Amirkabir University of Technology
(Tehran Polytechnic)
Vol. 45 - No. 2 - Fall 2013, pp. 1- 10

Amirkabir International Journal of Science & Research
(Modeling, Identification, Simulation & Control)
(AIJ-MISC)

Centralized Clustering Method To Increase Accuracy In Ontology Matching Systems

Samira Babalou[1], Mohammad Javad Kargar[2*], and Seyyed Hashem Davarpanah[3]

1- MSC student, Department of Computer Engineering, Faculty of Engineering, University of Science and Culture, Tehran, Iran
2- Assistant Professor, Department of Computer Engineering, Faculty of Engineering, University of Science and Culture, Tehran, Iran
3- Assistant Professor, Department of Computer Engineering, Faculty of Engineering, University of Science and Culture, Tehran, Iran

ABSTRACT

Ontology is the main infrastructure of the Semantic Web which provides facilities for integration, searching and sharing of information on the web. Development of ontologies as the basis of semantic web and their heterogeneities have led to the existence of ontology matching. By emerging large-scale ontologies in real domain, the ontology matching systems faced with some problem like memory consumption. Therefore, partitioning the ontology was proposed. In this paper, a new clustering method for the concepts within ontologies is proposed, which is called SeeCC. The proposed method is a seeding-based clustering method which reduces the complexity of comparison by using clusters' seed. The SeeCC method facilitates the memory consuming problem and increases their accuracy in the large-scale matching problem as well. According to the evaluation of SeeCC's results with Falcon-AO and the proposed system by Algergawy accuracy of the ontology matching is easily observed. Furthermore, compared to OAEI (Ontology Alignment Evaluation Initiative), SeeCC has acceptable result with the top ten systems.

KEYWORDS

Ontology matching, Clustering method, Large-scale matching, Semantic graph.

*Corresponding Author, Email: s.Babaloo@son.ir

1. INTRODUCTION

Ontologies are main structures of the Semantic Web, which provide facilities for integration, searching and sharing of information on the web through making understandable the existing information for machines [1]. Despite this critical role and due to the creation of ontologies by different people or methods even if for the same domain, there exists some heterogeneity. In order to resolve this problem, ontology matching systems were created.

Nowadays, there are many large-scale ontologies in the real domains such as the medical science domain. But, to process these large-scale ontologies, the existing ontology matching tools have some problems such as shortage of consumed memory or long time consumption that are real challenges [2]. For example, in the OAEI competition held in 2011 in large ontologies test with 2000-30000 classes, only 6 of 16 systems could process those ontologies [3]. In order to make able matching of large-scale ontologies, dividing the ontologies to some partitions is a way which has been proposed so far via the methods such as divide and conquer[4], clustering[5], and modularization[6]. In this paper, we propose a seeding-based clustering method for partitioning ontologies. In this method, a seed is defined for each cluster according to a Ranker phase. This method reduces complexity of the comparisons by comparing concepts with only seeds instead of all the other concepts. The seeding-based clustering also was used in wireless sensor network for efficient energy utilization [7, 8].

One of the main parts of a large-scale ontology matching system is the partitioning phase so that correctly conduction of this step increases the accuracy. According to the belief of Saruladha and et al. [9] that clustering ontology is the key solution for managing scalability issues for the ontologies, and Zhou et al. [10] believed that graph clustering methods are very useful for identifying densely connected groups in a large graph. This led us to use clustering approach for this phase of our method.

Our contribution is the use of seeding-based clustering method in ontology domain. This matter makes possible working on real large ontologies even using normal processors. We apply SeeCC to large ontologies in the anatomy test. This method divides large-scale ontology to several sub ontologies. First, all the concepts are ranked according to Ranker and ReRanker functions. Second, seed of clusters is determined based on the top highest ranked concepts by a distribution condition. Finally, the new membership function is used to cluster the remaining concepts.

The rest of the paper is organized as follows: Section II discusses the large-scale ontology matching problem. Section III explains the SeeCC method (Seeding-based Clustering Concepts), and section IV presents the experimental results of SeeCC with Falcon-AO [11] and a method proposed by Algergawy et al. [5]. The results are also compared with OAEI results. Section V concludes the paper.

2. REVIEW OF LITERATURE

Different architectures have been proposed for large-scale ontology matching systems [4, 5, 12]. Most of these methods have three main stages: (1) partitioning the large ontology to several sub-ontologies, (2) applying matching method to each pair of sub-ontologies, and (3) combining the results. We present a general architecture for a large-scale ontology matching in Figure (1). First, two ontologies enter and then partitioning methods are applied in order to divide the input ontologies into a number of sub-ontologies. After finding similar pairs of sub-ontologies, the matching method is applied and finally, partial results are combined.

When two large ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$are partitioned in to several sub-ontologies, at the simplest way, each sub-ontology of $\mathcal{O}_1$ is matched with each sub-ontology of $\mathcal{O}_2$. In this way, if $\mathcal{O}_1$ is partitioned into n sub-ontologies and $\mathcal{O}_2$ is partitioned into m sub-ontologies, in the worst case, n×m matchings must be done, which is a second order function. Therefore, for reducing this order, most of systems like [4, 5] proposed that only similar sub-ontologies should be matched; i.e, the pair of very dis-similar sub-ontologies are removed from the matching process. This phase is also known as the filtering phase.

