International Journal of Web-Based Learning and Teaching Technologies
Volume 16 • Issue 4 • July-August 2021

Reusing Alignments for Discovering Instances Correspondences

Wafa Ghemmaz, LIRE Laboratory, Abdelhamid Mehri Constantine 2 University, Algeria

Fouzia Benchikha, LIRE Laboratory, Abdelhamid Mehri Constantine 2 University, Algeria

Maroua Bouzid, GREYC Laboratory, University of Caen Normandie, France

ABSTRACT

Recently, instance matching has become a key technology to achieve interoperability over datasets, especially in linked data. Due the rapid growth of published datasets, it attracts increasingly more research interest. In this context, several approaches have been proposed. However, they do not perform well since the problem of matching instances that possess different descriptions is not addressed. On the other hand, the usage of the identity link owl:sameAs is generally predominant in linking correspondences. Unfortunately, many existing identity links are misused. In this paper, the authors discuss these issues and propose an original instance matching approach aiming to match instances that hold diverse descriptions. Furthermore, a novel link named ViewSameAs is proposed. The key improvement compared to existing approaches is alignment reuse. Thus, two novel methods are introduced: ViewSameAs-based clustering and alignment reuse based on metadata. Experiments on datasets by considering those of OAEI show that the proposed approach achieves satisfying and highly accuracy results.

KEYWORDS

Data Integration, Instance Matching, Linked Data, Metadata, sameAs, Semantic Web, ViewSameAs

INTRODUCTION

Data integration has been widely studied by the database community. In the web of data and especially in linked data, it becomes one of the main issues in data sharing and exploitation. However, ontology matching is one of the crucial tasks that support data integration where identifying correspondences presents a challenging problem. Instance matching (IM) is a subtask of ontology matching and seems to be an interesting solution. IM aims at discovering correspondences between instances, from various sources, that refer to the same real-world objects. Despite recent advances, IM still presents a real problem especially with the rapid growth of published data (Cukier & Mayer-Schoenberger, 2013**;** Zhang et al., 2008). It is a long-standing issue known as record linkage (Newcombe et al., 1959), merge/purge problem (Hernández & Stolfo, 1995) or reference reconciliation (Dong et al., 2005). For its importance, several approaches have been proposed such as ASL (Nguyen & Ichise, 2018), AIM-PC (Lu et al., 2018), RIMOM-IM (Shao et al., 2016), SERIMI (Araujo et al., 2011; 2015) and VMI (Li et al., 2013), approach of (Wang et al., 2013), FBEM (Stoermer & Rassadko, 2009), DSSim (Nagy et al., 2008) and HMatch(I) (Castano et al., 2008). Although the existing approaches work well in many cases of IM, they could fail to detect correspondences between instances that possess different descriptions.

DOI: 10.4018/IJWLTT.20210701.oa5

In IM, identifying correspondences or matches is achieved by establishing links between instances from multiple sources to be integrated. The most important link is the owl:sameAs one. However, a significant number of existing owl:sameAs links on the Web of data do not adhere to their formal semantic (Halpin et al., 2010). This is due to the diverse contexts of instances descriptions. For example, when the similarity score between two instances indicates that they are different, they should be considered as non-matches. But in fact, they could refer to the same real-world object. Another issue of misidentified instances occurs when two instances referring to different objects are still connected by owl:sameAs link. For these reasons, many studies have shown that the owl:sameAs can lead to inconsistencies. Some of them propose novel constructors or predicates to replace owl:sameAs (Raad et al., 2017; Idrissou et al., 2017; Halpin et al, 2010), while other works focus on the detection of incorrect or quality sameAs statements (De Melo, 2013; Papaleo et al., 2014). Our purpose is to deal with the problem of discovering more correct links between instances with multiple descriptions and that are provided by various sources.

In this paper, we propose a novel approach dealing with the IM problem where matching instances with diverse descriptions presents the principal aim. This problem is poorly addressed in the literature. The proposed approach is based on three processes: IM-PC (IM based on Property Classification), IM-VSA (IM based on ViewSameAs) and IM-AMD (IM based on Alignment MetaData). The first process aims to maximize the utilization of discriminative information within instances where a novel link is introduced named ViewSameAs. This last allows connecting partially similar instances. The second process IM-VSA allows discovering more correspondences by reusing the alignment results of IM_PC. A novel method named ViewSameAs-based clustering is thus proposed. By using metadata about IM_VSA, the third process IM_AMD allows improving our IM solution. In fact, the alignment reuse is considered as a new and helpful technique in IM where few works used it such as RIMOM-IM (Shao et al., 2016). The contributions of this paper can be summarized as follows:

- The IM problem is presented where two main issues are discussed: the correspondences detection through an IM approach and correspondences representation through the identity link *owl:sameAs*.
- A novel IM approach that addresses the problem of matching instances with diverse descriptions is presented.
- A novel link named *ViewSameAs* allowing linking partially similar instances is proposed. Moreover, it is helpful in detecting sameAs links. On the other hand, we propose other predicates that could be served as metadata to refine the matching result.
- Novel methods allowing alignment reuse (ViewSameAs-based clustering method and metadata) are proposed.
- The proposed approach is validated on several datasets by considering those of OAEI. Experimental results show that our proposal achieves satisfying results.

The remainder of the paper is organized as follows. In the next section, we present someresearch works related to the IM problem. A motivating example followed by an overview of the proposed solution is provided in section 3. In section 4, we explain the proposed approach in detail by giving at first general terminology of some useful concepts. Then, we explain each proposed process where formal definitions are provided. In section 5, we exploit a set of datasets; by considering those of OAEI 2009 and 2010; and then we provide and discuss the experimental results. In section 6, we discuss some related works. Finally, as conclusion, some prospects and future works are provided in section 7.

INSTANCE MATCHING IN LINKED OPEN DATA

IM techniques are interesting solutions to data integration in linked open data. In this regard, two main issues dealing to the IM problem are emerged: the correspondences detection through an IM approach and correspondences representation through the identity link *owl:sameAs*.

IM Approaches

In the literature, several approaches dealing with the IM problem are proposed. Diverse classifications are provided (Ghemmaz & Benchikha, 2016; Euzenat & Shvaiko, 2013; Ehrig, 2007). In fact, the result quality of an IM approach depends on (i) the correct using of the instances information and (ii) the measures utilized to find instances' similarity. Relying on this principle, the authors in (Ghemmaz & Benchikha, 2016) classified IM approaches in two categories: approaches based on instance properties classification and approaches based on interpretation of instances information.

Approaches based on instance properties classification: the matching process is principally based on the instances' information (properties). This information is classified in several types where each type is used in a given task to achieve a given goal. By analyzing existing classification, data structure and data semantic are the main axes in distinguishing information type. For example, VMI (Li et al., 2013) classifies the instances information in several categories. RIMOM-IM (Li et al., 2009; Shao et al., 2016; Tang et al., 2006) uses the semantic within instance properties in the matching process such as the distinctive information. Wang et al., (2013) classify the instances information in lexical and structural information.

