ViewSameAs: A Novel Link in Instance Matching Process

Wafa Ghemmaz and Fouzia Benchikha
LIRE Laboratory, STIS Department, Abdelhamid Mehri Constantine2 University, Constantine, Algeria

Keywords: Data Integration, Semantic Web, Ontology, Linked Data, Instance Matching, ViewSameAs.

Abstract: In recent years, the Web has evolved from a global information space of interlinked documents to a space where both documents and data are linked. To integrate and share data, instance matching has been become the fundamental issue especially with the rapid development of linked data. In this paper, we propose an instance matching approach based on two main processes: the former is based on property classification (IM_PC) and the later is based on ViewSameAs link (IM_VSA). To accelerate greatly the matching process, IM_PC determines at first the matching candidate by comparing the discriminative property values. Then, the refinement result is done by comparing the description property values. In IM_PC two links are established: identity SameAs link and a novel proposed link ViewSameAs that aims to keep track of instances which share similar discriminative property values. In instance matching, another problem should be addressed when instances may have different descriptions even if their meanings are similar. So, this problem is addressed in IM_VSA process. The aim of this later is trying to get more identity link SameAs by Clustering instances matched with ViewSameAs. The Clustered instances are modeled as bags.

1 INTRODUCTION

In recent years, the Web has evolved from a global information space of linked documents to a space where data are linked as well. Actually, many Linking Open Data (LOD) datasets have been published on the Web. With the rapid growth in publishing interlinked datasets on LOD by various communities, data integration becomes inevitable and beneficial. Moreover, data integration on these interlinked datasets requires alignment techniques for concepts and properties in the schema level and instances in the data level. The problem of ontology matching (schema and data) has been widely studied in the last decade (Kalfoglou and Schorlemmer, 2003; Choi et al., 2006; Shvaiko and Euzenat, 2013a), many schema matching approaches were proposed such as ASMOV (Jean-Mary et al., 2009), PRIOR (Mao et al., 2010) and iMatch (Albagli et al., 2012). Recently, as the number of ontology instances grows rapidly, the problem on data level namely "instance matching" attracts increasingly more research interest (Li et al., 2013). Instance matching aims to link different instances that denote the same real-world object across heterogeneous data sources by establishing *SameAs* link between them (Bizer et al., 2007).

To resolve the instance matching problem, several approaches are proposed such as: VMI (Li et al., 2013), COMA++ (Engmann and Maßmann., 2007) and SIRIMI (Araujo et al., 2015). The problem in the existing approaches is that there is no method to save instances which share important properties values. For this reason, a novel link *ViewSameAs* is proposed. In instance matching, another problem should be addressed when instances may have different descriptions even if their meanings are similar. So, with the proposed link *ViewSameAs*, this problem can be solved.

In this paper, we propose an instance matching approach based on instance properties classification. Two main processes are included: the first consists on comparing instances using discriminative property values and descriptive property values. As a result, *SameAs* and *ViewSameAs* links are established. The second process consists on discovering more *SameAs* links by clustering some *ViewSameAs* ones.

The rest of the paper is organized as follows: section 2 is about some related works. An overview of our approach is given in section 3 and detailed in section 4. The proposed link *ViewSameAs* is presented in section 5. Finally, conclusion and future work are given in section 6.

Ghemmaz, W. and Benchikha, F.
ViewSameAs: A Novel Link in Instance Matching Process.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 274-279
ISBN: 978-989-758-186-1

2 RELATED WORKS

Several approaches dealing with the instance matching problem are proposed in the literature. They can be classified in two categories:

2.1 Approaches based on Instance Properties Classification

Many approaches are based on classifying instance properties including, for example, VMI (Li et al., 2013) in which instance information are classified in six categories: URI, Name, Meta, descriptive property values, discriminative property values and neighbors. The weakness of this approach is related to the fact that the authors use descriptive information firstly in their matching process. This information is less relevant compared to the discriminative information, which is more decisional while comparing two object's descriptions

Wang et al., (2013) classify the instances information in lexical information and structural information. The comparison of an entity in a dataset with all the entities of another dataset represents the weakness of this approach.