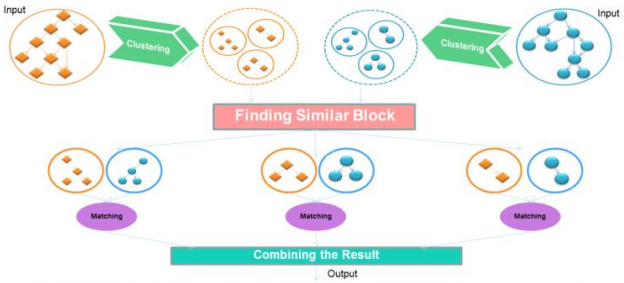

Fig. 1. Overall Architecture of a Large-Scale Ontology Matching System.

One of the main parts of large-scale ontology matching is the partitioning part. If it has done correctly, the accuracy goes up in the next phase. To partition ontologies, systems use approaches such as clustering (Algergawy et al. [5]), divide and conquer (Hu et al. [4], COMA++[12] and PBM[13]), and modularization

(MOM[6] and LogMap[14]). Divide and conquer method recursively breaks the large-scale ontologies into sub-ontologies, and clustering approach clusters related components to the same clusters. In addition to connected components, each module has an encapsulation property; hence, the "subclass" and "part-off" relations could not be in different modules. Furthermore, other approaches in some systems such as the parallelization via fragment-level (GOMMA[15]) and the machine learning and learning model (YAM++[16]). We show these methods in Table (1), which are explained in the "Division Strategy" column. The input, output, and Graphic User Interface (GUI) are respectively shown in columns 4, 5, and 6. The last column shows methods that system uses to match the concepts of ontologies

3. SEECC METHOD

As shown in Figure (2), SeeCC consists of three components. In pre-processing, ontologies are parsed and the number of cluster heads is determined. In Ranker component, Ranker and ReRanker functions are applied to ontologies in order to score concepts of an ontology. And, in the clustering component, first Cluster Heads (CH) are determined, then the remaining concepts are placed in the corresponding clusters according to a specified membership function

TABLE 1. COMPARISON OF THE MATCHING SYSTEMS OF LARGE ONTOLOGIES

No	Systems	Division Strategy	Input	Output	GUI	Matching method
1	LogMap[14]	Module extraction [17]	OWL	1:1	✓	Logic base
2	GOMMA[15]	Parallelization via fragment-level	OBO, OWL, RDF	1:1	×	Combination of name, comment and instance matcher
3	YAM++[16]	Machine learning and learning model	Graph data structure	1:1	✓	Combination of element and structural level and semantical matcher
٤	MOM[6]	Modualization	Transform OWL ontology in to E-Connections	1:1	×	Use OPM(Ontology Parsing graph-based Mapping)[١٨]
5	COMA++[12]	Divide & Conquer	XML, OWL	1:1	✓	Using a library of more than 15 matchers
٦	Algergawy et al [5]	Clustering based AHSCAN[19]	XML,RDFS, OWL	1:1 1:n	✓	COMA++
7	Hu et al[4]	Divide & conquer via Rock[20] algorithm	RDFS, OWL	1:1	-	GMO , V-Doc
8	PBM[13]	Inspired by ROCK[20] based on structural and linguistic proximity	RDFS, OWL	1:1	-	Matching block via anchors and virtual documents and TF/IDF

