac{\frac{|A|}{max\{|O_1|,|O_2|\}} \times \frac{\sum_{i=1}^{|A|} sim_i}{|A|}}{\frac{|A|}{max\{|O_1|,|O_2|\}} + \frac{\sum_{i=1}^{|A|} sim_i}{|A|}}, \tag{3}$$

where $|O_1|$, $|O_2|$ are respectively the entity number of two sensor ontology O_1 and O_2, $|A|$ is the mapping number of the alignment A, and sim_i is the i-th entity mapping's similarity. Further, the sensor ontology matching problem is defined as follows:

$$\begin{cases} max \quad f(X) \\ s.t. \quad X = (x_1, x_2, ..., x_{|O_1|})^T \\ \qquad x_i = 1, 2, \cdots, |O_2| \end{cases}, \tag{4}$$

where x_i is the ith entity mapping between ith entity in O_1 and x_ith entity in O_2.

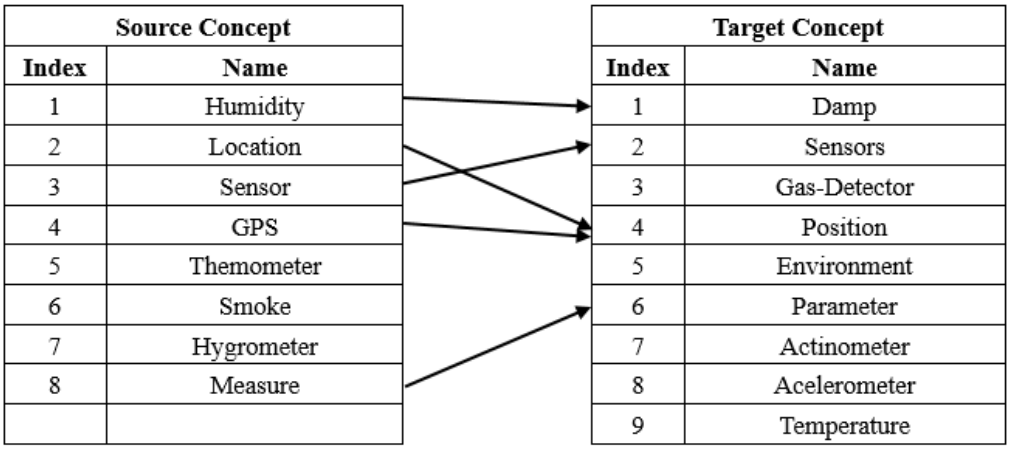
4. Compact co-Firefly Algorithm

FA is inspired by the social behaviour of fireflies, where fireflies produce short and rhythmic flashes for communication. The intuition behind FA is that fireflies tend to fly to the brighter locations, which is a solution with better objective function's value [18]. In this work, we propose a Compact co-Firefly Algorithm, that is, CcFA, which utilizes the compact encoding mechanism and the co-Evolutionary mechanism to improve the solution's quality. CcFA works with two sub-swarms, that is, the better swarm with a better elite solution and a worse swarm with a worse elite solution, which respectively apply different evolving strategy. In particular, the better swarm mainly focus on the exploitation which can increase the convergence speed, while the worse swarm emphasizes on the exploration which is able to enhance the solution's diversity. In each generation, the one with a better elite solution will become the better swarm, and in this way, two swarms can better trade off the algorithm's exploitation and exploration. To improve the searching performance, we utilize two Probability Vectors (PVs) to represent two sub-swarms, that is, PV_{better} and PV_{worse}.

4.1. Compact Encoding Mechanism

An alignment consists of several entity mappings, and each entity mapping can be described by two entities' indices. On this basis, in this work, we empirically choose the Gray code to encode an alignment, which is a binary encoding mechanism. An example of the compact encoding mechanism is shown in the Figure 1, where the index means the source concept index and the corresponding bit values are the target concept index that is encoded through Gray code, for example, the source concept "Measure" with index 8 is mapped to target concept "Parameter" with index 6 whose Gray code is 110.

In particular, Gray code 000 means a source concept is not mapped to any target concept. We need to point out that, given the the scale of target ontology n, the length of code is equal to $\lceil log_2 n \rceil$, and Figure 1 only shows a simple example of encoding with Gray code.



Source Concept	
Index	**Name**
1	Humidity
2	Location
3	Sensor
4	GPS
5	Themometer
6	Smoke
7	Hygrometer
8	Measure

Target Concept	
Index	**Name**
1	Damp
2	Sensors
3	Gas-Detector
4	Position
5	Environment
6	Parameter
7	Actinometer
8	Acelerometer
9	Temperature

The individual's chromosome

Code	001	100	010	100	000	000	000	110
Index	1	2	3	4	5	6	7	8

Figure 1. An example of compact encoding mechanism.