2.2 Approaches based on Interpretation of Instance Information

In these approaches, existing works use the similarity strategies or techniques to get more similar instances. For example, in COMA++ (Engmann and Maßmann, 2007), matching instances is based on two methods: content-based similarity and constraint-based similarity. Content-based similarity is based on string similarity functions such as edit-distance (Gusfield, 1997). Constraint-based similarity is based on numerical or pattern constraints of the ontology. The need to compare all instances of two ontologies represents the weakness of this approach.

In SIRIMI (Araujo et al., 2015), matching process combines direct-based matching with a class-based matching technique to infer *SameAs* relation over heterogeneous data.

There is a common weakness in the previous instance matching approaches. It concerns the final established link between similar instances. In these instance matching approaches, the identity link *owl:SameAs* is created between similar instances. This weakness arises when two instances have the same discriminative property values; including decisional and important information; and dissimilar descriptive property values.

In our approach, we propose a novel link *ViewSameAs* that will be established between instances which have similar discriminative property values. Because these last ones have an important weight in the matching process compared with the descriptive property values, *ViewSameAs* keeps the track of these instances.

Other classifications of instance matching approaches are described in (Shvaiko and Euzenat, 2013b; Ehrig, 2007).

3 APPROACH OVERVIEW

The traditional methods for instance matching usually try to find corresponding instances and compute similarity between an instance *i* in source ontology O_s and every instance in target ontology O_t. In the fact, there may be only a few possible instances in O_t that match *i*.

In instance matching, determining the matching candidate at first aims to accelerate greatly the matching process (Li et al., 2013). That represents the first challenge of our instance matching approach. To improve the efficiency of instance matching process, we try to find the matching candidate based on properties classification. For each instance, two types of instance information are distinguished: discriminative property values and descriptive property values.

- The discriminative property values are the characteristics of the instances which can be used directly to distinguish them.
- The descriptive property values are the descriptions of an instance.

In instance matching, another problem should be addressed when instances may have different descriptions even if their meanings are similar. So, in our approach, we propose a novel link *ViewSameAs* which aims to keep the track of instances that share discriminative property values.

Our approach takes two ontologies as input:O_s and O_t. For every instance $i_s \in O_s$, the goal is to find matching instances $i_t \in O_t$.The proposed approach contains two main processes: *Instance Matching process based on Property Classification* (IM-PC) and *Instance Matching process based on ViewSameAs* link (IM-VSA) as illustrated in Figure1

- ***IM-PC***: is performed in two main steps (Ghemmaz and Benchikha, 2015): the candidate selection and the result refinement. The former is

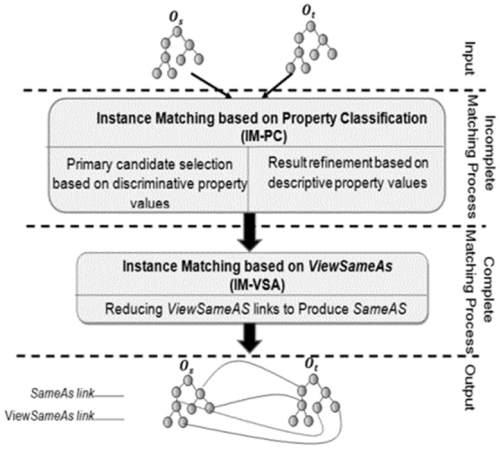

Figure 1: The proposed approach.

based on the discriminative property values and the later is based on descriptive property values. Once the final result is obtained, two types of link are established: *SameAs* and *ViewSameAs*.

- ***IM-VSA***: to get more improved result, this process tries to find more *SameAs* links by reducing *ViewSameAs* links.