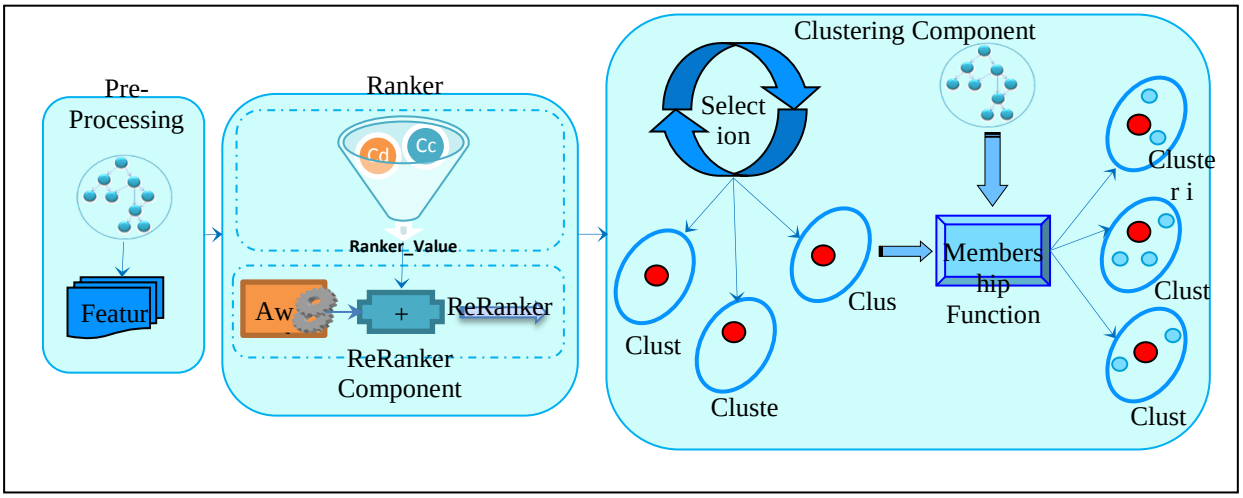

Fig. 2. Architecture of the SeeCC Method

In this paper, we use the following notations: Concept-Related Graph $\mathcal{g} = (\mathcal{C}, \mathcal{R}, \mathcal{L})$ is a labeled directed graph. $\mathcal{C} = \{c_1, \dots, c_n\}$ is a finite set of nodes presenting the concepts of ontology. $\mathcal{R} = \{r_1, \dots, r_m\}$ stands for a finite set of directed edges showing all relations between concepts in $\mathcal{O}$. $r_k \in \mathcal{R}$ denotes a directed relation between two adjacent concepts $c_i, c_j \in \mathcal{C}$ i.e. $r_k = (c_i, c_j)$. $\mathcal{L} = \{\ell_1, \dots, \ell_m\}$ is a finite set of labels of graph nodes that show the name of each concept. n is the number of nodes (concepts) and m is the number of edges (relationships) in $\mathcal{g}$. Also matrix $\mathcal{M}$ is generated from $\mathcal{g}$ to calculate central measures in the Ranker phase. In the $\mathcal{M}$ matrix as defined in Eq. 1, if a connection exists between two (c_i, c_j) vertices, each member of $\mathcal{M}$ i.e. (m_i, m_j) will be equal to one, otherwise, will be zero.

$$\mathcal{M} = \begin{cases} (m_i, m_j) = 1 \mid r_k = (c_i, c_j), r_k \in \mathcal{R}, m_i \in \mathcal{M}; i, j \leq n \\ O.W. \qquad 0 \end{cases} \quad (1)$$

The goal of clustering algorithm to partition vertices (V) to a set of separate clusters $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n$, so that cohesion of vertices in one cluster is high, while the coupling of two $\mathcal{T}_i$and $\mathcal{T}_j$ clusters is low. Eqs. (2) and (3) show that each two clusters do not have shared concepts and union of all clusters equals to the main ontology.

$$\forall\, \mathcal{T}_i, \mathcal{T}_j, i, j = 1, 2, \dots, n \; and \; i \neq j, \mathcal{T}_i \cap \mathcal{T}_j = 0 \quad (2)$$

$$\mathcal{T}_1 \cup \mathcal{T}_2 \cup \dots \mathcal{T}_n = \mathcal{O} \quad (3)$$

Although partitioning method is less time-consuming, as it was previously mentioned, instead of matching all the concepts of the large ontology at the final step, only similar clusters are matched which leads to less amount of time and calculations. SeeCC is constructed of the following four phases.

A. Phase 1: Pre-Processing

In the implementation of our new concept-related graph, we parsed and inferred ontology by Apache Jena[1] and then the concept-related graph is drawn by mapping the inferred result. The number of concepts in an ontology is the calculated as well as the number of cluster heads (i.e. $\mathcal{K}$) are automatically determined . As shown in Eq. (4), the number of concepts is divided by ε as the maximum size of each cluster with $\varepsilon < |\mathcal{O}|$.

$$\mathcal{K} = \frac{|\mathcal{O}|}{\varepsilon} \quad (4)$$

Effect on the Optimal Number of Cluster Head:

We have done one test for determining optimal $\mathcal{K}$. In this test, for different value of ε, we calculate the F-measure of final matching result in Conference and Anatomy datasets. In Table (2), the first column shows average of F-measure on the conference dataset, and the second column shows the F-Measure on the Anatomy dataset. According to this test, the best ε value for the Conference and Anatomy dataset are 300 and 600, respectively. We selected ε =600 among the 600-1000 size in the Anatomy dataset, because according to Hamdi et al.[21] we must tradeoff between the number of blocks and their accuracy.

TABLE 2. EFFECT ON OPTIMAL NUMBER OF CLUSTER HEAD

Size	*Conference dataset*	*Anatomy dataset*
$\varepsilon = 100$	*0.609*	***0.824***
$\varepsilon = 200$	*0.598*	***0.827***
$\varepsilon = 300$	*0.620*	***0.831***
$\varepsilon = 400$	*0.617*	***0.829***
$\varepsilon = 500$	*0.615*	***0.832***
$\varepsilon = 600$	*0.609*	***0.835***
$\varepsilon = 700$	*0.608*	***0.834***
$\varepsilon = 800$	*0.608*	***0.835***
$\varepsilon = 900$	*0.608*	***0.834***
$\varepsilon = 1000$	*0.608*	***0.834***

B. Phase 2: Ranking Of Concepts

In the seeding-based clustering algorithms, heads of clusters are selected as the nodes with an important role. Zhang [22] suggested a concept with crucial role as an "important" node. The "important" in the SeeCC is defined by theoretical graph metrics (in the Ranker function) and the effect of neighbors of a node (in the ReRanker function)

i. Ranker Function:

The importance of a node in a semantic graph is understandable through its edges [23]. This matter leads us to use graph-theoretic measures based on graph connections in the Ranker function. The definition of "centrality" measure on the vertices in a graph is derived from the social network analysis. Each person is given a score based on his or her position at the network showing the importance of each individual. Centrality measure is also used in reply to the quarries so that a central node with more accessibility to the other nodes is desirable.