Approaches based on the interpretation of instance information: the matching process is principally based on functions and methods to get the best similarity score. Some works combined existing basic similarity methods such as AIM-PC (Lu et al., 2018), ASL (Nguyen & Ichise, 2018) and SERIMI (Araujo et al., 2011; 2015), while others proposed novel formulas like FBEM (Stoermer & Rassadko, 2009), DSSim (Nagy et al., 2008) and HMatch (Castano et al., 2008).

Like all these cited approaches, we are interested to the IM problem. However, unlike them, we are interested in detecting correspondences between instances that hold diverse descriptions. As far as we know, no other work addresses this problem. On the other hand, in all these approaches, the identity link *owl:sameAs* is established between each corresponding instances pair. In web of data, the *owl:sameAs* link plays an important role in connecting datasets. However, it is very important to correctly decide whether two URIs refer to the same real-world object or not. In fact, several studies have confirmed the existence of significant numbers of *sameAs* links on the Web of data that do not adhere to their official semantics (Halpin et al, 2010; Idrissou et al., 2017; Ding et al., 2010; Bizer et al., 2007; De Melo, 2013; Raad et al. 2017; McCusker & McGuinness, 2010)**.**

Identity Within *owl:sameAs*

In linked data, identity links sameAs allow data integration without having to agree on uniform schemas or vocabularies. They are indispensable for making datasets interoperable. Due the huge quantity of published and independently developed datasets, the problem of identity becomes one of the most important challenges that should be taken into consideration when constructing and maintaining links. It is a long-standing issue and it is not particularly related to the linked data. It has been more and more studied and attracted more attention when encountered by different individuals attempting to independently knit their knowledge representations together using the same standardized language (Halpin et al., 2010).

The *owl:sameAs* is defined as stating "*that two URI references* [1] *actually refer to the same thing*" (Bizer et al., 2007). However, this notion of identity is problematic when entities or objects are considered the same in some contexts but different in others. A study on identity within *owl:sameAs* in (Halpin et al., 2010) showed that it is misused. As a result, similarity ontology (SO) has been defined including new identity properties derived from the original meta-properties of sameAs. These properties capture an important intuition specifically the inexistence of a strict definition of identity in the use of sameAs on the web. In an empirical study (Ding et al., 2010), authors propose a general strategy including several components to integrate information from the URIs in an *owl:sameAs* network. Then, they collect and analyze the naturally occurring of *owl:sameAs* networks to identify several issues involving the *owl:sameAs* property.

In (De Melo, 2013), a novel predicate has been defined for genuine identity named "*lvont:strictlySameAs*". This predicate belongs to the Lexvo.org Ontology and allows knowing whether a sameAs link was indeed intended in the strict sense or in a looser near-identity sense. Lexvo.org ontology provides also two near-identity predicates such as *lvont:nearlySameAs* and *lvont:somewhatSameAs*. Idrissou et al. (2017) have shown that the quality of the *sameAs* link depends not only on the resources' properties but also on the purpose or the task for which they are used. A system for constructing context-specific equality links is proposed. These links are decorated with rich metadata describing how, why, when and by whom they were generated. Instead of *owl:sameAs*, the *context:sameAs* link is proposed. This last provides a generic metadata that allow alignment reproducibility, and specific correspondence metadata for context-specific reusability and validation. In (Raad et al., 2017), authors propose a new contextual identity link named *identiConTo* that could be served as a replacement for *owl:sameAs* in linking identical instances in a specified context. They define also an algorithm for detecting contextual links named DECIDE. This last allows the detection of the most specific global contexts in which a couple of instances are identical.

In fact, most researches agree that the *sameAs* link is abused. Moreover, it is always possible to specify new constructs in semantic web to identify equality relations. As discussed above, existing works propose novel constructors to replace owl:sameAs or to detect incorrect or quality sameAs statements. In this paper, novel constructors are introduced: *ViewSameAs* for linking partially similar instances, *hasBigInstance* and *hasBagClass* which are served as metadata to refine the matching result.

MOTIVATION AND SOLUTION OVERVIEW

The integration of published data is mainly based on the construction of *owl:sameAs* links between them. Indeed, it is difficult to detect identity relations between instances with various descriptions where the similarity over all their properties indicates that they are different. Moreover, the similarity over their important properties is not sufficient to decide if they are similar or not. The following motivating examples identify some problems encountered in IM process. Then, we present an overview of our proposed solution to address them.

Motivating Example

In Table 1, two datasets to integrate are given: the source and the target datasets. The instance "sd:AhmedAli" from the source dataset and the instance "td:AhmedAli" from the target dataset refer to the same real-world object but each of them has specific characteristics depending on the description context. In this case, their matching presents a challenging problem. In another example, the source instance "sd:LIRE_Laboratory" and the target instance "td:Constantine2_University" have many attribute values in common but they refer to two different objects. In this case, establishing *owl:sameAs* link between them gives an incorrect representation of the reality even if they refer to the same localization. Thus, the problem of IM raises new challenges summarized in the following questions:

(1) What about instances holding different descriptions but refer to the same real world object?
(2) What about different instances that could be strongly similar depending on a user's viewpoint?

Table 1. Example of RDF[2] Triples

Source Dataset		
Subject	Predicate(attribut)	Object (Values)
sd:AhmedAli	rdfs:label	‘Ahmed Ali’
sd:AhmedAli	rdfs:type	foaf: person
sd:AhmedAli	sd:Sex	‘male’
sd:AhmedAli	sd:Birthdate	’15-07-1974’
sd:AhmedAli	sd:HasAfiliation	sd:LIRE_Laboratory
sd:AhmedAli	sd: Email	‘ahmedAli@univ-c2.com’
sd:AhmedAli	sd:is_a	‘researcher’
sd:AhmedAli	sd:ResearchGroup	‘SIBC’
Sd:AhmedAli	sd:HasProject	sd:prjXXY
sd:AhmedAli	sd:ResearchTopics	‘data integration’
sd:LIRE_Laboratory	rdfs:Label	‘Laboratoire d’Informatique Répartie’
sd:LIRE_Laboratory	sd:Address	‘ABDELHAMID MEHRI university, uneamed road, Ali Mendjli, Algeria’
sd:LIRE_Laboratory	sd:Latitude	‘36.244813’
sd:LIRE_Laboratory	sd:Longitude	‘6.568329700000049’
sd:LIRE_Laboratory	sd:HasDirector	sd:person1
sd:LIRE_Laboratory	sd:HasMember	sd:seq1
…	…	…
Target Dataset		
Subject	Predicate(attribute)	Object (Values)
td:Constantine2_University	td:HasLatitude	‘36.244813’
td:Constantine2_University	td:HasLongitude	‘6.568329700000049’
td:Constantine2_University	td:HasDepartment	td:Seq1
td:Constantine2_University	td:HasAddress	‘ABDELHAMID MEHRI university, uneamed road, Ali Mendjli, Algeria’
td:Constantine2_University	td:HasRector	td:person1
td:Constantine2_University	td:CreationDate	‘08_11_2011’
td:AhmedAli	td:HasName	‘Ahmed Ali’
td:AhmedAli	rdfs:type	foaf: person
td:AhmedAli	td:HasBirthdate	’15-07-1974’
td:AhmedAli	td:HasSex	‘male’
td:AhmedAli	td:HasAfiliation	td:STIS
td:AhmedAli	td:HasEmail	‘ahmedAli@univ-c2.com’
td:AhmedAli	td:HasTitle	‘Professor
td:AhmedAli	td:AfiliationDate	’15-11-2009’
…	…	…