A PV's dimension number is equal to the length of an individual, and each dimension represents the probability of being one corresponding to each individual bit's value. Since PV stores each individual bit's corresponding probability of being one, each bit value of a newly generated individual can be determined by comparing a random number with its corresponding probability in PV. Figure 2 shows an example of generating a solution through PV. Given a PV $(0.1, 0.3, 0.5, 0.9)^T$ where each element presents the probability of being 1 with respect to a solution's gene bit, generate four random numbers in [0,1], for example, 0.2, 0.4, 0.6 and 0.1, and we can determine a new solution by comparing them with PV's elements accordingly. To be specific, since 0.2 > 0.1, 0.4 > 0.3, 0.6 > 0.5 and 0.1 < 0.9, the newly generated solution is 0001. In addition, PV is updated in each generation to make the new individuals generated in the next are closer to the elite. With respect to the word "closer", it means the newly generated individual is more likely to be the elite solution. For example, given two PVs $PV_1 = (0.1, 0.1, 0.1, 0.1)^T$ and $PV_2 = (0.9, 0.9, 0.9, 0.9)^T$, it is obvious that a new individual generated by PV_2 is more likely to be 1111 than that generated by PV_1. Therefore, if the elite 1111 is the global optimal solution, we need to update PV_1 by moving it towards PV_2. In particular, if the bit value of the elite solution is 1 (or 0), PV's first element will be increased (decreased) by a $step$, which can make the newly generated solution more closer to the elite solution. Given a PV $(0.1, 0.3, 0.5, 0.9)^T$, an elite solution 1110 and $step = 0.1$, since the first bit value of the elite solution is 1, accordingly, we update the first element of PV by $step$, that is 0.2, which makes the first bit value of newly generated individual is more likely to be 1 (the same with elite solution's first bit value). Therefore, after updating all elements of PV, the newly generated individuals would be closer to the elite solution in terms of each bit value. When all elements of PV are closed to 1 or 0, the newly generated individuals will be same and the algorithm converges. For more about the theory analysis on the compact encoding mechanism, please see also the work of Harik et al. [7].

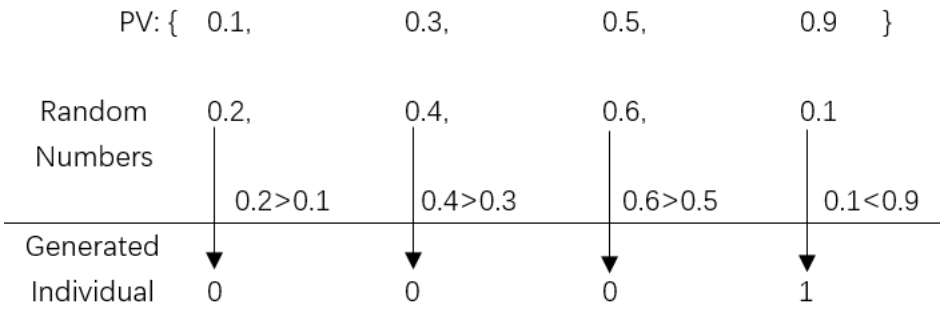


Figure 2. An example of generating an individual through PV.

4.2. Movement Operator

In the classic FA, a firefly ind_i's position is updated by moving it to a more attractive firefly ind_j by the α-step and β-step, which are respectively given as follows:

$$\alpha - step(ind_i) = ind_i + \alpha(rand() - 1/2). \tag{5}$$

$$\beta - step(ind_i) = \frac{\beta_0}{1 + \gamma r_{ij}^2} r_{ij}, \tag{6}$$

where α is a randomisation function, r_{ij} is two fireflies' distance, β_0 is the attractiveness when $r_{ij} = 0$, γ is the light absorption parameter.

4.2.1. Exploitation Strategy

FA's exploitation searches for the better individual in the vicinity of a solution, which is implemented through α-step. In this work, α-step can be implemented through a local search process on ind_i. Given a firefly ind, we first generate C new individuals through PV; then, utilize the binary crossover operator on each of the new individual and ind to obtain ind's neighborhood, finally, we select the elite from its neighborhood. The pseudo-code of α-step algorithm is shown in Algorithm 1.

Algorithm 1 α-step Algorithm

```
for int i = 0; i < C, i++ do
    generate a new solution ind_new through Probability Vector (PV);
    ind_i = ind.copy();
    int index = round(rand(0, C));
    while rand(0, 1) <0.6 do
        ind_{i,index} = ind_{new,index};
        index++;
        if index == C then
            index = 0;
        end if
    end while
end for
return the best individual in {ind_1, ind_2, ··· , ind_C}
```

4.2.2. Exploration Strategy

FA's exploration aims at keeping the the population's diversity, which is implemented through β-step. In this work, we utilize the edit distance to measure two fireflies ind_i and ind_j's distance:

$$edit(ind_i, ind_j) = \sum_{k=1}^{|ind_i|} |ind_{i,k} - ind_{j,k}|, \tag{7}$$

where $|ind_i|$ is the number of ind_i's bits, $ind_{i,k}$ and $ind_{j,k}$ are respectively the kth bit of ind_i and ind_j. Next, a new individual ind_i can be obtained according to $edit(ind_r, ind_s)$ and $\beta_0/(1+\gamma r_{ij}^2)$. The pseudo-code of β-step algorithm is shown in Algorithm 2.