In this section the following different centrality measures are defined:

1 Degree Centrality [24]: This is the simplest measure that calculates the number of connections of a vertex. In a directed graph, there is an in-degree and out-degree centrality that calculates the number of input and output links, respectively. The link between vertices can

[1] https://jena.apache.org/

be considered as an authority. Vertices in a graph with high degree of centrality are certainly more prominent than the others, because they have received a great deal of power. Eq. (5) shows the degree centrality.

$$C_d(i) = \text{degree}Centrality(c_i) \quad (5)$$

2 Betweenness Centrality [25]: This measure calculates the fraction of the shortest path from a vertex. Eq. (6) shows betweenness centrality, in which $\sigma_{s,t}(v)$ is the number of shortest paths from s to t through v, and $\sigma_{s,t}$ is the total number of shortest paths from s to t.

$$C_b(v) = \sum_{s,t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \quad (6)$$

3 . Closeness Centrality [24]: This measure shows the importance of nodes which are close to all other ones in the graph. In Eq (7) reaching cost of one node to all nodes of the graph is measured. In Eq. (7), the distance (i,j) function is the shortest path between i and j nodes in the graph.

$$C_c(i) = \frac{1}{\sum_{j \in V} \text{distance}(i,j)} \quad (7)$$

4. EcCentrality [26]: This measure calculates the maximum distance between pairs of nodes. The intuition is that one node is the central one if no node is away from it. It is calculated via Eq (8) given bellow.

$$C_e(i) = \frac{1}{\max_{j \in V} distance(i,j)} \quad (8)$$

5 . Stress Centrality [27]: This measure calculates the absolute number of the shortest paths through a node. Eq. (9) shows how to calculate the Stress Centrality where $\sigma_{st}(v)$ represents the number of the shortest paths from s to t via v.

$$SC(v) = \sum_{s,t \in V} \sigma_{st}(v) \quad (9)$$

To combine these criteria in the Ranker function, one test is conducted. We performed an evaluation using three different ontologies: “Linking”, “MICRO”, and “cmt” of Conference dataset of OAEI. In this test, all 32 combinations of the five criteria were assessed. Our goal is to define a method able to generate results that match as closely as possible to those produced by human experts. We asked a number of experts to select the top ten important concepts while we did not say anything about our criteria to them. Due to the difference between important concepts by experts, we selected the most common shared important concepts. Table 4 shows the top ten important concepts by experts. The results of 32 combinations of these criteria on Linking ontology has shown in Table 3. In each test we use one combination of C1-C5 criteria and select top ten important concepts, we also examine how many of these criteria are similar to the expert judge. The test examines which combination was more similar to the experts’ point of view. As a whole, we separate these tests by their accuracy. Among those criteria with accuracy 80%, we selected C1+C2 (Degree and Closeness centrality) because its performance was proven and it was used by [23, 28]. Furthermore, its computation is less complex and this selection reduces the computational complexity.

Therefore, the score of each node in this function is calculated by Eq. (10).

$$\text{Ranker\_Score}(c_i) = \frac{1}{\sum_{j \in V} \text{distance}(i,j)} + degreeCentrality(c_i) \quad (10)$$

where, $degreeCentrality(c_i)$ is the out degree of i node, and distance (i ,j) is the shortest path between nodes i and j in the graph.

Table 5 shows the top ten important concepts on three tested ontologies by the proposed Ranker function.

TABLE 3. THE ACCURACY OF COMBINATION OF THE FIVE CRITERIA ON "LINKING" ONTOLOGY

No	*Combination of Centrality Criteria*	*Accuracy*
1	*C1+C2, C1+C3, C1+C4, C1+C5, C1+C2+C4, C1+C2+C5, C1+C3+C4, C1+C3+C5, C1+C4+C5, C1+C2+C3+C4+C5*	***80%***
2	*C1, C3, C5, C2+C3, C2+C5, C3+C5, C4+C5, C1+C2+C3, C2+C3+C5, C3+C4+C5, C1+C2+C3+C4, C1+C2+C3+C5, C1+C2+C4+C5, C1+C3+C4+C5, C2+C3+C4+C5*	***70%***
3	*C2, C4, C2+C4*	***20%***
4	*C2+C3+C4, C2+C4+C5*	***50%***

TABLE 4. TOP IMPORTANT CONCEPTS BY EXPERTS

NO	*Ontology*	*Top Ten Important Concepts*
1	*Linkling*	***Role- Content- Person- submissionStatus- Submission- Event- SubmissionType- Setting- FullText- RegisteredPerson.***
2	*MICRO*	***Conference- Topic- ActivitySubmissionForm- Activity- OrganizingCommittee- Author- Organizer- Reviewer- Lecture- Person.***
3	*Cmt*	***Conference, Author, Reviewer, conferenceMember, Paper, Person, Document, Preference, Decision, Bid.***

2) ReRanker Function

In the ReRanker function, the effect of neighbors is calculated. For this purpose, the method proposed by Stuckenschmidt [29] is used, in which the network analysis technique is employed to determine the power of relations between nodes of a graph. By using this method,

nodes with fewer relations to the other nodes are assigned with lower scores. Accordingly, the current score of each node is divided into the number of its direct children and then this value is added to the score of all direct children as an award as it is shown in Eqs. (11) and (12).