To answer these questions, we propose an IM approach based on properties classification. Important properties; called discriminative; are used to create links Link_j describing a certain similarity degree between instances. Descriptive properties are specific properties according to particular contexts. They are used to refine and complete the matching providing correct matches.

For example, we can have as result:

```
               (sd:AhmedAli  Link_i    td:AhmedAli)
And      (sd:LIRE_Laboratory  Link_i   td:Constantine2_University)
Where:          Link_i= sameAs or Link_i≠sameAs   depending on if the
matching is complete or partial respectively.
```

The instances "sd:Ahmed Ali" and "td:Ahmed Ali" refer to the same person and share important properties in common such: label, type, Sex, Birthdate and Email. However, these properties are not sufficient to establish sameAs relation between them. For example, the instances pair (sd:IRE_ Laboratory, td:Constantine2_University) share the same geographical extents (Latitude, Longitude and Address) as important properties but they don't refer to the same object ($\text{Link}_i \neq$ sameAs). The approach that we propose allows exploiting the information intelligently within instances to produce correct links.

Solution Overview

To solve the challenges discussed above, we present an original IM approach allowing matching instances that possess diverse descriptions. It is based on three main principles:

1. Classifying instances properties in discriminative and descriptive properties.
2. Matching the partially similar instances by the proposed link named *ViewSameA*s (*ViewSameAs* ≠ sameAs).
3. Producing sameAs links between the partially similar instances (if possible).

As illustrated in Figure1, the proposed approach contains three main processes: (1) *IM based on Property Classification* process *(IM-PC)* (2) *IM based on ViewSameAs* process *(IM-VSA)* and (3) *IM based on Alignment Meta-Data* process *(IM-AMD).*

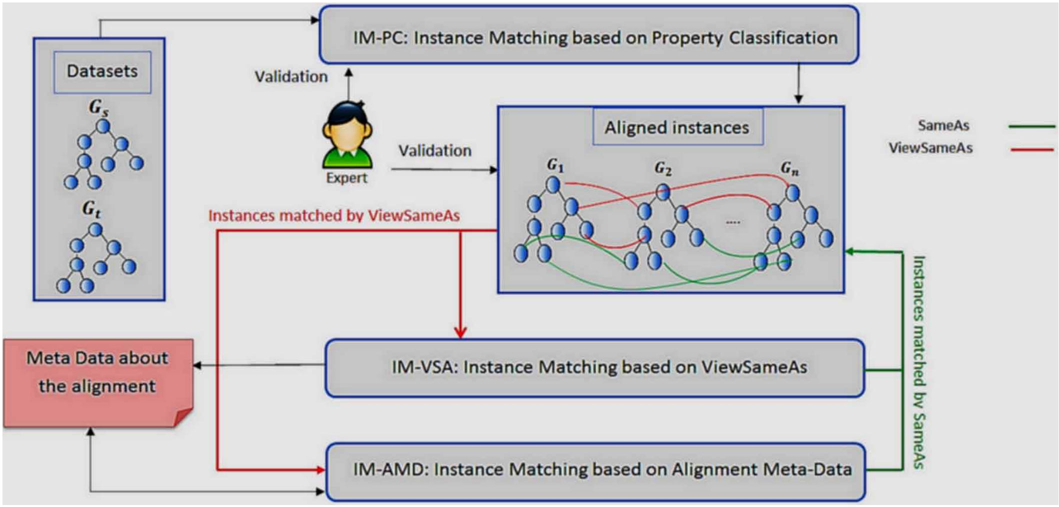
1. ***IM_PC:*** this process allows correspondences detection based on properties classification. As result, it produces a set of similar instances matched by *sameAs* link and a set of partially similar instances connected by the proposed link *ViewSameAs.*
2. ***IM-VSA:*** the aim of this process is to deal with the possibility to find more correspondences by using the alignments results produced by IM_PC. Basing on the *ViewSameAs*, a novel method is introduced named *ViewSameAs*-based clustering
3. ***IM-AMD:*** this process re-uses the IM_VSA results to detect more correspondences. It bases principally on a set of metadata where two novel predicates are introduced: *hasBigInstance* and *hasBagClass.*

In the proposed approach, the IM task is based principally on the alignment reuse which is considered as a new and helpful technique in IM. In section 4, we provide more explanation on the proposed processes and techniques.

THE DETAILED APPROACH

In this section, we detail each process and each step of the proposed approach. Before this, a general terminology is presented. It concerns basic definitions that will be used through the rest of the paper.

Figure 1. The proposed approach


IM-PC: Instance Matching based on Property Classification
Datasets
G_s
G_t
Validation
Expert
Validation
Aligned instances
G_1
G_2
G_n
....
SameAs
ViewSameAs
Instances matched by ViewSameAs
Instances matched by SameAs
Meta Data about the alignment
IM-VSA: Instance Matching based on ViewSameAs
IM-AMD: Instance Matching based on Alignment Meta-Data

Definition 1 (Data Graph): *The data is conceived as a set of RDF Graphs* **G.** *Let U denote the set of Uniform Resource Identifiers and L denote the set of literals. Every* $G \in \mathbf{G}$ *is a set of triples of the form* s,p,o *where s and p* $\in U$ *(subject and predicate) and o* $\in U \cup L$ *(object).*

Definition 2 (Instance features): *Let G be a dataset and I the set of instances in* G *. The features of a given instance i are:*

$$P(i)= \left\{p \middle| (s,p,o) \in G \wedge s \in I\right\} \tag{1}$$

$$O(i)= \left\{o \middle| (s,p,o) \in G \wedge s \in I \wedge o \in \left(L \cup U\right)\right\} \tag{2}$$

Note *P(i) is the set of predicates and O(i) the set of literals and URIs (objects).*

Definition 3 (Instance Matching): *Given two input dataset* G_s *and* G_t*,* G_s *is called the source dataset and* G_t *is called the target dataset. IM is defined to find corresponding instances in* G_t *for each instance in* G_s *. The result of IM can be represented as:*

$$InstAlign\left(G_s,G_t\right) = \{\left(i_x,i_y,conf\right) | i_x \in I_s, i_y \in I_t, conf \in \left[0,1\right]\} \tag{3}$$