When the edit distance between two PVs is too close, all PV_{worse}'s elements will be initialized. In particular, the edit distance between two PVs are defined as follows:

$$edit(PV_{better}, PV_{worse}) = \frac{\sum_{i=1}^{|PV|}(|PV_{better}^i - PV_{worse}^i|)}{|PV|}, \tag{8}$$

where $|PV|$ is PV's dimension number, PV_{better}^i and PV_{worse}^i are respectively the ith dimension of the PV_{better} and PV_{worse}.

Algorithm 2 β-step Algorithm

for $k = 0; k < |ind_i|; k$++ **do**
  if $ind_i[k] != ind_j[k]$ **then**
    put k into the list $index$;
  end if
end for
$totalNum = round(rand(0,1) \times Hamming(ind_i, ind_j))$;
$num = 0$;
$n = 0$;
while $num < totalNum$ **do**
  if $rand(0,1) < \frac{\beta_0}{1+\gamma r_{ij}^2}$ **then**
    $ind_i[index[n]] = (ind_i[index[n]] + 1)\ mod\ 2$;
    remove $index[n]$ from $index$;
    $num = num + 1$;
  end if
  $n = (n+1)\ mod\ index.length()$;
end while

4.3. Pseudo-Code of Compact co-Firefly Algorithm

In this work, CcFA uses the following configuration: the maximum generation $maxGen = 3000$, the attractiveness $\beta_0 = 1.0$, the light absorption parameter $\gamma = 0.02$, the local search's neighborhood scale $C = 5$, the step length for updating PV $step = 0.1$. These parameters represent a trade off configuration obtained in an empirical way to achieve the highest average alignment quality on all testing cases. CcFA's pseudo-code is shown in Algorithm 3.

CcFA applies two evolutionary strategies on PV_{better} and PV_{worse}, respectively. Through the competition between two elites, PV_{better} and PV_{worse} are updated. By adaptively switching the search strategies, CcFA can better trade off the algorithm's converging speed and the individuals' diversity. In particular, when all elements of PV_{better} and PV_{worse} are equal to 1 or 0, CcFA converges, and the algorithm terminates the loop and outputs $elite_{better}$.

Algorithm 3 Compact co-Firefly Algorithm

** Initialization **
$generation = 0$;
Initialize PV_{better} and PV_{worse} by setting all the probabilities inside as 0.5;
generate $elite_{better}$ and $elite_{worse}$ through PV_{better} and PV_{worse}, respectively;
** Evolving Process **
while $generation < maxGen$ **do**
  ** Exploitation **
  generate a solution ind_{new} through PV_{better};
  ind_{new} = $alpha$-step(ind_{new}); // see also Algorithm 1
  ** Competition **
  $compete(elite_{better}, ind_{new})$;
  if $winner == ind_{new}$ **then**
    $elite_{better} = ind_{new}$;
  end if
  ** PV Update **
  for $i = 0; i < PV_{better}.length; i$++ **do**
    if $winner[i] == 1$ **then**
      $PV_{better}[i]_{better} = PV_{better}[i] + step$;
    else
      $PV_{better}[i] = PV_{better}[i] - step$;
    end if
  end for
  ** Exploration **
  generate a solution ind_{new} through PV_{worse};
  ind_{new} = $beta$-step(ind_{new}, ind_{worse}); // see also Algorithm 2
  ** Competition **
  $compete(elite_{worse}, ind_{new})$;
  if $winner == ind_{new}$ **then**
    swap($elite_{worse}, ind_{new}$);
  end if
  ** PV Update **
  for $i = 0; i < PV_{worse}.length; i$++ **do**
    if $winner[i] == 1$ **then**
      $PV_{worse}[i] = PV_{worse}[i] + step$;
    else
      $PV_{worse}[i] = PV_{worse}[i] - step$;
    end if
  end for
  ** Communication **
  $compete(elite_{better}, elite_{worse})$;
  if $winner == elite_{worse}$ **then**
    swap($elite_{better}, elite_{worse}$);
    swap(PV_{better}, PV_{worse});
  end if
  update PV_{worse}; // see also Section 4.2.2
  ** Judge whether the algorithm converges **
  if all elements of PV_{better} and PV_{worse} are equal to 1 or 0 **then**
    break;
  end if
end while
output $elite_{better}$;

5. Final Alignment Determination

In this work, we utilize a greedy heuristic to filter the alignment. First, we remove those correspondences with similarity value lower than 0.88 from the obtained alignment to ensure the precision of the final alignment. Here, we utilize the threshold 0.88 by referring to Fernandez et al. [38]. Then, we sort the resting correspondences by descending similarity, and select them one by one into the final alignment as long as they do not conflict with previous selected ones. In this work, we use a logic reasoning approach to judge whether two correspondences conflict with each other. According to Wang [39], an ontology's entities are often organized by their "is-a" relationships, and a correct alignment should be consistent with that hierarchy. In Figure 3, if the correspondences (a_1, b_1) has high similarity values, that is, a_1 matches b_1, the mapping between a_1's super-concepts a_2 (or sub-concept a_3) and b_1's sub-concept b_3 (or super-concept b_2), that is, (a_2, b_3) and (a_3, b_2) both conflict with (a_1, b_1).