$$ReRanker\_Score = Award(c_i) + Ranker\_Score\ (c_i) \quad (11)$$

TABLE 5. TOP TEN IMPORTANT CONCEPTS BY THE PROPOSED RANKER FUNCTION

No	Ontology	Top Ten Important Concepts	Accuracy
1	*Linkling*	*Role, SubmissionType, Person, FullText, RegisteredPerson, Content, Event, Settings, Place, Abstract*	***80%***
۲	*MICRO*	*OrganizingCommittee, WorkShop, Organizer, Reviewer, Person, Author, OutsideReferee, TutorialProposal, ActivitySubmissionForm, Activity*	***70%***
۳	*Cmt*	*Person, Conference, Preference, ProgramCommittee, Document, Decision, Bid, AuthorNotReviewer, Reviewer, ProgramCommitteeMember*	***70%***

TABLE 6. NECESSITY OF RERANKER FUNCTION

No	without ReRank	ReRank by d=1	ReRank by d=2
Average	*0.6134*	*0.6147*	***0.6119***

where

$$Award(c_i) = \frac{Ranker\_Score\ (c_i)}{|\Psi(c_i, d)|} \quad (12)$$

$Ranker\_Score\ (c_i)$ is defined in the Eq (5) and d is set to 1 in order to calculate the direct children of c_i node. $\Psi(c_i, d)$ is $Connexion$ that is explained in phase 3.

Also, according to another test that has shown in Table 6 on the conference dataset of OAEI, adding award to the children in the next non-immediate levels (such as grandchildren), i.e. ReRanke with 2 levels would reduce the accuracy of ontology matching.

C. Phase 3: Determining Cluster Head

If nodes with the highest score are selected as the cluster heads, distribution would be disregarded. To avoid this problem, the distance between two cluster heads is measured, and among the highest score nodes, those with d distances from each other are selected as the cluster heads. For this purpose, a $Connexion$ set with d levels of each node is defined. One node can be selected as a cluster head if it does not exist in the $Connexion$ set of their previous cluster heads. $Connexion$ set of a concept $c_i \in \mathcal{C}$, namely$\Psi(c_i, d)$ is defined in Eq. (13).

$$\Psi(c_i, d) = SubClass(c_i, d) \cup SuperClass(c_i, d)\} \quad (13)$$

$\Psi(c_i, d)$ is a set in which all the concepts with d levels that effect on node c_i. Here, the $SubClass(c_i, d)$ is the children of c_i with d hierarchical levels, and $SuperClass(c_i, d)$ is the parent of c_i with d hierarchical levels. The $Connexion$ set is also used in the membership function.

D. Phase 4: Finalise Clustering

At first, SeeCC creates one cluster for each cluster head. Then, it places direct children in the corresponding cluster and finally, for the remaining nodes, the membership function is used to determine the cluster of each node. Step two reduces the time complexity, because fetch of membership function for all nodes is time consuming. While by placing the nodes via the call of membership function, the same results would still be achieved. Because the membership function of SeeCC in the structural similarity measure considers the shortest path between each node and cluster heads, each child has the minimum shortest path from his or her parent.

1) Membership Function

For all the concepts in the ontology, $\mathcal{F}$ is used as a membership flag of a cluster. If $\mathcal{F}$ of c concept is false, it means c is not assigned to any cluster and thus, the membership function is called for the c concept. In addition, the $\mathcal{F}$ flag can only have one value, i.e. each node can be placed in only one cluster so that no overlap is observed in clusters. The membership function determines that each concept $c_i \epsilon \mathcal{C}$should be placed in which $\mathcal{T}_i, i < \mathcal{K}$ cluster. For this determination, similarity of c_i with all $\mathcal{CH}_i$ is calculated and then $c_i$is placed in a cluster with maximum similarity. Eq. (14) shows this matter. Using the proposed membership function, each concept is compared with Cluster Heads, instead of comparing with all concepts like whatever was done in [4, 5]. The proposed method that uses seeding of cluster head to reduce the complexity of comparison.

$$c_i \epsilon \mathcal{CH}_k \Big| \mathcal{CH}_k = \max_{k \in \mathcal{K}} MemberShipFunc(c_i, \mathcal{CH}_k) \quad (14)$$

In order to measure the membership of a concept to a cluster head, a linear weighted combination of structural and string similarity measures is calculated as Eq. (15).

$$\text{MemberShipFuncCwithCH}(c_i, \mathcal{CH}_k) = \alpha \times \text{StructuralSimilarity}(c_i, \mathcal{CH}_k) + (1 - \alpha) \times \text{StringSimilarity}(c_i, \mathcal{CH}_k) \quad (15)$$

where $\alpha + \beta = 1$. We set experimentally $\alpha = 0.7$ and $\beta = 0.3$ String Similarity and Structural Similarity are explained in the next steps.

2) String Similarity Measure

As previously mentioned, $\mathcal{L} = \{\ell_1, \ldots, \ell_m\}$ is a finite set of labels that show the name of each concept. In this section, the name of the concept is used to calculate string

similarity between two nodes. Algergawy and et al. [30] showed that the name of nodes are the most dominant features. For this purpose, the Levenshtein distance [31] is used being also called string edit distance. Levenshtein distance is appropriate for variable length strings. This measure is very similar to the pair of string matching.

