SBUEI: Results for OAEI 2012

Aynaz Taheri, Mehrnoush Shamsfard

Computer Engineering Department, Shahid Beheshti University, Tehran, Iran
ay.taheri@mail.sbu.ac.ir,m-shams@sbu.ac.ir

Abstract. In this paper, we describe our system, SBUEI, for instances coreference resolution between various sources even with heterogeneous schemas. It is the first participation of SBUEI in instance matching track of Ontology Alignment Evaluation Initiative campaign. We present the results of SBUEI in the 2012 OAEI competition in two tracks: Sandbox and IIMB. SBUEI considers the instance coreference resolution in both schema and instance levels. The process of matching is applied to both levels consecutively to let the system discover identical instances.

1 Presentation of the system

Linked data resources have influential roles in conducting the future of semantic web. In recent years, different data providers have produced many data sources in Linking Open Data (LOD) cloud upon different schemas. Increases in the amount of linked data in LOD is not the only challenge of publishing linked data; rather, matching and linking the linked data resources are also equally important. The fourth rule of publishing linked data in [1] explains the necessity of linking URIs to each other. In the web of linked data, there are obviously many different kinds of schemas in various linked data resources. Therefore, we confront with schema heterogeneity in order to do coreference resolution. The importance of this issue motivated us to create a new system, SBUEI, for entity coreference resolution.

SBUEI deals with the both problems of instance matching and schema matching. SBUEI proposes an interleaving of instance and schema matching steps to find coreferences or unique identities in two sources. This approach is applicable to find unique identities in two linked data sources. SBUEI, unlike systems such as [2, 3, 4] -which uses just instance matching- or systems such as [5, 6] -which use just schema matching- exploits both levels of instance and schema matching. The main difference between SBUEI and other systems like [7], which exploit both levels, is that SBUEI exploits an interleaving of them while [7] exploits them sequentially one after the other (starts instance matching after completing schema matching). SBUEI utilizes schema matching results in instance matching and use the instance matching results in order to direct matching in schema level. SBUEI also has a new approach for instance matching.

1.1 State, purpose, general statement

SBUEI begins matching process by receiving two similar concepts of two ontologies called anchors. In fact, the inputs of SBUEI are two ontologies, two data sets of instances and the anchors (two equivalent concepts from the ontologies [8]).

Fig. 1 shows an example of performing SBUEI. In this figure two ontologies, *O1* and *O2* are represented. Each of them has a set of instances (*I1* and *I2*) . *a1* and *b1* are the anchors which are the two equivalent concepts of two ontologies. SBUEI begins the work with confidence to equality of *a1* and *b1* and starts searching instances of two concepts *a1* and *b1* to find instances with unique identity. This task is done by a new coreference resolution algorithm, described in [9].

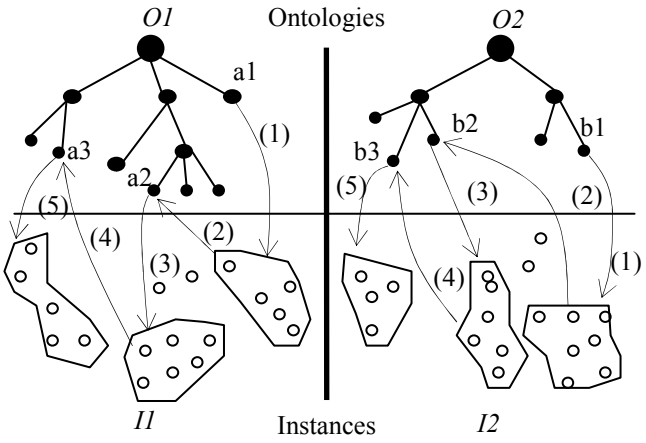

Fig1.Interleaving schema and instance matching process

This is the first transition between schema level and instance level. It is the first step in discovering instances with unique identity and indicated by arrow (1) in the figure. After discovering instances with unique identity, SBUEI utilizes these identical discovered instances and analyzes them in order to estimate similarities between concepts of schema. Similar concepts are those which have similar instances. As Fig. 1 shows, after doing resolution process between instances of *a1* and *b1* and analyzing the results, SBUEI estimates similarities between *a2* and *b2*. This is the first transition from instance level to schema level, which is represented by arrow (2). Schema matcher receives feedback from instance matcher and recognizes two equal concepts from *O1* and *O2* ontologies. After recognition of two equal concepts, SBUEI returns to instance level again (arrow (3). These processes continue consecutively until there are no instances or concepts for matching or there is not possibility for SBUEI to find more alignments. Therefore, SBUEI has two main components that are illustrated in Fig. 2: (instance matcher and schema matcher).