Each 3-tuple $\left(i_x,i_y,conf\right)$ *in* $InstAlign\left(G_s,G_t\right)$ *indicates that instance* i_x *in* I_s *(set of instance in* G_s *) is matched to instance* i_y *in* I_t *(set of instances in* G_t *) with the confidence* $conf$ *.*

Definition 4 (sameAs): The *sameAs link is defined to link entities referring to the same real world object. Given two instances* i_x *and* i_y *from* I_s *and* I_t *respectively.*

$$\left(i_x, i_y, SameAs\right) if\, and\, only\, if: \qquad \forall i_x \in I_s \ , \forall i_y \in I_t, conf \geq \delta \tag{4}$$

Each $i_x \in I_s$ *is matched to* $i_y \in I_t$ *by the sameAs if the confidence value* $conf$ *is more than or equal to a predefined threshold* ′ *.*

The definitions 2, 3 and 4 are formalized in a novel and proper formula according to our IM solution.

IM_PC

The IM_PC is the first process of the proposed approach. It includes four sequential steps: pre-processing, property classification, primary candidate selection and result refinement.

Pre-Processing

At this level, all the instances' information of two dataset G_s and G_t are extracted to classify them in the next stage. Due to the variations and errors in the data, the instances' information is usually not directly comparable. All the properties should be consistent to each other. Moreover, those that contain missing values and stop words; like "a" and "the"; must also be removed. In the proposal of this paper, the similarity is computed by using COSINE similarity which is based on word segmentation. Thus, nature language processing (NLP) technology can be used to segment the text into words (Manning et al, 2014).

Property Classification

For IM task, it is possible to select a subset $P^{'}\left(i\right) \subset P\left(i\right)$. In this way, each instances pair comparison requires only $\mid P^{'}\left(i\right) \mid$ similarity computations which is far less than $\mid P\left(i\right) \mid$. We classify instances properties as discriminative properties (DisP) and descriptive properties (DesP).

Definition 5 (DisP and DesP): *The discriminative properties DisP are the vital and key properties that characterize a given instance i. The descriptive properties DesP are the properties that hold a specified description of i depending on a given context. Each* $p \in P\left(i\right)$ *can be DisP or DesP:*

$$P\left(i\right) = DisP\left(i\right) \cup DesP\left(i\right) \tag{5}$$

$$DisP\left(i\right) \subset P\left(i\right) \wedge DisP\left(i\right) \cap DesP\left(i\right) = \phi \tag{6}$$

$$DesP\left(i\right) \subset P\left(i\right) \wedge DesP\left(i\right) \cap DisP\left(i\right) = \phi \tag{7}$$

Note that $DisP(i)$ can be selected automatically; the typical example is *rdf:type, owl:FunctionalProperty* and *owl:InverseFunctionalProperty*. Others must be selected and validated by an expert (semi-automatic step). Actually, to compare the property values, it's indispensable to know the matching properties and the mappings before starting the matching process. For example, we cannot compare the death date and the birth date of a person; this surely will produce wrong result, that's why a schema matching tool and the expert validation are indispensable. Once all the $DisP$ have been selected, the other properties are considered as $DesP$. After classification, IM_PC generates for each *i* the set of DisP Values (DisPV) while the DesP Values (DesPV) will be produced in the third stage.

Definition 6 (DisPV and DesPV): *The discriminative property values DisPV and the descriptive property values DesPV are the instantiation of DisP and DesP respectively. Each* $o \in O(i)$ *associated to instance i over the predicates p can be a DisPV or DesPV:*

$$p \in DisP(i) \Leftrightarrow o \in DisPV(i) \tag{8}$$

$$p \in DesP(i) \Leftrightarrow o \in DesPV(i) \tag{9}$$

$$O(i) = DisPV(i) \cup DesPV(i) \tag{10}$$

$$DisPV(i) \subset O(i) \wedge DisPV(i) \cap DesPV(i) = \phi \tag{11}$$

$$DesPV(i) \subset O(i) \wedge DesPV(i) \cap DisPV(i) = \phi \tag{12}$$

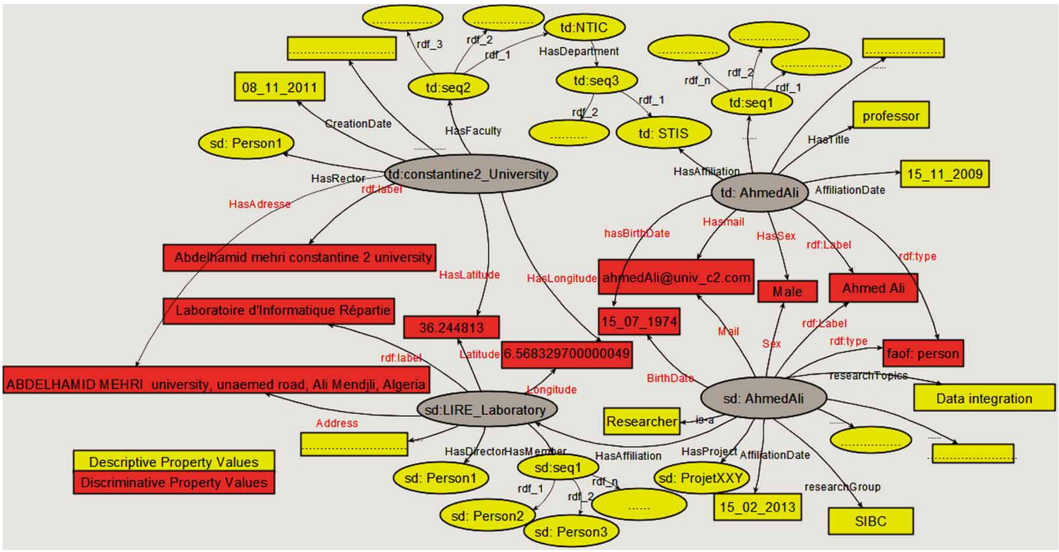
Example: The Figure 2 shows an example of instances where:

- "sd:AhmedAli" and "td:AhmedAli" refer to the same person, share in common the same DisP and DisPV while the DesP and DesPV are different.
- "sd:LIRE_Laboratory" and "td:Constantine2_University" refer to two different objects with different DesP and DesPV while possessing the same DisP and DisPV and referring to the same localization.