3) Structural similarity measure

In this measure, structural similarities such as similarity of paths, connections, and edges are used. Algergawy et al. [5] and Lin et al. [32] have used this measure before. In fact, these concepts with similar connections are semantically more similar to each other [١٩] and are placed in the same group. The structural similarity used in this study is shown in Eq. (16).

$$StructuralSimilarity(c_i, \mathcal{CH}_k) = \frac{1}{dist} + ShareNeighbour(c_i, \mathcal{CH}_k) \quad (16)$$

where, $ShareNeighbour(c_i, \mathcal{CH}_k)$ function calculates the number of share neighbors of c_i concept with $\mathcal{CH}_k$ cluster head. ShareNeighbour plays an important role in structural similarity, because similar concepts have similar neighbors [32]. The $dist$ is defined as.

$$dist = \frac{2 \times shortestDistance\ (c_i, \mathcal{CH}_k)}{shortestDistance(\mathcal{CH}_k, \mathcal{PS}_{ik}) + shortestDistance(c_i, \mathcal{PS}_{ik})} \quad (17)$$

where $\mathcal{PS}_{ik}$ is the nearest share parent between c_i and $\mathcal{CH}_k$ that is given below.

$$\mathcal{PS}_{ik}(c_i, \mathcal{CH}_k) = nearest\ \mathcal{P}arent\ \mathcal{S}hare(c_i, \mathcal{CH}_k) \quad (18)$$

4. Experimental Results

To evaluate SeeCC, an open source Falcon-AO system [2] was used. It was implemented in Java with Apache 2.0 license. Falcon-AO has some components including PBM (Partition Block Match). PBM [13] is used for large-scale ontology matching which was replaced by SeeCC.

All the experiments were carried out on Intel core i5 with 4 GB internal memory on Windows 7 with Java compiler 1.7. Evaluation of the SeeCC was done by standard tests using valid ontologies parsed with Jena Apache, and the mapping functions were implemented by Alignment API[3] . The standard information retrieval metrics as shown in Eqs. (19-21) were used to assess the results

$$\text{. Precision} = \frac{\text{number of correct found alignments}}{\text{number of found alignments}} \quad (19)$$

$$\text{Recall} = \frac{\text{number of correct found alignments}}{\text{number of exixting alignments}} \quad (20)$$

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

The OAEI dataset (http://oaei.ontologymatching.org), was tested in Conference and Anatomy sections and its results were compared with the results of Algergawy [5]. Moreover, the Falcon-AO system was implemented in the same setting and its results were compared to the SeeCC method. The Conference data set containing of 16 ontologies is much used in ontology matching systems. The Anatomy data set contains medicine ontologies including two data sets of human and mouse anatomy with 3306 and 2746 concepts, respectively.

The SeeCC method was run on the Conference data set and the results were compared with those of the Falcon-AO. Falcon-AO has 4 matchers so that for ontologies with more than 5000 concepts, PBM matcher is fetched. We changed this threshold value from 5000 to 100 since in the Conference test, the size of ontologies is less than 5000 and we wanted Falcon-AO to use PBM. The column 3 in Table (7) shows the results Falcon-AO by using the PBM method. As a whole by comparing with the SeeCC, we see 10.7% improvement in this test. Also, we tested Falcon-AO without these changes of the threshold shown in column 4 in the table 3. The results show that SeeCC is better about 7 % than other matchers of Falcon-AO even in small ontologies.

Error! Reference source not found.3 shows the results of comparison of SeeCC, Falcon-AO and the system proposed by Algergawy [5] on the Anatomy conference by two ontologies with 3306 and 2746 concepts. SeeCC method is 11 percent more accurate than the Algergawy method and 14.4 percent more accurate than Falcon-AO.

Figure 4 and figure 5 compare SeeCC with the top ten systems, participating in OAEI competition, in the Conference track and in the Anatomy track, respectively. For simplicity of the chart, only F-Measure of each system is shown in figure 4 and figure 5. The horizontal axis shows the participating systems and the vertical axis shows F-measure. We see that SeeCC method has comparable results with the others.

5. Conclusions

In this paper, we introduced a new clustering method, SeeCC, in order to solve the problem of large ontologies. SeeCC partitions large-scale ontology to several sub-ontologies and converts large-scale ontology matching to

[2] http://ws.nju.edu.cn/falcon-ao

[3] http://alignapi.gforge.inria.fr

several small ontology matching sub-problems. A Ranker function was presented to determine the cluster head. Selection of cluster head of high score nodes was done using the distribution condition.

TABLE 7. COMPARISON OF THE SEECC WITH FALCON-AO ON THE CONFERENCE DATA SET.

No	*1st ontology*	*2nd ontology*	*Falcon-AO*		*SeeCC*
			With PBM	***Without PBM***	
1	Cmt	conference	0.46	0.54	**0.6**
2	Cmt	confOf	0.28	0.44	**0.35**
3	Cmt	Edas	0.73	0.69	**0.73**
4	Cmt	Ekaw	0.56	0.54	**0.63**
5	Cmt	Iasted	0.8	0.66	**0.8**
6	Cmt	sigkdd	0.74	0.8	**0.8**
7	Conference	confOf	0.64	0.56	**0.67**
8	Conference	edas	0.59	0.52	**0.57**
9	Conference	ekaw	0.43	0.48	**0.52**
10	Conference	iasted	0.38	0.45	**0.45**
11	Conference	sigkdd	0.64	0.68	**0.69**
12	confOf	edas	0.51	0.48	**0.54**
13	confOf	ekaw	0.67	0.63	**0.65**
14	confOf	iasted	0.42	0.4	**0.4**
15	confOf	sigkdd	0.29	0.66	**0.67**
16	Edas	ekaw	0.58	0.6	**0.63**
17	Edas	iasted	0.5	0.46	**0.48**
18	Edas	sigkdd	0.63	0.6	**0.64**
19	Ekaw	iasted	0.57	0.6	**0.64**
20	ekaw	sigkdd	0.7	0.7	**0.7**
21	iasted	sigkdd	0.59	0.7	**0.78**
	Average		0.56	0.58	**0.62**

