Consolidation of Linked Data Resources upon Heterogeneous Schemas

Aynaz Taheri
NLP Research Lab, Computer Engineering Dept
Shahid Beheshti University
Tehran, Iran
ay.taheri@mail.sbu.ac.ir

Mehrnoush Shamsfard
NLP Research Lab, Computer Engineering Dept
Shahid Beheshti University
Tehran, Iran
m-shams@sbu.ac.ir

Abstract— **Linked data resources have influential roles in conducting the future of semantic web. They are growing more and more, and the amount of published data is increasing at a fast pace. It causes some new concerns arise in the context of semantic web. One of the most important issues is the large amount of data that is produced as identical entities in heterogeneous data sources by different providers. This is a barrier to intelligent applications or agents that are going to utilize linked data resources. It prevents us from utilizing the potential capacity of web of data. Linked data resources are valuable when we could exploit them altogether. Therefore, we could obviously perceive the importance of linked data integration. In this paper, we propose a new approach for linked data consolidation. It helps us to have a consolidation process even between resources with heterogeneous schemas. In this approach, we are going to find more identical instances locally. This means that we direct our instance coreference resolution around the two instances which are certainly identical. The neighbors of two similar instances are a good source for our approach to proceed. In addition, these neighbors are beneficial for estimating some similarities between concepts of two heterogeneous schemas.**

Keywords-Linked Data; Consolidation; Ontology; Schema; Instance.

I. INTRODUCTION

Linked data has profound implications for the future of semantic web. Nowadays, the amount of published linked data is increasing and web of data is growing more and more. Linking Open Data (LOD) [24] project is the realization of web of data. Web of data includes billions of RDF [25] triples that are accumulated by different data providers. Accretion of data in Linking Open Data project is not the only challenge of publishing linked data; rather, matching and linking the linked data resources are also equally important and can improve the effective consuming of linked data resources. Linked data integration is one of the main challenges that become more important considering development of linked data. Without these links, we confront with isolated islands of datasets, which could not exploit knowledge of each other. The fourth rule of publishing linked data in [1] explains the necessity of linking URIs to each other. When there are possibilities of applying integrated linked data sources, information retrieval and utilizing linked data on the web would be thriving. Thus, we need identification and disambiguation of entities in different data sources. Unique entity identification in variant resources causes reduction of problems about data processing in heterogeneous data resources.

We created a new approach for entity coreference resolution in linked data resources. The proposed approach receives two ontologies, two sets of instances as linked data sources and two similar concepts from two ontologies. Instance matching algorithm initiates its process among the instances of two similar concepts that are received from the inputs. In fact, the instance matcher is now assured of equality of these two concepts and knows that it can find identical instances among the instances of the two concepts. Our approach searches for finding identical instances by applying a new method that is explained in section 2. We use the properties of instances and their values to discover similar instances. In addition, neighbors of instances are the other significant features that we apply for identifying instances. Neighbors have prominent roles in the performance of our method. After finding identical instances, we continue the process locally around the identical instances. Identical instances are good points for our algorithm to proceed since searching around two identical instances raises the possibility of finding equal instances. Another great merit of finding similar instances in the neighborhood of identical instances is to help us contend with heterogeneous schemas. Section 3 explains about this issue.

This paper is structured as follows: Section 2 outlines the instance matching algorithm. Section 3 explains how instance matching of our approach helps us in overcoming difficulties of schemas heterogeneity. Section 4 discusses our experiments over one dataset. Section 5 points to some related works and finally, Section 6 concludes this paper.

II. INSTANCE MATCHING

The process of instance coreference resolution needs to receive a pair of concepts from two ontologies. These two concepts are equal and we are going to find identical individuals among their instances.

A. Create a Net around the Instances

We introduce a new construction that is called *Net*, as the basis of our instance matching algorithm.