Primary Candidate Selection Based on DisPV

To determine the matching candidates, IM_PC starts by comparing DisPV. It generates; for each i_x in I_s and i_y in I_t; the $DisPV(i_x)$ and $DisPV(i_y)$ to obtain the sets $\mathrm{DisPV}_{\mathrm{s}}$ and $DisPV_{\mathrm{t}}$ respectively where $DisPV_s = \bigcup_{i_x \in I_s} DisPV(i_x)$ and $DisPV_t = \bigcup_{i_y \in I_t} DisPV(i_y)$. Then, the $DisPV_{\mathrm{s}}$ of each i_x will be compared with $DisPV_t$ of each i_y using Vector based similarity. Before similarity

Figure 2. Example of DisPV and DesPV



computation, IM_PC generates a Virtual document of the set of $DisPV\left(i_x\right)$ for each i_x in I_s (i_y in I_t). Then, the terms in the virtual document of each instance are represented as a vector and their weights are assigned using TF-IDF method (Salton & McGill, 1986). The similarity between each instances pair; based on *DisPV*; is computed using COSINE distance between their virtual documents.

After similarity computation, the result is *AlignDP* a set of instance pairs that have $sim\left(i_x,i_y\right)$ exceeded a predefined threshold ^{3}. ^{3} is a similarity threshold denoting the minimum level of matching required for considering two instances as similar. The instances in *AlignDP* are considered partially similar and they will be more compared in the next stage.

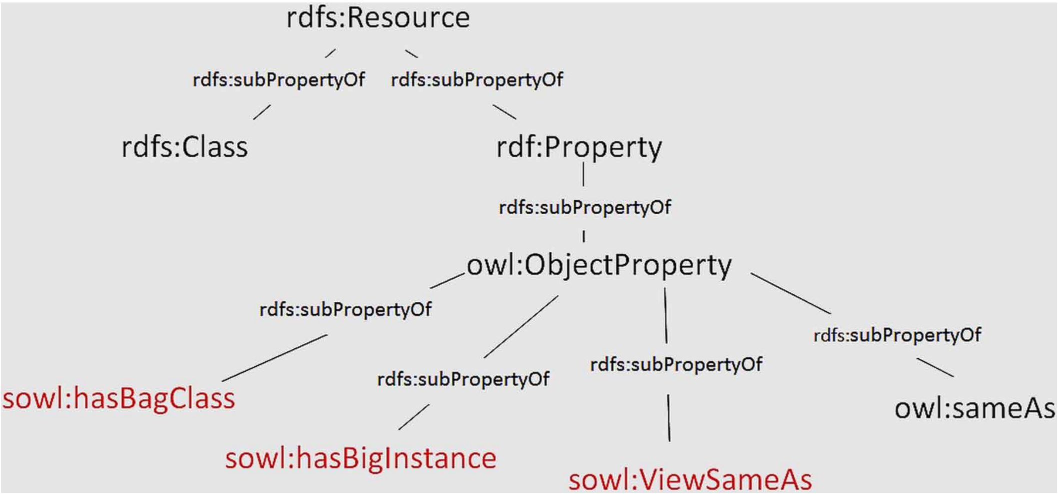
Result Refinement Based on DesPV

In this stage, the DesPV of instances in AlignDP are selected to complete the comparison process. A virtual document of DesPV is generated for each instance pair. The similarity computation of DesPV is done by the same method applied for DisPV. Instances pairs that have similarity value exceeded ^{3} are considered as similar ones. Then, an identity link *sameAs* is established between them.

In fact, uncompleted instances' information or a particular instances' description depending on a specific context or domain implies surely unlinked correspondences. For example; in Figure 2; the instances "sd:AhmedAli" and "td:AhmedAli" will not be linked even if they refer to the same person. Moreover, "sd:LIRE_Laboratory" and "td:Constantine2_University" will not be linked even if they refer to the same localization. The way in which we address these two challenges is by linking these instances through a novel link *ViewSameAs.*

Definition 7 (ViewSameAs): *The ViewSameAs link is defined to link partially similar instances. Given two instances i_x and i_y from I_s and I_t respectively. ViewSameAs is established depending on similar DisPV within instances. It is defined as:*

Figure 3. Sub-property relationship between OWL, RDFS and the proposed properties



$$\left(i_x, i_y, ViewSameAs\right) if\, and\, only\, if : \forall\ i_x \in I_s\ ,\ \forall\ i_y \in\ I_{t,}$$

$$DisPV\left(i_x\right) \equiv DisPV\left(i_y\right) \wedge DesPV\left(i_x\right) \not\equiv DesPV\left(i_y\right) \tag{15}$$

Each $i_x \in I_s$ *is matched to* $i_y \in I_t$ *by ViewSameAs if* $DisPV\left(i_x\right)$ *are similar to* $DisPV\left(i_y\right)$ *and* $DesPV\left(i_x\right)$ *are dissimilar to* $DesPV\left(i_y\right)$.

The novel construct ViewSameAs is formalized to represent a certain similarity degree. It is proposed principally to improve the IM task; i.e. it is helpful in discovering *sameAs* links. The *sowl:ViewSameAs* is a sub-property of *owl:ObjectProperty* where *sowl* is the ontology in which this kind of relationship is defined. The Figure 3 presents the sub-property relationship between the existing properties of OWL, RDFS and the proposed one.

The interdependence between the instances' information and the type of matching link in which *ViewSameAs* is introduced requires the adoption of the following SameAs definition.

Definition 8 (sameAs): *The identity link sameAs is defined to link similar instances. Given two instances* i_x *and* i_y *from* I_s *and* I_t *respectively, sameAs is established depending on the similar DisPV and DesPV within entities. It is defined as:*

$$\left(i_x, i_y, SameAs\right) if\, and\, only\, if :$$

$$\forall\ i_x \in I_s\ ,\ \forall i_y \in\ I_t \tag{16}$$

$$DisPV\left(i_x\right) \equiv DisPV\left(i_y\right) \wedge DesPV\left(i_x\right) \equiv DesPV\left(i_y\right)$$

Each $i_x \in I_s$ *is matched to* $i_y \in I_t$ *by the sameAs if* $DisPV\left(i_x\right)$ *and* $DisPV\left(i_y\right)$ *are similar to* $DesPV\left(i_x\right)$ $DesPV\left(i_y\right)$ *respectively.*

The output of this step is: *(i) AlignSA* including a set of quadruplet $(i_x, i_y, conf, SameAs\ )$ and *(ii) AlignVSA* including a set of quintuplet $(\ i_x, i_y, conf, ViewSameAs, vote\ ).vote$ refers to the number of common similar DesPV between $\left(i_x, i_y\right)$.

IM_VSA

The IM_VSA process is performed in three main steps: Detection of *ViewSameAs* links, Instances clustering and Replacing *ViewSameAs* by *sameAs*.

Detection of ViewSameAs Links

This step allows detecting instances, obtained by IM_PC, that are matched by *ViewSameAs* link in order to match them by *sameAs* link. Given a set of datasets **G** = { G_1, $G_2, \ldots, G_n$ } where: $\exists i_1 \in G_1$, ($i_1 i_2,$ ViewSameAs) $\wedge$ $\left(i_1, i_3,\ \text{ViewSameAs}\right) \ldots \left(\ i_1, i_n, \text{ViewSameAs}\right) |(i_1 \in G_1, i_2 \in G_2, \ldots, i_n \in G_n)$ and $n > 2$. For a given instance that has only a single *ViewSameAs* link with another instance, the possibility of being similar can't be treated without useful information from other instances.