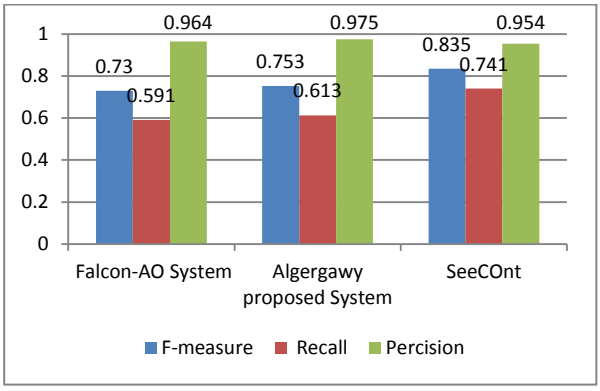

Fig. 3. Comparison of SeeCC with State-of-the-art in the Anatomy Method.

For each selected cluster head, one cluster was created and for the remaining nodes, a membership function was called which to reduce the number of comparisons. In fact, in order to compare all the concepts with each other, in the worst case in a graph with n nodes, $n \times (n-1)$ comparing operations per node was used with n^2 complexity, while in SeeCC a concept was only compared with cluster heads which the number of cluster head ($\mathcal{K}$) was much less than that of the concepts ($\mathcal{K} \ll Number\ of\ CH$) or ($\mathcal{K} \ll |\mathcal{O}|$) As a whole in SeeCC because of it use the seeding based algorithm, the number of comparison is $\mathcal{K} \times n$ which $\mathcal{K} \ll n$.)

Test results showed that the SeeCC method, compared to Falcon-AO (using PBM) and Falcon-AO (without using PBM) in the Conference test of OAEI, had respectively 10.7 and 7% improvement. Also in the Anatomy test of OAEI, the SeeCC method was 11 and 14.4 % better than the Algergawy's proposed system [5] and Falcon-AO, respectively. Moreover, the comparison of SeeCC method with the systems participating in OAEI in the Conference and Anatomy tests indicated that the SeeCC method had acceptable results.

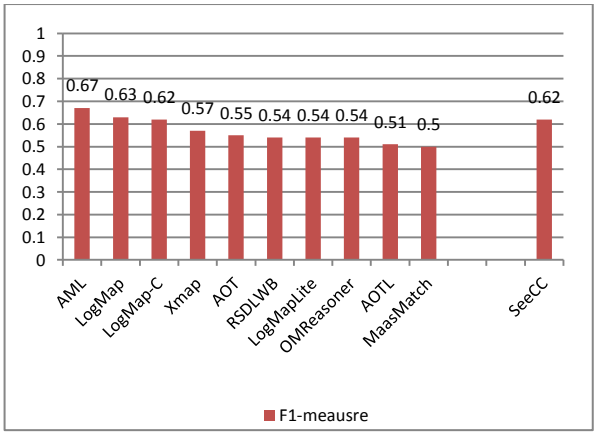

Fig. 4. Comparing SeeCC method with the Top Ten Systems Participating in OAEI Competitions in the Conference track.

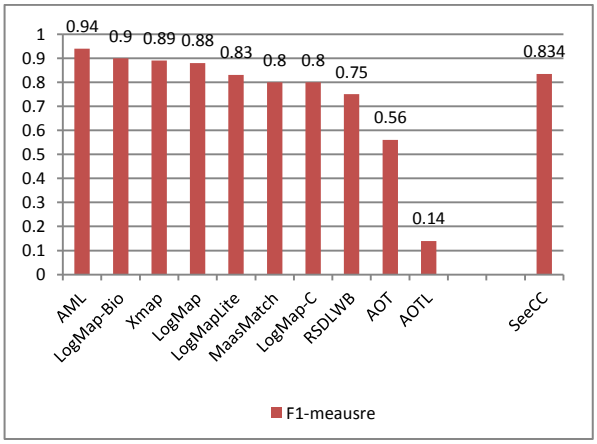

Fig. 5. Comparing SeeCC Method with the Top Ten Systems Participating in OAEI Competitions in the Anatomy track

REFERENCES

[1] Hendler, J. "Agents and the semantic web," Intelligent Systems, IEEE, vol. 16, no.2, pp. 30-37, 2001.

[2] Euzenat, J., C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn, "Ontology alignment evaluation initiative: six years of experience," in Journal on data semantics XV, Springer. pp. 158-192, 2011.

[3] Euzenat, J., A. Ferrara, W. van Hage, L. Hollink, C. Meilicke, A. Nikolov, et al. "Results of the

Ontology Alignment Evaluation Initiative 2011." in 6th OM workshop, 2011.

[4] Hu, W, Y. Qu, and G. Cheng, "Matching large ontologies: A divide-and-conquer approach," Data & Knowledge Engineering, vol. 67, no. 1, pp. 140-160, 2008.