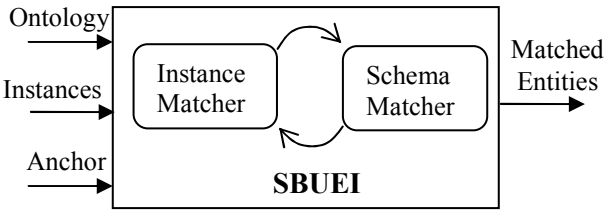

Fig2. Main Components of SBUEI

1.2 Specific techniques used

As described before, the instance coreference resolution algorithm has two phases which are executed iteratively. The first phase needs to receive an anchor as input. As the first and second phases are executed in a cycle, for the first round, the user should provide this input, but in the next times the input of the first phase (the anchors) is provided by the output of the second phase.

1.2.1 Instance coreference resolution

The instance matching process of SBUEI is completely explained in [9]. In this section, we explain the instance matching process of SBUEI concisely.

First step: create Linked Instances Cloud

We introduce a new construction that is called Linked Instances Cloud (LIC), as the basis of our instance matching algorithm.

For two equivalent concepts that we receive as input, we must create LICs. For each instance of two similar concepts, we make one LIC. If SBUEI wants to make a LIC for a specific instance, it extracts all the triples that their subjects are our intended instance and adds them to the LIC. In this way, all the neighbors of our intended instance are found. Then, SBUEI finds the triples that their subjects are instances which belong to the LIC. This means that the neighbors of the neighbors of our intended instance are found and added to the LIC. This process is actually like depth first search among neighbors of instances. SBUEI traverse across the neighbors of the instance and has a maximum depth for traversing.

The process of creating LICs is done for all of the instances of the two concepts. Creating LICs 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.

Second step: compute similarity between LICs and finding identical instances

In this step, the LICs of two equal concepts should be compared. Each LIC from one concept is supposed to be compared with all LICs of the other concept in order to find similar LICs. Starting points of two similar LICs, would be identical instances. For comparing two LICs, triples of two LICs 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 and a token-based measure for comparing string values of properties. Similarity values of triples objects are added together for obtaining similarity value of two LICs. Similarities of properties values are added with a particular coefficient which has inverse relations to the depth of the subject of triples in LIC. We use a weighted sum for computing similarity of LICs. We normalize the sum of similarities of properties values in two LICs into a range of 0 and 1 and select the most similar LICs.

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

Third step: finding identical instances in the vicinity of identical instances

We found some identical instances with utilizing their LICs. In this step, we continue the process of matching on those LICs of the previous step that led to discovering identical instances. The strategy in this step is searching locally around the identical instances in order to find new equal instances. This means that if two instances are identical, then there is possibility that their neighbors are similar too. The process of comparing instances is similar to what mentioned in the previous steps.

1.2.2 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 LIC_i and LIC_j (i and j are two identical instances that are detected in the second step), concepts that 'm' and 'n' belong to them would be good candidates to be similar.

The approach repeats this step for every two similar LICs and considering to identical instances in two similar LICs, estimates similarities between concepts. SBUEI used a measure in order to find a similarity value between two concepts.

The second phase is done by a schema matcher. It receives feedback from the first phase, which contains some similarities between concepts from the viewpoint of instance matcher. At this time, schema matcher begins the process of matching in schema level by applying some ontology matching algorithms. SBUEI compares all of these similarity values that are proposed by instance matcher or obtained by schema matcher, and choose a pair of concepts that have the most similarity. SBUEI repeats these two phases consecutively.

When SBUEI wants to do ontology matching, it considers to the concepts that are proposed as equal concepts in the previous iterations and the process of ontology matching starts in the neighborhood of these concepts. We applied the definition of concept neighborhood in [8]. Schema matcher utilizes two kinds of matchers: lexical matcher and structural matcher. Lexical matcher uses Princeton WordNet [10], EditDistance method and Wu-Palmer measure [11] for computing lexical similarities. In [12] structure based techniques are divided into two groups based on the internal structure and relational structure. SBUEI utilizes internal and relational structures for computing similarities between concepts.