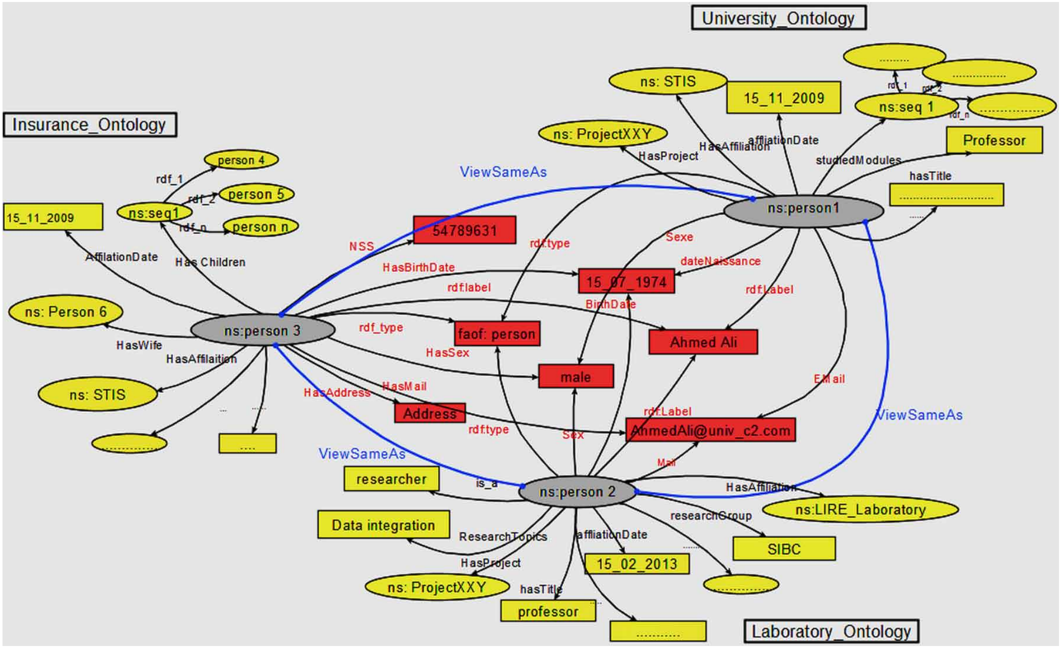
Let $(AlignVSA_1$, $AlignVSA_2, \ldots,$ $AlignVSA_n$) the set of partially similar instances of all possible alignments between { $G_1, G_2, \ldots, G_n$ }. For each $i \in AlignVSA$, IM_VSA detects the related *ViewSameAs* links where: $AlignVSA = \bigcup_{Y=1}^{n} AlignVSA_Y$.

Figure 4 illustrates an example of a person who is represented in different contexts where person1, person2 and person3 refer to the object "Ahmed Ali", if we suppose that $i_1 = \text{person1}$ so all the *ViewSameAs* links matched to this instance will be detected.

```
Algorithm 1. Instances' Detection
Input: AlignVSA.
Output: $AlignVSA'$
        $AlignVSA' \leftarrow \emptyset$, $AlignVSA'_i \emptyset$ , x = 0, D = false
        For each $i \in AlignVSA$
        D = $i$. Detect(ViewSameAs);
        If D = true then
        x= x+1
        $AlignVSA'_i = AlignVSA'_i \cup \left(i_x, i_y, conf, ViewSameAs, vote\right)$
        Else
        $x$
        If $x \geq 2$ then
          $AlignVSA' \leftarrow AlignVSA' \cup AlignVSA'_i$
          End if
Return $AlignVSA'$
```

The algorithm that allows *ViewSameAs* detection is given in Algorithm 1. Each *ViewSameAs* link that connects i_x to the set of instances ($i_{y1}, i_{y2}, \ldots, i_{yn}$) is detected. In Parallel, the number of the detected *ViewSameAs* links $\left(x\right)$ is calculated for each i_x. If x is equal or more than 2, $AlignVSA'_i$; the set of quintuplet $\left(i_x, i_y, conf, ViewSameAs, vote\right)$; is selected from $AlignVSA$ to $AlignVSA'$.

Figure 4. Example of instance with different descriptions



Instance Clustering

This step aims to group instances in $AlignVSA'$ in clusters. However, for each $i \in AlignVSA'$ a cluster $Cluster_i$ is constructed. The cluster of i_{x1} is represented as:

$$Cluster_{i_{x1}} \begin{pmatrix} i_{x1}, i_{y1}, conf_1, ViewSameAs, vote_1 \\ i_{x1}, i_{y2}, conf_2, ViewSameAs, vote_2 \\ \ldots.. \\ i_{x1}, i_{yn}, conf_i, ViewSameAs, vote_i \end{pmatrix} \tag{17}$$

Where: $i_{x1}, i_{y1..yn} \in AlignVSA'$ and $\left(i_{x1}, i_{y1..n}, conf_i, ViewSameAs, vote_i\right) \in AlignVSA'$

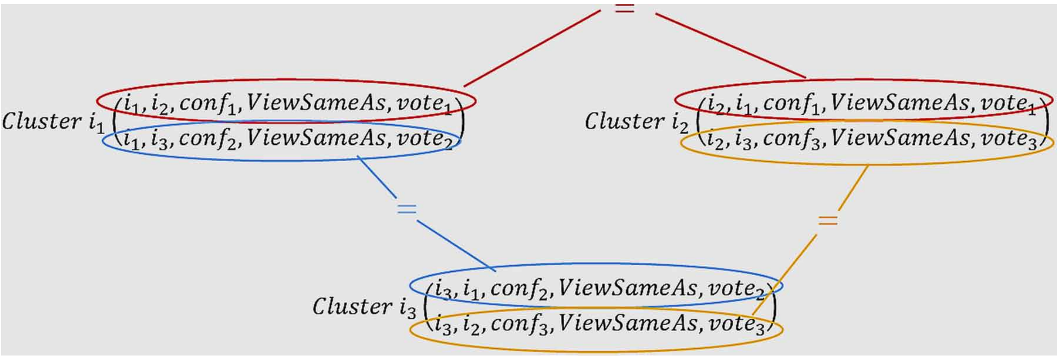
Based on the example depicted on figure 4, IM_VSA constructs three clusters:

$$Cluster_{person1} \begin{pmatrix} person1, person2, conf_1, ViewSameAs, 2 \\ person1, person3, conf_2, ViewSameAs, 2 \end{pmatrix}$$

$$Cluster_{person2} \begin{pmatrix} person2, person1, conf_1, ViewSameAs, 2 \\ person2, person3, conf_3, ViewSameAs, 0 \end{pmatrix}$$

$$Cluster_{person3} \begin{pmatrix} person3, person1, conf_2, ViewSameAs, 2 \\ person3, person2, conf_3, ViewSameAs, 0 \end{pmatrix}$$