4 INSTANCE MATCHING APPROACH

The proposed instance matching approach consists of the following two processes: IM-PC is based on the type of instances information to identify corresponding instances and IM-VSA is implied to get more correspondences based on *ViewSameAs* links as illustrated in Figure 2. We introduce our approach in more detail below.

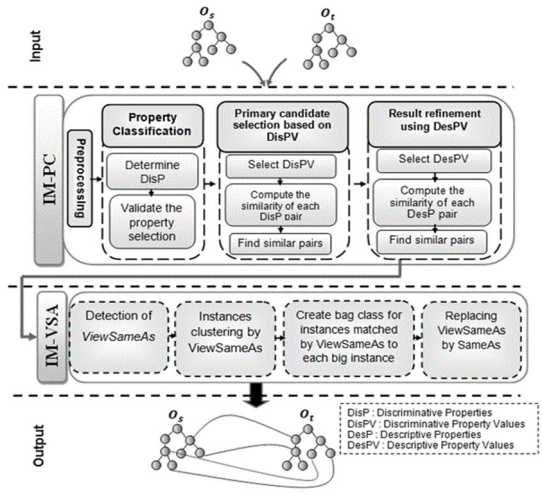

Figure 2: Instance matching approach.

4.1 IM-PC

IM-PC is composed of four main stages. In the next sub-sections, we give a description of each stage.

4.1.1 Pre-processing

At this level, all the properties and instances information of two ontologies O_s and O_t are extracted.

4.1.2 Properties Classification

In this stage instances' properties are classified as discriminative properties and descriptive properties. Some of discriminative properties can be selected automatically; the typical example is rdf:type. Others must be specified by an expert. Once all the discriminative properties have been selected, the other properties are considered as descriptive ones.

Figure 3 presents an example of a person instance. The properties «rdf:type», «HasSex», «HasMail» and «rdf:label» are considered as discriminative ones with discriminative values «foaf:person», «Female», «fouzia_benchikha@ univ-constantine2.dz» and «fouzia benchikha» respectively. The descriptive properties are «affliationDate», «hasTitle», «StudiedModules».

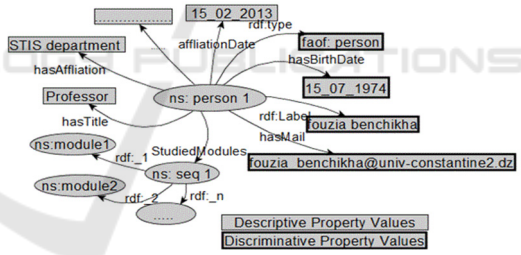

Figure 3: An example within an instance.

4.1.3 Primary Candidate Selection based on Discriminative Properties

In this step, detailed in Algorithm1, all instances' properties won't be compared at the same time. To determinate the matching candidates, we start by comparing the discriminative property values. However, having two ontologies O_s and $O_t$with the set of their instances I_s and I_t respectively, we generate; for each instance i_s in I_s and for each instance i_t in I_t; the discriminative property values $\mathrm{DisPV_s}$ and $DisPV_t$ respectively. Then, each i_s *in* $DisPV_s$ will be compared with each i_t *in* $DisPV_t$ by the similarity function $CalculateSim(i_s, i_t)$. γ is a similarity threshold denoting the minimum level of

matching required for considering two instances as similar ones. The algorithm output is *AlignDP* including instances considered as partially similar and that will be more compared in the next stage.

Algorithm 1: Candidate Selection based on discriminative property values.

```
Input: I_s and I_t.
Output: AlignDP.
1. DisPVs←∅ , DisPVt←∅, AlignDP← ∅.
2. For each i_s ∈ I_s
3.  DisPV_s = DisPV_s ∪ generateDisPV(i_s)
4. For each i_t ∈ I_t
5.  DisPV_t = DisPV_t ∪ generateDisPV(i_t)
6. For each (i_s ∈ DisPV_s) and (i_t ∈ DisPV_t)
7.   Conf_i = CalculateSim(i_s, i_t)
8.  If Conf_i ≥ γ
9.   AlignDP ←AlignDP ∪ (i_s, i_t, conf_i)
10.  End if
11. Return AlignDP
```

4.1.4 Result Refinement using Descriptive Properties Values

Descriptive property values of instances in *AlignDP,* obtained in the previous stage, are compared using the *CalculateSim*(i_s, i_t) function (see Algorithm2).