1.4 Link to the system and parameters file

The website of SBUEI is http://nlp.sbu.ac.ir/sbuei/sbuei.html More information about SBUEI is presented here.

1.5 Link to the set of provided alignments (in align format)

The alignments of SBUEI for OAEI campaign is available at: http://nlp.sbu.ac.ir/sbuei/download.html

2 Results

In this section, we present the results obtained by SBUEI in the OAEI 2012 campaign. SBUEI participated in two tracks: Sandbox and IIMB. The results are evaluated in comparison with some gold standard alignments.

2.1 Sandbox Track

Sandbox is a simple data set and contains 11 test cases. Test cases contain some kinds of transformations such as data value transformation, structural transformation and logical transformation. The transformations are not as hard as the transformations of IIMB track. The data set is generated artificially. Table 1 represents the total amounts of precision, recall and F Measure for this data set.

Table 1. Sandbox Results

Test Cases	1-11
Precision	0.95
Recall	0.98
F Measure	0.96

We have encountered some reductions in precision and recall value. So, we analyzed the result and found some problems in the data set and reference alignments:

- There are some URI aliases in each test case. For example, see the URI1 and URI2 in test case 000 (test case 000 is the test case that other test cases must be matched against this test case):
  URI1: http://oaei.ontologymatching.org/2012/IIMBDATA/m/0bvgl_4
  URI2: http://oaei.ontologymatching.org/2012/IIMBDATA/m/0bvgm5l
  These two URIs depict the same identity. Both of them have exactly the same properties and values.
  On the other hand, we have some URI aliases in test case 001, such as URI3 and URI4.

URI3: http://oaei.ontologymatching.org/2012/IIMBDATA/m/item1009947992294508239
URI4: http://oaei.ontologymatching.org/2012/IIMBDATA/m/item5956760174121985261
URI3 and URI4 describe identical instances. Moreover, URI1, URI2, URI3 and URI4 refer to an entity and present the same identity. SBUEI found these alignments: (URI1, URI3), (URI1,URI4), (URI2,URI3), (URI2,URI4). However, only two alignments (URI1,URI4) and (URI2,URI3) belong to gold standard alignments. Therefore, our precision has decreased.

- We found an incorrect alignment in the reference alignments. See the following URI (URI5 from test case 000) which describe the English language.
  URI5: http://oaei.ontologymatching.org/2012/IIMBDATA/en/english
  In test case 001, there is an instance with the following URI (URI6) which its identity is the same as the URI5 and represents the English language.
  URI6: http://oaei.ontologymatching.org/2012/IIMBDATA/en/item7208291329366150827
  We can find the alignment (URI5, URI6) in gold standard alignments. Nevertheless, we can also find another incorrect alignment for URI5. URI5 is matched incorrectly with an instance with URI7.
  URI7: http://oaei.ontologymatching.org/2012/IIMBDATA/en/item6773019142593325946
  So, we have these alignments in gold standard alignments: (URI5, URI6), (URI5, URI7). But, SBUEI found only (URI5, URI6) as two identical instances. Hence, its recall has declined.

2.2 IIMB Track

IIMB data set is extracted from Freebase and includes 80 test cases. Each test case contains some kinds of different transformations. Test cases 1 to 20 contain data value transformation, 21-40 contain structural transformation, 41-60 contain logical transformation and 61-80 contain a combination of these transformations. All of these 80 test cases must be matched against a source test case. Table 2 shows the results of SBUEI on different groups of test cases (based on their transformations).

Table 2. IIMB Results

Transformations	1-20	21-40	41-60	61-80	overall
Precision	0.95	0.96	0.91	0.58	0.87
Recall	0.98	0.98	0.85	0.5	0.85
F Measure	0.97	0.98	0.87	0.53	0.86

We observed some problems in the IIMB task such as those problems that we mentioned in Sandbox task. These problems such as URI aliases have decreased our precision.

3 General comments

In this section, we provide some comments about our system and OAEI 2012 campaign.

3.1 Comments on the results

The results of our system are very promising. SBUEI obtained high value for precision, recall and F-measure in Sandbox task and test cases 1-40 of IIMB task. SBUEI has much better performance in test cases with data value transformation and structural transformation than test cases with logical transformation and combinational transformations. This means that SBUEI is very resistant to modifications such as changes in data format, removing, adding and hierarchal changing of properties. As we expected, SBUEI has its weakest performance in front of combinational transformations, and it is completely normal for systems to have weaker performance against combinational transformations than other transformation because it contains all kinds of transformation together. However, it is one of the most important shortcomings of our system and it is very beneficial to improve it by applying new techniques.