For two equivalent concepts that we receive as input, we must create Nets. For each instance of two concepts, we make one Net. If we have an instance that its URI is ‘i’, we explain how to create a Net for instance ‘i’. For creating this Net, all of the triples whose subjects are instance ‘i’ are

 ISBN: 978-1-61208-240-0

extracted and added to the Net. Then, in the triples that belong to the Net, we find neighbors of instance 'i'. If instance 'j' is one of the neighbors of instance 'i', the same process is repeated for instance 'j'. Triples, whose subjects are instance 'j', are added to the Net, and the same process is repeated for neighbors of the instance 'j'. This process is actually like depth first search among neighbors of instance 'i'. To avoid falling in a loop and eliminating the size of search space, the maximum depth of search is experimentally set to 5. This depth gives us the best information about an instance and its neighbors. The Net that is created for instance 'i' is called Net_i. Starting point for this Net is instance 'i'.

The process of creating Nets is done for all of the instances of the two concepts. Creating Nets helps us in recognizing instance identities. Identities of instances are sometimes not recognizable without considering the instances that are linked to them, and neighbors often present important information about intended instances. In some cases in our experiments we observed that even discriminative property-value pairs about an instance may be displayed by its neighbors. Figure 1 shows an illustration about an instance that its neighbors describe its identity. This example is taken from IIMB dataset in OAEI 2010. Figure 1 shows $\mathrm{Net}_{\mathrm{item2117}}$. 'Item2117' is the starting point of this Net and is an Actor, Director and a character-creator. Each instance in the neighborhood of 'Item2117' describes some information about it. For example, 'Item7448' explains the city that 'Item2117' was born in and 'Item2705' explains the name of the 'Item2117'.

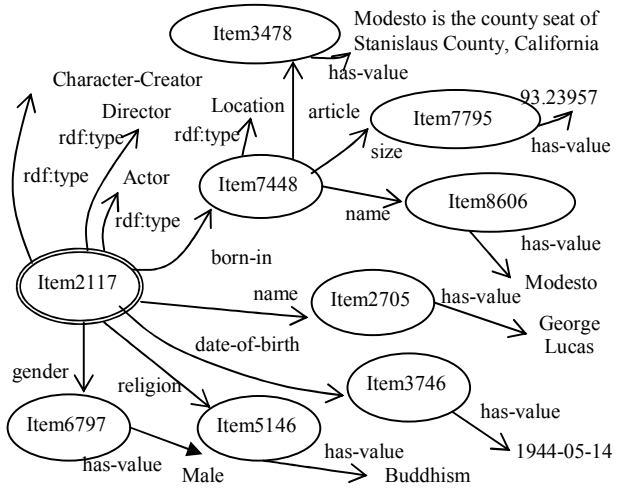

Figure 1. An Illustration of Net

Not only does creating Nets help us in discovering identities of instances, but also it helps us to find locally more similar instances. This issue is explained in the second part of Section B.

B. Compute the Nets Similarities

In the previous step, Nets of two equal concepts were created. In this step, we must compare them.

1) Finding identical instances

Each Net from one concept is supposed to be compared with all Nets of the other concept in order to find similar Nets. Starting points of two similar Nets would be equal. Each Net is composed of some triples that are extracted from the dataset. Therefore, triples of two Nets should be compared. In this process, only triples whose objects are data type values (and not instances) would participate in the comparison. Properties values are very important in comparison.

We use edit distance method for comparing string values of properties. Some properties explain comments about instances. In these situations, we used a token-based measure for computing similarity.

Similarity values of triples objects are added together for obtaining similarity value of two Nets. We applied a threshold for Edit Distance method. This threshold was found by making a benchmark and execution of edit distance algorithm based on the benchmark. We round the threshold to one decimal point and the value of threshold is 0.6.