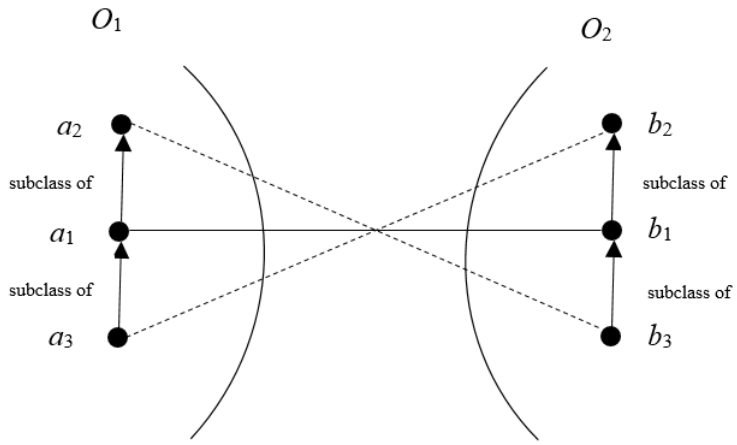


Figure 3. An example of inconsistent mappings.

6. Experiment

6.1. Experimental Configuration

In the experiments, the OAEI's Conference track with ra1 version [40] and three pairs of real sensor ontologies are used to test CcFA's performance. We compare CcFA with two SIA-based ontology matching techniques, that is, EA-based matcher [41], PSO-based matcher [42] and four state-of-the-art sensor ontology matching systems, that is, SOBOM [43], CODI [44], ASMOV [45] and FuzzyAlign [38], whose code and configuration parameters are available online. SOBOM works based on the syntax and structure based similarity measures, and it can obtain better results when the literal of concept and ontology hierarchy structure is complete. CODI utilize the syntax based similarity measure and the Markov Logic based probabilistic model to produce the alignment. ASMOV determines the alignment through an iterative way, and it also takes into consideration three kinds of similarity measure to calculate the similarity values. FuzzyAlign use a fuzzy rule-based strategy to combine three categories of similarity measure, and it utilizes the EA to determine a threshold for filtering the final alignment.

The obtained alignments are evaluated through the f-measure [46], which works based on the golden standard alignment. EA, PSO and CcFA's results are the mean values of thirty independent runs. CcFA use the parameters (see also Section 4.3) which can ensure the highest average alignment quality on all exploited testing cases, EA and PSO utilize the configurations according to their own literatures, and all SIA-based techniques' results are the average of thirty independent runs.

6.2. The Results of Statistical Comparison

The conference track requires matching seven ontologies describing the domain of organizing conferences, that is, Cmt (http://msrcmt.research.microsoft.com/cmt), Pcs (http://precisionconference.com), OpenConf (http://www.zakongroup.com/technology/openconf.shtml), Edas (http://edas.info/), Ekaw (http://ekaw.vse.cz), Iasted (http://iasted.com/conferences/2005/cancun/ms.htm) and Sigkdd (http://www.acm.org/sigs/sigkdd/kdd2006). Three pairs of real sensor ontologies are MMI Device ontology vs SSN ontology, CSIRO sensor ontology vs SSN ontology, and MMI Device ontology vs CSIRO sensor ontology. In particular, MMI Device Ontology (http://mmisw.org/ont/mmi/device) is developed by the Marine Metadata Project (MMP) for promoting the exchange, integration and use of marine data; SSN ontology (https://www.w3.org/TR/vocab-ssn) is developed by the World Wide Web Consortium (W3C) for modeling the knowledge in the sensor network domain; and CSIRO sensor ontology (https://www.w3.org/2005/Incubator/ssn/wiki/SensorOntology2009) is developed by M. Compton et al. from CSIRO (Australia) for providing a semantic description of sensors in terms of the sensor grounding and operation specification. All these sensor ontoloies are all widely used and open to achieve. The reasons that we select SSN ontology, CSIRO sensor ontology and MMI Device ontology in the experiment are: (1) they have defined lots of overlapping information with different representations; (2) SSN is one of the most used global reference ontologies, and it provides the alignment with another upper ontology–DOLCE ultra lite (http://ontologydesignpatterns.org/ont/dul/DUL.owl), which can be used as the golden alignment to measure an alignment's quality, that is, calculate the f-measure value. Table 1 shows the statistical information about the above ontologies.

Table 1. A brief description on the sensor ontologies.

Name	Class Scale	Datatype Property Scale	Object Property Scale
Cmt	36	10	49
Pcs	23	14	24
OpenConf	62	21	24
Edas	104	20	30
Ekaw	77	0	33
Iasted	140	3	38
Sigkdd	49	11	17
MMI Device ontology	55	43	28
SSN ontology	19	2	36
CSIRO sensor ontology	75	11	61

We utilize the Friedman's test [47] and Holm's test [48] to carry out the statistical experiment in terms of the alignments' quality. In particular, Friedman's test [47] is used to figure out whether there are any differences among these competitors, and Holm's test [48] is used to check whether one competitor statistically outperforms others. In Friedman's test, we need to reject the null-hypothesis that all the competitors are equivalent. Therefore, the computed value $\mathcal{X}_r^2$ must be equal to or greater than the tabled critical chisquare value at the specified level of significance α. Here, we choose $\alpha = 0.05$, and we need to consider the critical value for 6 degrees of freedom since we are comparing 7 matchers, that is, $\mathcal{X}_{0.05}^2 = 12.592$.