3.2 Discussions on the way to improve the proposed system

Our system participated for the first time in this competition and we focused a lot more on technical issues and our new algorithm than some usability aspects. Considering that SBUEI is a recently created approach, does not have appropriate user interface. Therefore, it is important to make a powerful user interface for SBUEI. Our future target includes utilizing some methods such as semi supervised learning algorithms to find discriminable properties in the LICs. This will help us to find similar LICs efficiently and optimize our system in order to improve some scalability aspects of our system.

3.3 Comments on the OAEI 2012 procedure

In OAEI 2012, SEALS platform is used for evaluating participating systems in all the tracks except for instance matching. It would be very beneficial for instance matching track to be run on a platform like SEALS.

3.4 Comments on the OAEI 2012 test cases

In the IIMB track of OAEI 2011, we had test cases which the size of their data sets had been heavily increased compared to the preceding years. Each test case size was more than 20MB. Therefore, participants had to deal with large data sets and their systems were evaluated considering some scalability aspects. In OAEI 2012, the sizes of data sets are not as much as the last year and they are declined. The large data sets

are more challenging for systems. Thus, it will be useful to have a better and stronger evaluation by large data sets. Moreover, we encountered some problems in reference alignments that we discussed about them in section 2. It is better to have more accurate data sets and reference alignments.

4 Conclusion

In this paper, we have described our system, SBUEI, for instance matching. SBUEI is applicable in various data sets with heterogeneous schemas. SBUEI pays attention to matching in both schema and instance level. The architecture, the main algorithms and the specific techniques of SBUEI have been presented in this report. Our experiments in Sandbox and IIMB showed that our approach achieved high precision and recall. This was the first participation of SBUEI and we obtained promising results; however, there are more technical issues that can improve the performance of SBUEI. We are going to optimize our system based on what was mentioned earlier in the future work.

References

1. Bizer, C., Heath, T. and Berners-Lee, T. Linked Data-The Story So Far, Int. J. Semantic Web Inf. Syst. 5(3), 1-22 (2009)
2. Hu, W., Chen, J. and Qu, Y. A Self-training Approach for Resolving Object Coreference Semantic Web, In: 20th International World Wide Web Conference, India (2011)
3. Noessner, J., Niepert, M., Meilicke, C., Stuckenschmidt. Leveraging Terminological Structure for Object Reconciliation, In:7th Extended Semantic Web Conference, Greece (2010)
4. Sais, F., Niraula, N., Pernelle N. and Rousset, M. LN2R a knowledge based reference reconciliation system: OAEI 2010 results, In: 5th International Workshop on Ontology Matching , China (2010)
5. Jain, P., Hitzler, P., Sheth, A. P., Verma, K. and Yeh, P. Z. Ontology Alignment for Linked Open Data, In: 9th International Semantic Web Conference, China (2010)
6. Parundekar, R. Knoblock, C. A. and Ambite, L. Linking and building of ontologies of linked data. In: 9th International Semantic Web Conference, China (2010)
7. Nikolov, A., Uren, V., Motta, E. and Roeck, A. Overcoming schema heterogeity between linked semantic repositories to improve coreference resolution, In: 4th Asian Semantic Web Conference, China (2009)
8. Seddiqui, Md. Hanif and Aono M. An Efficient and Scalable Algorithm for Segmented Alignment of Ontologies of Arbitrary Size. J. Web Sem. 7(4), 344-356.
9. Taheri, A., Shamsfard, M. Consolidation of Linked Data Resources upon Heterogeneous Schemas, In Proceedings of the Sixth International Conference on Advances in Semantic Processing (SEMAPRO 2012), Spain, September 2012.
10. Fellbaum, C. WordNet: An Electronic Lexical Database,MIT Press, Cambridge, MA (1998)
11. Wu, Z. and Palmer, M. Verb Semantics and Lexical Selection. In: 32nd Annual Meeting of the Association for Computational Linguistics. Las Cruces (1994)
12. Euzenat, J. and Shvaiko, P. Ontology Matching, 1st ed., Springer, Berlin Heidelberg (2007)