After calculating similarity of properties values, we computes similarity of two Nets. Similarity of two Nets is dependent on similarity of their properties values. Triples in two Nets have specific importance depend on the depth of their subjects (instances that triples belong to) in the Nets. Depth of instances is estimated toward the starting point of the Net. When depth of an instance in a Net increases, its effectiveness on similarity computation of Nets decreases. The following triples belong to $\mathrm{Net}_{\mathrm{item2117}}$ in Figure 1.

1 ('Item6797', has-value, Male)
2 ('Item3746', has-value, 1944-05-14)
3 ('Item7795', has-value, 93.23957)
4 ('Item3478', has-value, Modesto is the county seat of Stanislaus County, California)

The above triples describe some information about the starting point of $\mathrm{Net}_{\mathrm{item2117}}$. Two first triples explain that 'item2117' has male gender and date of his birth is 1994-05-14. Instances in the subjects of these two triples have depth equal to two. Two second triples explain that 'item2117' has born in a city that its size is 93.23957 and also is the county seat of Stanislaus County, California. Instances in the subjects of these two triples have depth equal to three. As you can see, the first two triples have more important role for determining the identity of 'item2117' than the second two triples. Gender of a person and date of his birth is more important than some comments about the city that he lives in. Nevertheless, this does not mean that existence of instances with greater depth are not beneficial in the Nets; rather, they are less important in identity recognition of the starting point of the Net than those with less depth.

In this regard, similarities of properties values are added with an particular coefficient. We use a weighted sum for computing similarity of Nets. The coefficients in this sum have inverse relations to the depth of the subject of triples in Net.

We normalize the sum of similarities of properties values in two Nets into a range of 0 and 1 by dividing the result to the sum of the numbers of triples in two Nets. After finding

 ISBN: 978-1-61208-240-0

the similarities between all the Nets of two concepts, we sort the similarity values in a list based on the descending order, and most similar Nets are selected respectively. An one to one relation is made between similar Nets. Nets with similarity values less than 0.5 are omitted. This threshold is obtained experimentally. We made a benchmark of our Nets and selected the best threshold which could represent us the similarity threshold.

When two Nets are selected as two similar Nets, we consider their starting points as identical instances. In this way, some identical instances could be found regarding to their properties and their neighbors.

2) Finding identical instances in the vicinity of identical instances

We found some identical instances with utilizing their Nets. In this step, we continue the process of matching on those Nets of the previous step that led to discovering equal instances or in the other words, those Nets that have equal starting points. The strategy in this step is searching locally around the identical instances in order to find new equal instances. Seddiqui, et al. [20] created an algorithm for ontology matching and their algorithm is based on the idea that if two concepts of two ontologies are similar, then there is a high possibility that their neighbors are similar too. We use this idea but in instance level. This means that if two instances are identical, then there is possibility that their neighbors are similar too.

Suppose that 'i' and 'j' are two instances that are detected identical in the previous step. Their Nets are called $\mathrm{Net_i}$ and $\mathrm{Net_j}$. In this step we describe how the approach finds more identical instances in $\mathrm{Net_i}$ and $\mathrm{Net_j}$. For discovering similar instances in $\mathrm{Net_i}$ and $\mathrm{Net_j}$, we compare instances in these two Nets. The process of comparing instances is similar to what mentioned in the first part of section B. Instances would be compared regarding their properties and values.

Finding identical instances of two concepts initially costs a lot in first part of section B because of considering all neighbors of an instance; later we can find locally more identical instances by paying low computational cost.

III. COMPUTE CONCEPT SIMILARITIES IN SCHEMA LEVEL

After finding identical instances in the neighborhood of identical instances, now it is time to find similarities between concepts in two heterogeneous schemas. In this part, instance matcher gives feedback to us for finding similar concepts in schema level. If we find some similar instances such as 'm' and 'n' in the instances of $\mathrm{Net_i}$ and $\mathrm{Net_j}$, concepts that 'm' and 'n' belong to them would be good candidates to be similar.