In Table 2, each value represents the f-measure, and the number in round parentheses is the corresponding computed rank. The computed $\mathcal{X}_r^2 = 119.73$, which is greater than 12.592, and therefore, the null hypothesis is rejected. Then, the Holm's test is further carried out. As shown in Table 2, since our approach ranks with the lowest value, it is set as a control matcher that will be compared with others.

Table 2. Friedman's test on the alignment's quality.

Matching Task	SOBOM	CODI	ASMOV	FuzzyAlign	EA	PSO	CcFA
OAEI's Conference Track							
Cmt-Pcs	0.50(7)	0.75(5)	0.59(6)	0.87(2)	0.82(4)	0.85(3)	**0.91(1)**
Cmt-OpenConf	0.21(7)	0.38(5)	0.28(6)	0.45(3)	0.40(4)	**0.48(1.5)**	**0.48(1.5)**
Cmt-Edas	0.48(6)	0.75(5)	0.42(7)	0.86(1.5)	0.78(3.5)	0.78(3.5)	**0.86(1.5)**
Cmt-Ekaw	0.52(7)	0.75(3)	0.59(6)	0.88(2)	0.66(4)	0.64(5)	**0.92(1)**
Cmt-Iasted	0.54(6)	0.78(5)	0.50(7)	0.87(2.5)	0.82(4)	0.87(2.5)	**0.90(1)**
Cmt-Sigkdd	0.14(7)	0.68(2)	0.34(6)	0.61(4)	0.60(5)	0.64(3)	**0.72(1)**
Pcs-OpenConf	0.40(7)	0.75(4)	0.51(6)	0.86(2)	0.74(5)	0.80(3)	**0.89(1)**
Pcs-Edas	0.44(7)	0.75(5)	0.50(6)	**0.88(1.5)**	0.83(4)	0.86(3)	**0.88(1.5)**
Pcs-Ekaw	0.38(6)	0.70(5)	0.32(7)	0.87(2)	0.83(4)	0.86(3)	**0.90(1)**
Pcs-Iasted	0.54(6)	0.73(5)	0.50(7)	0.89(2)	0.84(3)	0.82(4)	**0.93(1)**
Pcs-Sigkdd	0.40(7)	0.70(5)	0.59(6)	0.80(2)	0.77(3)	0.75(4)	**0.85(1)**
OpenConf-Edas	0.14(7)	0.36(5)	0.27(6)	0.59(2)	0.50(3)	0.56(4)	**0.67(1)**
OpenConf-Ekaw	0.25(7)	0.41(4.5)	0.28(6)	0.52(2)	0.44(3)	0.41(4.5)	**0.54(1)**
OpenConf-Iasted	0.20(7)	0.70(5)	0.31(6)	0.71(4)	**0.79(1.5)**	0.75(3)	**0.79(1.5)**
OpenConf-Sigkdd	0.35(7)	0.61(5)	0.48(6)	0.82(2.5)	0.82(2.5)	0.79(4)	**0.85(1)**
Edas-Ekaw	0.10(7)	0.25(6)	0.39(4)	0.44(2)	0.34(5)	0.41(3)	**0.54(1)**
Edas-Iasted	0.32(6)	0.72(3)	0.20(7)	0.84(2)	0.62(5)	0.70(4)	**0.88(1)**
Edas-Sigkdd	0.25(7)	0.58(5)	0.33(6)	0.66(3)	0.69(2)	0.65(4)	**0.73(1)**
Ekaw-Iasted	0.30(7)	0.64(5)	0.37(6)	0.87(2)	0.81(4)	0.82(3)	**0.94(1)**
ekaw-Sigkdd	0.22(7)	0.78(3)	0.25(6)	**0.74(1.5)**	0.70(4)	0.68(5)	**0.74(1.5)**
Iasted-Sigkdd	0.17(7)	0.72(5)	0.20(6)	0.77(2)	0.74(3.5)	0.74(3.5)	**0.83(1)**
Three Pairs of Real Sensor Ontologies							
MMI Device-SSN	0.77(7)	0.80(5)	0.73(6)	0.88(2)	0.82(4)	0.85(3)	**0.92(1)**
CSIRO-SSN	0.78(5.5)	0.79(4)	0.75(6)	0.88(2)	0.78(5.5)	0.87(3)	**0.94(1)**
MMI Device-CSIRO	0.72(7)	0.78(4)	0.75(5)	0.87(2)	0.74(6)	0.82(3)	**0.90(1)**
Average	0.38 (6.72)	0.66 (4.52)	0.43 (6.08)	0.76 (2.22)	0.70 (3.85)	0.72 (3.43)	**0.89 (1.10)**

In Holm's test, $z-$value is the testing statistic for comparing the ith and jth matchers, which is used for finding the p-value (the corresponding probability from the table of the normal distribution). p-value is then compared with $\alpha = 0.05$, which is an appropriate level of significance. According to Table 3, we can state that CcFA statistically outperforms other competitors on f-measure at 5% significance level. In Figure 4, we compare CcFA with OAEI's participants (http://oaei.ontologymatching.org/2019/results/conference/index.html) in terms of average f-measure.