[5] Algergawy, A., S. Massmann, and E. Rahm. "A clustering-based approach for large-scale ontology matching," in Advances in Databases and Information Systems, Springer, 2011.

[6] Wang, Z., Y. Wang, S. Zhang, G. Shen, and T. Du, "Matching large scale ontology effectively," in The Semantic Web–ASWC 2006, Springer, pp. 99-105, 2006.

[7] Khan, M., N. Javaid, M. Khan, A. Javaid, Z. Khan, and U. Qasim, "Hybrid DEEC: Towards Efficient Energy Utilization in Wireless Sensor Networks," arXiv preprint arXiv:1303.4679, 2013.

[8] Bsoul, M., A. Al-Khasawneh, A.E. Abdallah, E.E. Abdallah, and I. Obeidat, "An energy-efficient threshold-based clustering protocol for wireless sensor networks," Wireless personal communications, vol. 70, no. 1, pp. 99-112, 2013.

[9] Saruladha, K., G. Aghila, and B. Sathiya. "A partitioning algorithm for large scale ontologies," in Recent Trends In Information Technology (ICRTIT), 2012 International Conference on. IEEE, 2012.

[10] Zhou, Y, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," Proceedings of the VLDB Endowment, vol. 2, no. 1, pp. 718-729, 2009.

[11] Hu, W. and Y. Qu, "Falcon-AO: A practical ontology matching system. Web Semantics," Science, Services and Agents on the World Wide Web, vol. 6, no.3, pp. 237-239, 2008.

[12] Do, H.-H. and E. Rahm, "Matching large schemas Approaches and evaluation," Information Systems, vol. 32, no.6, pp. 857-885, 2007.

[13] Hu, W., Y. Zhao, and Y. Qu, "Partition-based block matching of large class hierarchies," in The Semantic Web–ASWC 2006, Springer, pp. 72-83, 2006.

[14] Jiménez-Ruiz, E. and B.C. Grau, "Logmap: Logic-based and scalable ontology matching," in The Semantic Web–ISWC 2011, Springer, pp. 273-288, 2011.

[15] Kirsten, T., A. Gross, M. Hartung, and E. Rahm, "GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution," J. Biomedical Semantics, vol. 2, pp. 6, 2011.

[16] Ngo, D. and Z. Bellahsene, "YAM++: a multi-strategy based approach for ontology matching task," in Knowledge Engineering and Knowledge Management, Springer, pp. 421-425, 2012.

[17] Grau, B.C., I. Horrocks, Y. Kazakov, and U. Sattler. "Just the right amount: extracting modules from ontologies," in Proceedings of the 16th international conference on World Wide Web, ACM, 2007.

[18] Wang, Z., Y. Wang, S. Zhang, G. Shen, and T. Du, "Ontology Pasing Graph-based Mapping: A Parsing Graph-based Algorithm for Ontology Mapping," Journal of Donghua University, vol. 23, no.6, pp. 5, 2006.

[19] Yuruk, N., M. Mete, X. Xu, and T.A. Schweiger. "AHSCAN: Agglomerative hierarchical structural clustering algorithm for networks. in Social Network Analysis and Mining," ASONAM'09. International Conference on Advances in. 2009 IEEE, 2009.

[20] Guha, S., R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," Information systems, vol. 25, no.5, pp. 345-366, 2000.

[21] Hamdi, F., B. Safar, C. Reynaud, and H. Zargayouna, "Alignment-based partitioning of large-scale ontologies," in Advances in knowledge discovery and management, Springer, pp. 251-269, 2010.

[22] Zhang, X., H. Li, and Y. Qu," Finding important vocabulary within ontology ", in The Semantic Web–ASWC 20062006, Springer. p. 106-112..

[23] Graves, A., S. Adali, and J. Hendler. "A Method to Rank Nodes in an RDF Graph," International Semantic Web Conference (Posters & Demos). 2008.

[24] Kermarrec, A.-M., E. Le Merrer, B. Sericola, and G. Trédan, "Second order centrality: Distributed assessment of nodes criticity in complex networks," Computer Communications, vol. 34, no. 5, pp. 619-628, 2011.

[25] Freeman, L.C," A set of measures of centrality based on betweenness ". Sociometry, 1977: p. 35-41.

[26] Hage, P. and F. Harary, "Eccentricity and centrality in networks," Social networks, vol. 17, no.1, pp. 57-63, 1995.

[27] Koschützki, D., K.A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, "Centrality indices," Network analysis, Springer, pp. 16-61, 2005.

[28] Zhang, X., G. Cheng, and Y. Qu, "Ontology summarization based on rdf sentence graph,"

Proceedings of the 16th international conference on World Wide Web, ACM, 2007.

[29] Stuckenschmidt, H, "Network analysis as a basis for partitioning class hierarchies," W8: Semantic Network Analysis, pp. 43, 2005.

[30] Algergawy, A., R. Nayak, and G. Saake, "Element similarity measures in XML schema matching," Information Sciences, vol. 180, no. 24, pp. 4975-4998, 2010.

[31] Levenshtein, V.I., "Binary codes capable of correcting deletions, insertions and reversals," in Soviet physics doklady, 1966.

[32] Lin, F. and K. Sandkuhl, "A survey of exploiting wordnet in ontology matching," in Artificial Intelligence in Theory and Practice II2008, Springer, pp. 341-350, 2008.