Algorithm 2: Result refinement based on descriptive property values.

```
Input: AlignDP.
Output: AlignSA, AlignVSA.
1. AlignSA← ∅, AlignVSA← ∅
2. For each i_s ∈ AlignDP
5.  DesPV_s=generateDesPV(i_s)
6. For each i_t ∈ AlignDP
7.  DesPV_t=generateDesPV(i_t)
8. For each (i_s ∈ DesPV_s) ∧ (i_t ∈ DesPV_t)
9.  Conf_i = CalculateSim(DesPV_s, DesPV_t)
10. If Conf_i ≥ σ
11.  AlignSA←AlignSA ∪ (i_s, i_t, conf_i, SameAs)
12. Else
13. AlignVSA←AlignVSA ∪ (i_s, i_t, conf_i, ViewSameAs, vote_i)
14. End if
15. Return AlignSA, AlignVSA
```

Instances that have similarity value more than σ are considered as similar ones. The output of Algorithm2 is: (i) AlignSA including a set of quadruplet $(i_s, i_t, conf_i, SameAs)$ and (ii) AlignVSA including a set of quintuplet $(i_s, i_t, conf_i, ViewSameAs, vote_i)$. SameAs link is created between instances that have similar discriminative and descriptive property values and $ViewSameAs$ link is established between instances that have similar discriminative property values and dissimilar descriptive property values. $vote_i$ refers to the number of similar property values between each instance pair and is used to establish identity link SameAs based on the proposed link ViewSameAs.

4.2 IM-VSA

The aim of this process is to deal with the possibility to get more identity link *SameAs.* IM-VSA is basically made of four main steps presented below.

4.2.1 Detection of ViewSameAs

The first step of IM_VSA allows detecting instances matched with the proposed link *ViewSameAs* in order to match them using the identity link *SameAs*. Figure 4 illustrates an example of person instance that is represented in different contexts. The instances *person1, person2, person3* and *person4* refer to the same object: *Benchikha fouzia.*

- *Person1* is an instance defined in "*University ontology"*,
- *Person2* is an instance defined in "*Laboratory ontology"*,
- *Person3* is an instance defined in "*Insurance ontology "*
- *Person 4* is defined in "*Social Ontology*".

These instances share the same discriminative property values but each of them has a special description according to a specified context or viewpoint. Thus, the proposed link *ViewSameAs* is generated between each pair of instances. We argue that the descriptive property values of person1, person2, person3 are included in the set of descriptive property values of person4.

4.2.2 Instances Clustering

The goal of this step is to cluster instances matched with *ViewSameAs*. Thus, for each instance i_s to i_{t1}, i_{t2} ... i_{in} with *ViewSameAs*, an instance Cluster *Cluster x* is represented as:

Cluster x: $(i_s, i_{t1}, conf_1, ViewSameAs, vote_1)$.
$(i_s, i_{t2}, conf_2, ViewSameAs, vote_2)$.
.....
$(i_s, i_{tn}, conf_n, ViewSameAs, vote_n)$.