The approach repeats this step for every two similar Nets and considering to identical instances in two similar Nets, estimates similarities between concepts. We used a measure in order to find a similarity value between two concepts. C_1 and C_2 are two concepts that we made Nets for their instances and then compared their Nets. C_3 and C_4 are two concepts that we have concluded their similarity from the neighbor instances of C_1 and C_2 instances. Then, we define the similarity value of C_3 and C_4 based on the ratio of neighbor instances of C_1 and C_2 instances that concluded similarity between C_3 and C_4, to the number of Nets in C_1 and C_2.

The approach gives us some similarity values between concepts of two ontologies. In the implemented approach, we did not apply any other methods for ontology matching. We used these similarity values and managed the matching process manually. In fact, similarity values conducted our matching process significantly. These equal concepts are inputs for the next execution of instance matcher.

IV. EXPERIMENTTS

We used a dataset of OAEI [5], a benchmarking initiative in the area of semantic web. We report the experimental results of our proposed approach on IIMB dataset in OAEI 2011. IIMB composed of 80 test cases. Each test case has OWL ontology and a set of instances. Information of test cases in IIMB track is extracted from Freebase dataset. IIMB divided test cases in four groups. Test cases from 1 to 20 have data value transformations, 21 to 40 have structural transformations, test cases from 40 to 60 have data semantic transformations and 61 to 80 have combination of these three transformations. All of these 80 test cases are supposed to be matched against a source test case. We choose IIMB 2011 test cases for the evaluation because this track of OAEI has all kinds of transformations and we could compare all aspects of our system against the other system. Moreover, the size of IIMB 2011 has increased greatly compared to last years and is more than 1.5 GB. Increased amount of the dataset size lets us evaluate scalability of our approach. Unfortunately, there has been just one participant in this track, CODI [10], with which we will compare our results. This shows the scalability difficulties in systems performance at large scale datasets.

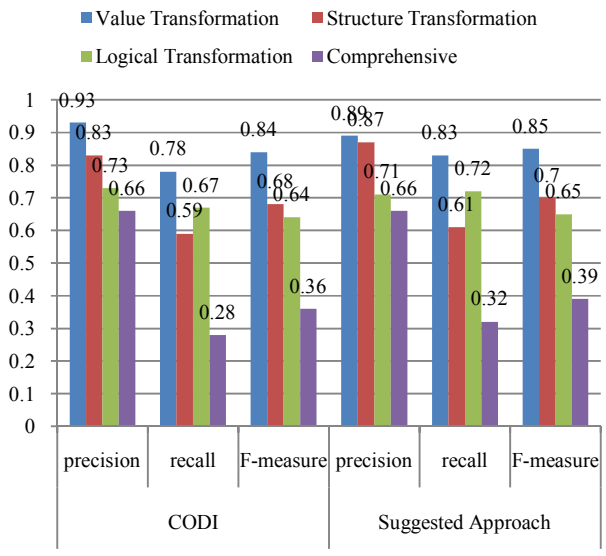

Figure 2. Results of OAEI'11 IIMB Track

We observe in Figure 2 that the recall values of our approach in four kinds of transformations are better than

CODI but this is not always true for precision value. The operations of our approach is clearly better than CODI in datasets with structure transformation considering three aspects of precision, recall and F-measure. This means that our approach is more stable in modifications such as removing, adding and hierarchal changing of properties.