Table 3. Holm's test on the alignment's quality.

i	approach	$z-$value	unadjusted $p-$value	$\frac{\alpha}{k-i}, \alpha = 0.05$
8	FuzzyAlign	2.22	0.030	0.050
7	EA	3.43	1.91×10^{-4}	0.025
5	PSO	3.85	1.08×10^{-5}	0.012
3	CODI	4.52	4.25×10^{-8}	0.008
2	ASMOV	6.08	1.55×10^{-15}	0.007
1	SOBOM	4.45	4.24×10^{-17}	0.006

As can be seen from the tables and figure, the f-measure values obtained by CcFA outperform all the other competitors, which shows that CcFA can effectively optimize the ontology alignments. In particular, the quality of alignment of CcFA is better than EA and PSO, which shows that CcFA's compact encoding mechanism and compact operators can better trade off the algorithm's exploration and exploitation. Since none of the similarity measures can effectively distinguish all the heterogeneous concepts in any situations, it is necessary to aggregate several similarity measures to improve the result's precision. We utilize a hybrid similarity measure which combines three kinds of similarity measures to calculate the entity similarity value, and therefore CcFA's results are significantly higher

than other systems that only take into consideration one or two categories of similarity measure, such as SOBOM, CODI, DOME, Lily, ALin, LogMap family and Wiktionary. However, FuzzyAlign, ASMOV, AML, SANOM and ONTMAT1 apply too many similarity measures that lead to the conflicting results, which decreases the recall value. Thus, how many similarity measures should be selected and combined to ensure the quality of the alignment will be one of our future work.

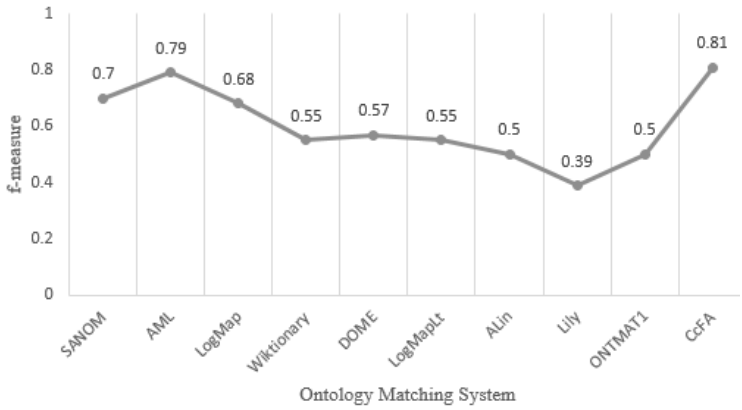


Figure 4. Comparison with state-of-the-art ontology matching systems on the Conference track.

Currently, there is a new trend for developing lightweight sensor ontologies, such as the Sensor, Observation, Sample, and Actuator (SOSA) ontology [49], which only provides the specification for the kernel SSN's entities that involves in the acts of observation, actuation and sampling, and IoT-Lite [50]. CcFA can also represent an efficient approach for matching these lightweight sensor ontologies since they own less entities and the search space is relatively smaller. However, since the entities' semantic relationships in the lightweight ontologies could be more complicated and richer, lightweight sensor ontology alignment is more complex, that is, one source ontology entity is mapping with more than one target ontology entity, and the relationships could be equivalence or subsumption. To address this complex matching problem [51], a feasible method is to introduce various mapping patterns [52] into CcFA to detect the complex correspondences, which is one of our future work.

7. Conclusions and Future Work

In order to support the information integration of various sensor ontologies, in this paper, a general-purpose ontology matching technique based on CcFA is proposed. Our proposal makes use of compact α-step and β-step to implement the discrete exploitation and exploration, and trade off them during the evolving process through two PVs. The experiment shows the effectiveness of this combination of compact encoding mechanism and co-Evolutionary mechanism, and our approach statistically outperforms other competitors on alignment's quality at 5% significance level when matching sensor ontologies and other ontologies in the domain of organizing conferences.

In the future, we will further study the technique that can adaptively select and combine various similarity measures according to different heterogeneity situation. Moreover, we will improve CcFA based approach to match the large-scale sensor ontologies, which is a challenge in the ontology matching domain. Another challenge in ontology matching domain is the problem of Instance Coreference Resolution (ICR) [53] in the sensor network domain, which requires matching large-scale sensor instances in the Linked Open Data cloud (LOD). Currently, there is no SIA-based technique that could effectively solve ICR, and we are also interested in addressing this challenge with CcFA. In this work, we mainly aim at matching sensor ontologies, as well as other general ontologies, such as those in the domain of organizing conferences. With respect to the specific ontologies like biomedical ontology and geographical ontology, directly applied our proposal to match them could yield low precision and recall because these tasks require specific background knowledge base and complex

forms of alignment. Therefore, we would like to extend the CcFA-based matching technique to address these matching tasks in the specific domains.