Figure 5. Clustering duplication


=
Cluster i_1 $(i_1, i_2, conf_1, ViewSameAs, vote_1)$
$(i_1, i_3, conf_2, ViewSameAs, vote_2)$
Cluster i_2 $(i_2, i_1, conf_1, ViewSameAs, vote_1)$
$(i_2, i_3, conf_3, ViewSameAs, vote_3)$
=
=
Cluster i_3 $(i_3, i_1, conf_2, ViewSameAs, vote_2)$
$(i_3, i_2, conf_3, ViewSameAs, vote_3)$

Actually, the aim of clustering in this paper is to identify correspondences by using a set of partially similar instances; for example, the partially similar instances pair (*person1*, *person2)* is clustered with (*person1*, *person3)* in $Cluster_{person1}$ to treat the possibility of finding the correspondence of *person1* by using the information within *person2* and *person3*. However, the same instances set will be clustered several times in diverse clusters; depending on the number of connected instances; which produces duplication. In fact, these clusters group the same instances but with different parameters values.

Let $Cluster_{i1}$, $Cluster_{i2}$ *and* $Cluster_{i3}$ be three clusters as illustrated in Figure 5, we observe that they contain the same instances: i_1 , i_2 and i_3. Furthermore, each quintuplet $(i_x, i_y, conf_i, ViewSameAs, vote_i) \in Cluster_{i1}$ is also belonging to $Cluster_{i2}$ or $Cluster_{i3}$. To eliminate this duplication, IM_VSA introduces the notion of *dominant cluster.*

Definition 9 (Dominant cluster): *A dominant cluster is a group of partially similar instances with the highest number of vote. Let* $\left(Cluster_1, Cluster_2, \dots, Cluster_n\right)$ *the clusters that group the same set of instances* $\left(i_1 \dots i_n\right)$, $Cluster_x$ *is the dominant cluster if and only if:*

$$\forall Cluster_x, \nexists\ Cluster_y \in \left(Cluster_1, Cluster_2, \dots, Cluster_n\right) - Cluster_x \ ,\ \sum_{i=1}^{n} vote_{iy} > \sum_{i=1}^{n} vote_{ix}$$

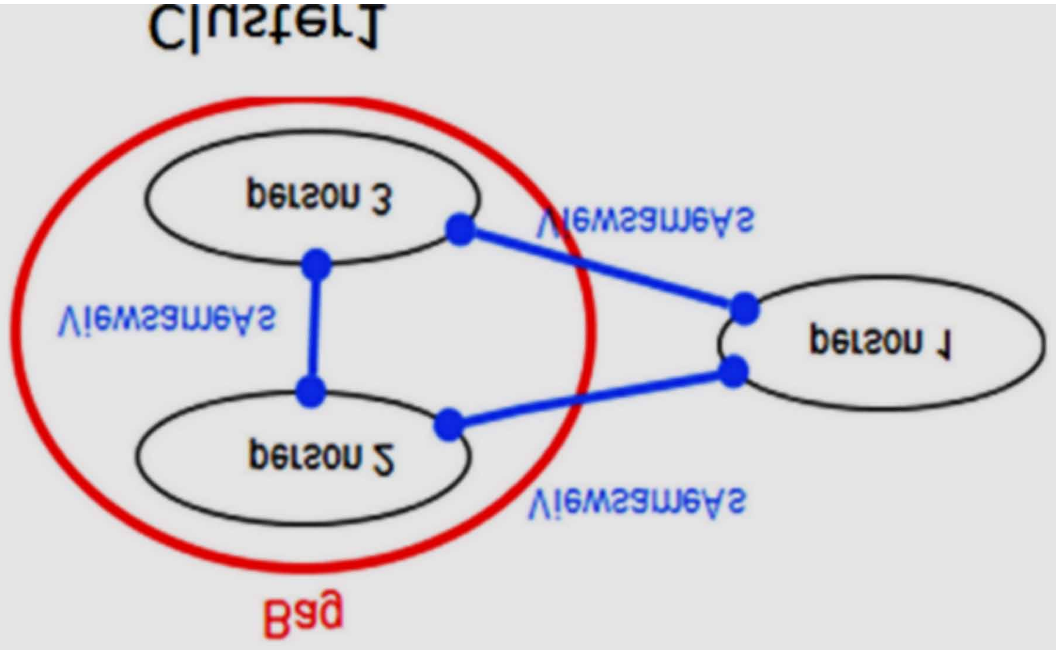
Based on the example presented in Figure 4, the dominant cluster is:

$$Cluster_{i1}\begin{pmatrix} person1\ , person2, conf_i, ViewSameAs, 2 \\ person1\ , person3, conf_i, ViewSameAs, 2 \end{pmatrix}$$

After instance clustering, a bag class is introduced using the variable $vote$. It is defined as:

Definition 10 (Bag Class): *A Bag Class is a set of partially similar instances, where every instances pair in the cluster has at least one DesPV in common. Let* $Cluster_{ix}$ *be a dominant cluster*

Figure 6. Instances bag of cluster1 where n=3



including $\{(i_x, i_{y1}) \ldots (i_x, i_{yn})\}$ *a set of instances pairs,* Bag_{i_x} *is a Bag class if* $\forall\ (i_x, i_{yn}) \in Cluster_{ix}$ *:* $vote_i \neq 0$*. A Bag is modelled as:*

$$Bag_{i_x} = \bigcup_{m=1}^{n} i_{ym}$$

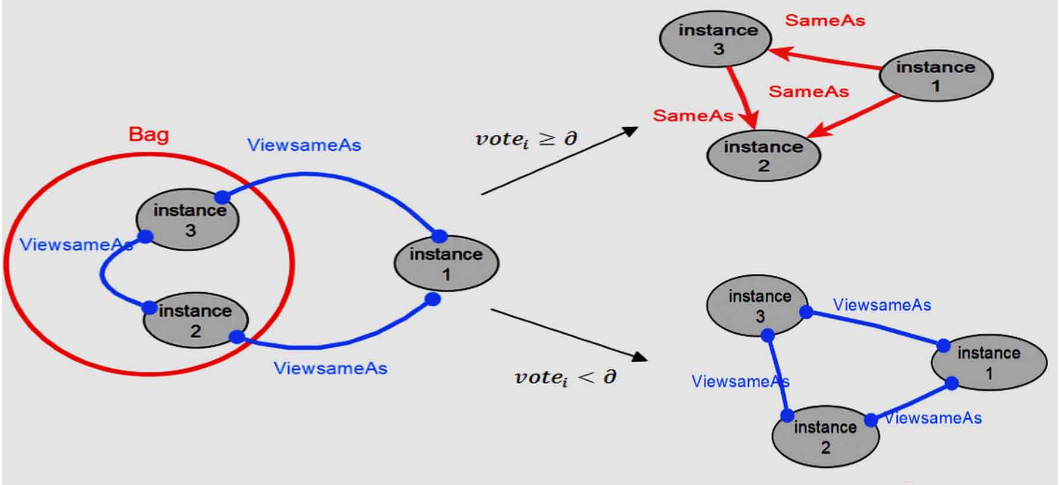
Note that i_x *is called the "Big instance".*