V. RELATED WORK

The problem of entity coreference resolution is not a new challenge. There are a large number of related works on this issue in the context of database and the problem is called record linkage. We state some of these works in the area of entity coreference resolution in the context of semantic web. Raimond, et al. [16] proposed a method for interlinking two linked data music-related datasets that have similar ontologies. Hassanzadeh and Consense [6] described how they interlinked a linked data source about movies with other data sources in LOD by applying some exact and approximate string similarity measures. In [22], a method for linking WordNet VUA (WordNet 3.0 in RDF) to DBpedia is proposed. Finding identical instances of foaf:person at social graph is explained in [17] by computing graph similarity. Hogan, et al. [7] proposed an approach that capturing similarity between instances is based on applying inverse functional properties in OWL language. Noessner, et al. [15] used a similarity measure for computing similarity of instance matching between two datasets with the same ontology. LN2R [18] is a knowledge based reference reconciliation system and combines a logical and a numerical method. Hogan and colleagues [8] proposed a method for consolidation of instances in RDF data sources that is based on some statistical analysis. ObjectCoref [9] is a self-training coreference resolution system based on a semi supervised learning algorithm. Song and Heflin [21] described an unsupervised learning algorithm in order to find some discriminable properties as candidate selection key. Zhishi.links [14] is a distributed instance matching system. It does not follow any special techniques for schema heterogeneity. It uses an indexing process on the names of instances. HMatch(τ) [3] is an instance matcher and use HMatch 2.0 for TBox matching and then tries to capture the power of properties at instance identification. RiMOM [23], ASMOV [12] and AgreementMaker[4] are three ontology matching systems that recently equipped with instance matchers. CODI [10] is also a system for ontology and instance matching and is based on markov logic. Nikolov and colleagues proposed Knofuss architecture [13] that contains both schema and instance level. Linked Data Integration Framework (LDIF) [19] has two main components Silk Link Discovery Framework [11] and R2R Framework [2] for identity resolution and vocabulary normalization respectively.

What distinguish our approach from the aforementioned approaches is that our approach considers that the neighbors of an instance are important in order to find similarity between identical instances. We proposed a new approach for finding identical instances.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new approach for linked data consolidation. Instance resolution process starts after getting two equal concepts as input by instances matcher. Instance matcher creates Nets around the instances of two equal concepts and then compares these Nets. Our approach selects the Nets with most similarity value and considers that as similar Nets. Similar Nets have identical instances in their starting points. Instance matcher searches instances in the similar Nets in order to find identical instances around their equal starting points. After discovering instances with the same identity in Nets, instance matcher utilizes them and computes some similarity values between concepts in the schema level. It sends us most similar concepts as a feedback for starting the instance matching again.

Our future target includes utilizing some methods for schema matching in our approach. We could devise a schema matcher for our approach so that schema and instance matchers could perform consecutively. Furthermore, we must apply a better method for finding the threshold that is the final approver of two similar Nets. It is better to find a heuristic measure in order to find a dynamic threshold.

REFERENCES

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data-The Story So Far," Int. J. Semantic Web Inf. Syst. vol. 5, no. 3 ,2009, pp. 1-22.

[2] C. Bizer and A. Schultz, "The R2R Framework: publishing and discovering mapping on the web," Proc. 1st International Workshop on Consuming Linked Data ,Shanghai, China, Nov. 2010.

[3] S. Castano, A. Ferrara, S. Montanelli, and D. Lorusso, "Instance matching for ontology population," Proc.16th symposium on advanced database systems, Italy, Jun. 2008, pp. 121-132.

[4] I. F. Cruz, C. Store, F. Caimi, A. Fabiani, C. Pesquita, F. M.Couto, and M. Palmonari, "Using AgreementMaker to align ontologies for OAEI 2011," Proc. 6th International Workshop on Ontology Matching. Bonn, Germany, 24 Oct. 2011.

[5] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. Trojahn, "Ontology Alignment Evaluation Initiative: Six Years of Experience," J. Data Semantics, vol. XV, 2011, pp. 158–192.

[6] O. Hassanzadeh and M. Consense, "Linked movie data base," Proc. 2nd Link Data on the Web, Madrid, Spain, Apr. 2009.

[7] A. Hogan, A. Harth, and S. Decker, "Performing object consolidation on the semantic web data graph," Proc. 1st Identity, Identifiers, Identification Workshop, Banff, Canada, May 2007.

[8] A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann, "Some entities are more equal than others: statistical methods to consolidate linked data," Proc. 4th International Workshop on New Forms of Reasoning for the Semantic Web , Crete, Greece, May 2010.