Author Contributions: Investigation, J.C.; Methodology, X.X.; Writing—original draft, X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Nos. 61503082 and 61403121), the Natural Science Foundation of Fujian Province (No. 2016J05145), the National Key R&D Program of China (No. 2018YFC0407101), the Fundamental Research Funds for the Central Universities (No. 2019B22314), the Program for New Century Excellent Talents in Fujian Province University (No. GY-Z18155), the Program for Outstanding Young Scientific Researcher in Fujian Province University (No. GY-Z160149) and the Scientific Research Foundation of Fujian University of Technology (Nos. GY-Z17162 and GY-Z15007).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Delin, K.A.; Jackson, S.P. Sensor web: a new instrument concept. In Proceedings of the SPIEs Symposium on Integrated Optics, San Jose, CA, USA, 20–26 January 2001.
2. Corcho, O.; García-Castro, R. Five challenges for the semantic sensor web. *Semantic Web* **2010**, *1*, 121–125. [CrossRef]
3. Wang, X.; Zhang, X.; Li, M. A survey on semantic sensor web: sensor ontology, mapping and query. *Int. J. u-e-Serv. Sci. Technol.* **2015**, *8*, 325–342.
4. Sheth, A.P.; Henson, C.; Sahoo, S.S. Semantic sensor web. *IEEE Internet Comput.* **2008**, *12*, 78–83. [CrossRef]
5. Xue, X.; Pan, J.S. A Compact Co-Evolutionary Algorithm for sensor ontology meta-matching. *Knowl. Inf. Syst.* **2017**, *56*, 335–353. [CrossRef]
6. Acampora, G.; Loia, V.; Vitiello, A. Enhancing ontology alignment through a memetic aggregation of similarity measures. *Inf. Sci.* **2013**, *250*, 1–20. [CrossRef]
7. Harik, G.R.; Lobo, F.G.; Goldberg, D.E. The compact genetic algorithm. *IEEE Trans. Evol. Comput.* **1999**, *3*, 287–297. [CrossRef]
8. Baraglia, R.; Hidalgo, J.I.; Perego, R. A hybrid heuristic for the traveling salesman problem. *IEEE Trans. Evol. Comput.* **2001**, *5*, 613–622. [CrossRef]
9. Ahn, C.W.; Ramakrishna, R.S. Elitism based compact genetic algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 367–385.
10. Neri, F.; Mininno, E.; Karkkainen, T. Noise analysis compact genetic algorithm. In Proceedings of the 2010 European Conference on the Applications of Evolutionary Computation, Istanbul, Turkey, 7–9 April 2010; pp. 602–611.
11. Neri, F.; Mininno, E.; Iacca, G. Compact particle swarm optimization. *Inf. Sci.* **2013**, *239*, 96–121. [CrossRef]
12. Mininno, E.; Neri, F.; Cupertino, F.; Naso, D. Compact differential evolution. *IEEE Trans. Evol. Comput.* **2010**, *15*, 32–54. [CrossRef]
13. Tan, K.; Yang, Y.; Goh, C. A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Trans. Evol. Comput.* **2006**, *10*, 527–549. [CrossRef]
14. Mu, C.; Jiao, L.-C.; Liu, Y. M-Elite coevolutionary algorithm for numerical optimization. *Journal of software* **2009**, *20*, 2925–2938. [CrossRef]
15. Wang, X.; Liu, Q.; Fu, Q.; Zhang, L. Double elite coevolutionary genetic algorithm. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 67–75.
16. El-Abd, M. Cooperative coevolution using the brain storm optimization algorithm. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016.
17. Wang, H.; Wang, W.; Cui, L.; Sun, H.; Zhao, J.; Wang, Y.; Xue, Y. A Hybrid Multi-Objective Firefly Algorithm for Big Data Optimization. *Appl. Soft Comput.* **2018**, *69*, 806–815. [CrossRef]
18. Karthikeyan, S.; Asokan, P.; Nickolas, S.; Page, T. A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. *Int. J. Bio-Inspired Comput.* **2015**, *7*, 386–400. [CrossRef]
19. Alboukaey, N.; Joukhadar, A. Ontology Matching as Regression Problem. *J. Digit. Inf. Manag.* **2018**, *16*, 1.
20. Khoudja, M.A.; Fareh, M.; Bouarfa, H. Ontology Matching using Neural Networks: Survey and Analysis. In Proceedings of the 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, 24–25 November 2018.