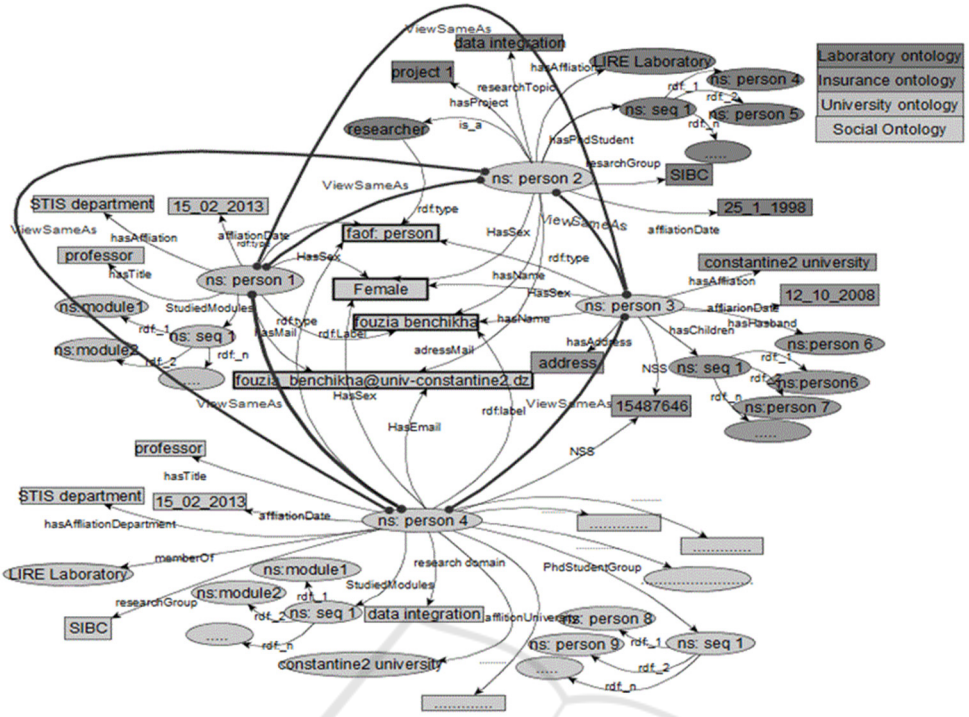

Figure 4: Cloud of person instances.

Based on the example presented in Figure 4, the instances Clusters are:

Cluster1: (**person1**, person2, $conf_1$, ViewSameAs, 0).
(**person1**, person3, $conf_2$, ViewSameAs, 0).
(**person1**, person4, $conf_3$, ViewSameAs, 5).
Cluster2: (**person2**, person1, $conf_1$, ViewSameAs, 0)
(**person2**; person3, $conf_4$, ViewSameAs, 0).
(**person2**, person4, $conf_5$, ViewSameAs, 3).
Cluster3: (**person3**, person1, $conf_2$, ViewSameAs, 0)
(**person3**, person2, $conf_4$, ViewSameAs, 0).
(**person3**, person4, $conf_6$, ViewSameAs, 4).
Cluster4: (**person4**, person1, $conf_3$, ViewSameAs, 5)
(**person4**, person2, $conf_5$, ViewSameAs, 3).
(**person4**, person3, $conf_6$, ViewSameAs, 4).

4.2.3 Creating Instance Bag

The instances bag can be only created if and only if the descriptive property values of the target instances are included in the descriptive property values of the source instance (big instance). For example, each of Cluster 1, Cluster 2 and Cluster 3 can't be considered as a collection of instances that refer to the same instance person1, person2 and person3 respectively. However, in Cluster 4 the instances person1, person2 and person3 can be considered as a collection of instances that refer to the same instance person4. In this case, a bag will be created for these instances and an identity link *SameAs* between this bag and person4 will be created. Person4 can be called the big instance.

4.2.4 Replacing ViewSameAs Link by SameAs Link

ViewSameAs links are conserved in the bag and *SameAs* link will be created. This last is established between instances in bag and the big instance. The result of IM-VSA applied on the example below is schematically represented in Figure 5.

Figure 5: From ViewSameAs to SameAs links.

5 THE ViewSameAs LINK

The proposed link *ViewSameAs* has the following advantages:

- It keeps the track of instances which share important properties especially discriminative

ones. These instances could be identical and refer to the same real word object or they could be different but considering as similar according to an agent viewpoint (Ghemmaz and Benchikha, 2015).