[9] W. Hu, J. Chen, and Y. Qu, "A Self-training Approach for Resolving Object Coreference Semantic Web," Proc. 20th International World Wide Web Conference, India, Mar. 2011, pp. 87-96.

[10] J. Huber, T. Sztyler, J. Noessner, and C. Meilicke, "CODI : Combinatorial Optimization for Data Integration – Results for OAEI 2011," Proc. 6th International Workshop on Ontology Matching. Bonn, Germany, Oct. 2011.

[11] R. Isele, A. Jentzsch, and C. Bizer. "Silk server- adding missing links while consuming linked data," Proc. 1st International Workshop on Consuming Linked Data,Shanghai, China, Nov. 2010.

[12] Jean-Mary, Y.R., Shironoshita, E.P., and Kabuka, M.R. "ASMOV: Results for OAEI 2010," Proc. 5th International Workshop on Ontology Matching. Shanghai, China, Nov. 2010.

[13] A. Nikolov, V. Uren, E. Motta, and A. Roeck, "Overcoming schema heterogeity between linked semantic repositories to improve coreference resolution," Proc. 4th Asian Semantic Web Conference, Shanghai, China, Dec. 2009, pp. 332-346.

 ISBN: 978-1-61208-240-0

[14] X. Niu, S. Rong, Y. Zhang, and H., Wang, "Zhishi.links results for OAEI 2011," Proc. 6th International Workshop on Ontology Matching, Bonn, Germany, Oct. 2011.

[15] J. Noessner, M. Niepert, C. MeiNetke, and H. Stuckenschmidt, "Leveraging Terminological Structure for Object Reconciliation," Proc.7th Extended Semantic Web Conference, Heraklion, Greece, May 2010, LNCS 6089, pp. 334-348.

[16] Y. Raimond, C. Sutton, and M.Sandler, "Automatic Interlinking of music datasets on the semantic web," Proc. 1st Link Data on the Web, Beijing, China, Apr. 2008.

[17] M. Rowe, "Interlinking Distributed Social Graphs," Proc. 2nd Linked Data on the Web Workshop, Madrid, Spain, Apr. 2009.

[18] F. Sais, N. Niraula, N. Pernelle, and M. Rousset, "LN2R a knowledge based reference reconciliation system: OAEI 2010 results," Proc. 5th International Workshop on Ontology Matching Shanghai, China, Nov. 2010.

[19] A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker, "LDIF-Linked data integration framework," Proc. 2nd International Workshop on Consuming Linked Data ,Bonn, Germany, Oct. 2011.

[20] Md. Hanif Seddiqui, and M. Aono, "An Efficient and Scalable Algorithm for Segmented Alignment of Ontologies of Arbitrary Size," J. Web Sem. Vol. 7, Jan. 2009, pp. 344-356.

[21] D. Song, and J. Heflin, "Automatically generating data linkage using a domain-independent candidate selection approach," Proc. of the 10th International Semantic Web Conference ,Koblenz, Germany, Oct. 2011, LNCS 7031, pp. 649-664.

[22] A. Taheri, and M. Shamsfard, "Linking WordNet to DBpedia," Proc. 6th Global WordNet Conference, Matsue, Japan, Jan. 2012, ISBN: 978-80-263-0244-5, pp. 344-348.

[23] Z. Wang, X. Zhang, L. Hou, Y. Zhao, J. Li, Y. Qi, and J. Tang, "RiMOM Results for OAEI 2010," Proc. 5th International Workshop on Ontology Matching. Shanghai, China, 2010.

[24] www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/Linking OpenData [retrieved: July, 2012]

[25] "RDF Vocabulary Description Language 1.0: RDF Schema", http://www.w3.org/TR/rdf-schema/ [retrieved: July, 2012]

 ISBN: 978-1-61208-240-0