21. Dhouib, M.T.; Zucker, C.F.; Tettamanzi, A.G. An Ontology Alignment Approach Combining Word Embedding and the Radius Measure. In *International Conference on Semantic Systems*; Springer: Cham, Switzerland, 2019; pp. 191–197.
22. Assi, A.; Mcheick, H.; Karawash, A.; Dhifli, W. Context-aware instance matching through graph embedding in lexical semantic space. *Knowledge-Based Syst.* **2019**, *186*, 104925. [CrossRef]
23. Ali, F.; Kwak, K.S.; Kim, Y.G. Opinion mining based on fuzzy domain ontology and Support Vector Machine: A proposal to automate online review classification. *Appl. Soft Comput.* **2016**, *47*, 235–250. [CrossRef]
24. Xue, X.; Pan, J.S. A segment-based approach for large-scale ontology matching. *Knowl. Inf. Syst.* **2017**, *52*, 467–484. [CrossRef]
25. Amrouch, S.; Mostefai, S.; Fahad, M. Decision trees in automatic ontology matching. *Int. J. Metadata, Semant. Ontol.* **2016**, *11*, 180–190. [CrossRef]
26. Araújo, T.B.; Pires, C.E.S.; da Nóbrega, T.P.; Nascimento, D.C. A fine-grained load balancing technique for improving partition-parallel-based ontology matching approaches. *Knowledge-Based Syst.* **2016**, *111*, 17–26. [CrossRef]
27. Amin, M.B.; Khan, W.A.; Hussain, S.; Bui, D.M.; Banos, O.; Kang, B.H.; Lee, S. Evaluating large-scale biomedical ontology matching over parallel platforms. *Iete Tech. Rev.* **2016**, *33*, 415–427. [CrossRef]
28. Martinez-Gil, J.; Montes, J.F.A. Evaluation of two heuristic approaches to solve the ontology meta-matching problem. *Knowl. Inf. Syst.* **2011**, *26*, 225–247. [CrossRef]
29. Ginsca, A.L.; Iftene, A. Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment. In Proceedings of the 9th Roedunet International Conference, Sibiu, Romania, 24–26 June 2010; pp. 118–122.
30. Xue, X.; Wang, Y. Optimizing ontology alignments through a Memetic Algorithm using both MatchFmeasure and Unanimous Improvement Ratio. *Artif. Intell.* **2015**, *223*, 65–81. [CrossRef]
31. He, Y.; Xue, X.; Zhang, S. Using artificial bee colony algorithm for optimizing ontology alignment. *J. Inf. Hiding Multimed. Sig. Process* **2017**, *8*, 766–773.
32. Xue, X.; Chen, J. A Compact co-Firefly Algorithm for Matching Ontologies. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 2629–2632.
33. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
34. Xue, X. A compact firefly algorithm for matching biomedical ontologies. *Knowl. Inf. Syst.* **2020**, 1–17. [CrossRef]
35. Shu-Chuan, C.; Xue, X.; Pan, J.S.; Wu, X. Optimizing Ontology Alignment in Vector Space. *J. Internet Tech.* **2020**, *21*, 15–22.
36. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
37. Kondrak, G. N-gram similarity and distance. In Proceedings of the International symposium on string processing and information retrieval, Buenos Aires, Argentina, 2–4 November 2005; pp. 115–126.
38. Fernandez, S.; Marsa-Maestre, I.; Velasco, J.R.; Alarcos, B. Ontology alignment architecture for semantic sensor web integration. *Sensors* **2013**, *13*, 12581–12604. [CrossRef]
39. Wang, P. Lily-LOM: An efficient system for matching large ontologies with non-partitioned method. In Proceedings of the 2010 International Conference on Posters & Demonstrations Track, Shanghai, China, 9 November 2010; pp. 69–72.
40. Zamazal, O.; Svátek, V. The ten-year ontofarm and its fertilization within the onto-sphere. *J. Web Semant.* **2017**, *43*, 46–53. [CrossRef]
41. Wang, J.; Ding, Z.; Jiang, C. GAOM: genetic algorithm based ontology matching. In Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06); Guangzhou, China, 12–15 December 2006; pp. 617–620.
42. Bock, J.; Hettenhausen, J. Discrete particle swarm optimisation for ontology alignment. *Inf. Sci.* **2012**, *192*, 152–173. [CrossRef]
43. Xu, P.; Wang, Y.; Cheng, L.; Zang, T. Alignment Results of SOBOM for OAEI 2010. *Ontol. Matching* **2010**, *203*, 7–11.
44. Noessner, J.; Niepert, M.; Meilicke, C.; Stuckenschmidt, H. Leveraging terminological structure for object reconciliation. In Proceedings of the Extended Semantic Web Conference, Heraklion, Greece, 30 May–3 June 2010; pp. 334–348.

45. Jean-Mary, Y.R.; Shironoshita, E.P.; Kabuka, M.R. Ontology matching with semantic verification. *J. Web Semant.* **2009**, *7*, 235–251. [CrossRef]
46. Rijsberge, C.J.V. *Information Retrieval*; University of Glasgow: London, UK, 1975.
47. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
48. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **1979**, 65–70.
49. Janowicz, K.; Haller, A.; Cox, S.J.; Le Phuoc, D.; Lefrançois, M. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *J. Web Semant.* **2019**, *56*, 1–10. [CrossRef]
50. Bermudez-Edo, M.; Elsaleh, T.; Barnaghi, P.; Taylor, K. IoT-Lite: A lightweight semantic model for the Internet of Things. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 90–97.
51. Thiéblin, E.; Haemmerlé, O.; Hernandez, N.; Trojahn, C. Survey on complex ontology matching. *Semant. Web* **2019**, 1–39. [CrossRef]
52. Ritze, D.; Meilicke, C.; Šváb-Zamazal, O.; Stuckenschmidt, H. A pattern-based ontology matching approach for detecting complex correspondences. In Proceedings of the ISWC Workshop on Ontology Matching, Chantilly, VA, USA, 25 October 2009; pp. 25–36.
53. Xue, X.; Wang, Y. Using memetic algorithm for instance coreference resolution. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 580–591. [CrossRef]