Based on the example presented in Figure 4, person1 and person2 refer to the same real-world object but each of them is described in a specified context as illustrated in Figure 6.

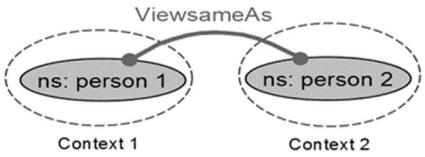

Figure 6: An example within an instance in different contexts.

- It helps to Cluster instances that refer to the same instance as presented in Figure 5 for keeping discovered *SameAs*.
- In the case of insertion or updating operation, it eliminates the comparison of instances which judged definitively different, and, it improves the search time of instances which share some discriminative property values.

In order to prove the efficiency of the proposed link *ViewSameAs*, we are currently working on its validation using existing datasets.

6 CONCLUSIONS

In this paper, we have presented an instance matching approach based on instance properties classification. It consists of two main processes, the first one is based on the discriminative property values and the second one is based on a novel *ViewSameAs* link. In our approach, two types of links will be established between similar instance pairs: *SameAs* link and *ViewSameAs* link. This last is proposed to keep the track of instances which share similar discriminative property values. Currently, we are working on the validation of our instance matching approach, which implies the validation of the *ViewSameAs* link.

An experiment will be carry out by using dataset from OAEI (Ontology Alignment Evaluation Initiative).The result and the performance of the proposed approach will then be further discussed.

REFERENCES

Albagli, S., Ben-Eliyahu-Zohary, R. and Shimony, S. 2012. Markov network based ontology matching. *Journal of Computer and System Sciences*, 78(1), pp.105-118.

Araujo, S., Tran, D., de Vries, A. and Schwabe, D. 2015. SERIMI: Class-Based Matching for Instance Matching Across Heterogeneous Datasets. *IEEE Trans. Knowl. Data Eng.*, 27(5), pp.1397-1440.

Bizer, C., Cyganiak, R. and Heath, T. 2007. *How to publish Linked Data on the Web*. Available at: http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/.

Choi, N., Song, I. and Han, H. 2006. A survey on ontology mapping. *ACM SIGMOD Record*, 35(3), pp.34-41.

Ehrig, M. (2007). *Ontology alignment*. New York: Springer.

Engmann, D., Maßmann, S., 2007. Instance matching with COMA++. *In: Proceedings of Datenbanksysteme in Business, Technologie and Web (BTW 07),* pp. 28–37.

Ghemmaz, W., Benchikha, F. 2015. Instance Matching based on the Discriminative Property Values, Paper presented at *the 5th International Conference on Information and Communication Technology and Accessibility (ICTA 2015), Morroco, December 21-23.*

Gusfield, D. 1997, Algorithms on Strings Trees and Sequences, *Cambrige University Press*.

Jean-Mary, Y., Shironoshita, E. and Kabuka, M. 2009. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), pp.235-251.

Kalfoglou, Y. and Schorlemmer, M. 2003. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1), pp.1-31.

Li, J., Wang, Z., Zhang, X. and Tang, J. 2013. Large scale instance matching via multiple indexes and candidate selection. *Knowledge-Based Systems*, 50, pp.112-120.

Mao, M., Peng, Y. and Spring, M. 2010. An adaptive ontology mapping approach with neural network based constraint satisfaction. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1), pp.14-25.

Shvaiko, P. and Euzenat, J. 2013a. Ontology Matching: State of the Art and Future Challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1), pp.158-176.

Shvaiko, P. and Euzenat, J. 2013b. Ontology Matching. Springer Berlin Heidelberg, 2nd edition.

Wang, Z., Li, J., Zhao, Y., Setchi, R. and Tang, J. 2013. A unified approach to matching semantic data on the Web. *Knowledge-Based Systems*, 39, pp.173-18.