Definition 11 (Big Instance): *A big instance is the instance that has the highest number of vote with all or with the most instances of the same dominant cluster. Let* $Cluster_{ix}$ *be a dominant cluster including the set of instances* $(i_1 \ldots i_n)$. i_x *is the big instance if and only if:*

$$\forall i_x, \nexists i_y \in \left\{\left(i_1 \ldots i_n\right)\right\} - \left\{i_x\right\}, \sum_{i=1}^{n} vote_{y_i} > \sum_{i=1}^{n} vote_{x_i} \tag{20}$$

Using the current example and as illustrated in figure 6, $Cluster1$ is modeled as a *Bag Class,* where: $Bag_{person1} = \left\{person2, person3\right\}$ and *person1* is the big instance.

Figure 7. From ViewSameAs to sameAs



Replacing ViewSameAs Link By sameAs Link

In this step, IM_VSA produces *sameAs* link based on the *ViewSameAs* link as shown in Figure 7. The alignment is done using the following formula:

$$InstAlign\left(i_x, Bag_{i_x}\right) = \begin{cases} \left(i_x, i_{y1}, SameAs\right) \ldots and \left(i_x, i_{yn}, SameAs\right) if : \forall \ vote_i \geq \partial \\ \left(i_x, i_{y1}, ViewSameAs\right) \ldots \left(i_x, i_{yn}, ViewSameAs\right) if : \exists \ vote_i < \partial \ otherwise \end{cases}$$

A *ViewSameAs* link will be replaced by a *sameAs* link if the vote between the big instance i_x and each instance $(i_{y1}, i_{y2}, \ldots, i_{yn})$ in Bag_{i_x} exceeds ∂. Otherwise, *ViewSameAs* links between the instances in Bag_{i_x} and with i_x are conserved (as illustrated in Figure 7).

Let $Bag_{ix} = \left\{i_{y1} \ldots i_{yn}\right\}$ and i_x is the Big instance belong $Cluster_{i_x}$:

$$Cluster_{i_x} \begin{cases} (i_x, i_{y1}, conf_1, ViewSameAs, Vote_1) \\ \left(i_x, i_{y2}, conf_2, ViewSameAs, Vote_2\right) \\ \ldots \\ \left(i_x, i_{yn}, conf_n, ViewSameAs, Vote_n\right) \end{cases}$$

Case 1: $\forall vote_i \geq \partial\left(i = 1..n\right)$

$$\text{If}\begin{cases}Vote_1 \geq \partial \\ Vote_2 \geq \partial \\ \cdots \\ Vote_n \geq \partial\end{cases}\text{Then}\begin{cases}(i_x, i_{y1}, SameAs) \\ (i_x, i_{y2}, SameAs) \\ \cdots \\ (i_x, i_{yn}, SameAs)\end{cases}\text{And}\left\{\forall\ (i_x, i_{ym}), (i_x, i_{ym}, SameAs)\,/\,m = 1..n\ \right\} \tag{22}$$

In case1, each instances pair (i_x, i_y); will be connected with the *sameAs* instead of *ViewSameAs*.

Case 2: $\forall\, vote_i < \partial\left(i = 1..n\right)$

$$\text{If}\begin{cases}Vote_1 < \partial \\ Vote_2 < \partial \\ \cdots \\ Vote_n < \partial\end{cases}\text{Then}\begin{cases}(i_x, i_{y1}, conf_1, ViewSameAs, Vote_1) \\ \left(i_x, i_{y2}, conf_2, ViewSameAs, Vote_2\right) \\ \cdots \\ \left(i_x, i_{yn}, conf_n, ViewSameAs, Vote_n\right)\end{cases} \tag{23}$$

And $\left\{\forall\ (i_x, i_{ym}), (i_x, i_{ym}, ViewSameAs)\,/\,m = 1..n\ \right\}$

In case2, the *ViewSameAs* links in $Cluster_{i_x}$ are conserved, i.e. each instances pair (i_x, i_y) keeps the connection with *ViewSameAs.*

Case 3: $\exists vote_i < \partial\ and\ \geq \partial\ \ (i = 1..n\,)$

$$\text{If}\begin{cases}\exists vote_i < \partial \\ \quad and \\ \quad \geq \partial\end{cases}\text{Then}\begin{Bmatrix} Case1 where: \\ Bag'_{i_x} = \left\{i_{y1} \ldots i_{yn}\right\} - \{i_y, vote_{i_y} < \partial\}\end{Bmatrix} \tag{24}$$

In case 3, if one of the instances $i_y \in \left\{i_{y1} \ldots i_{yn}\right\}$ has insufficient number of vote with the big instance i_x, then it will be eliminated and we complete this case as case1.-

In the next section, we show how the Big instance and Bag class can improve the matching results.

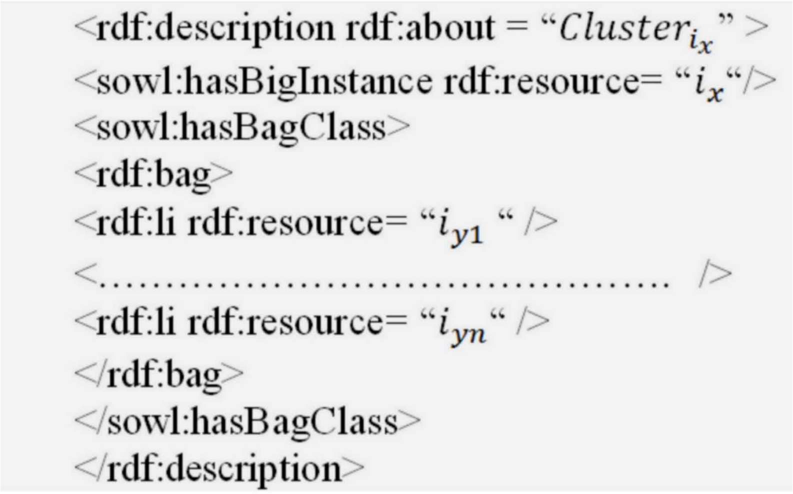
IM_AMD

The key idea behind the IM-AMD process is to benefit from the Mata-data about IM_VSA alignment by using the Big instance and the Bag class instances. To clarify, the creation of the metadata file is based on $AlignVSA$' where the input of IM_AMD is $AlignVSA$.

Alignment Meta-Data

This section presents the structure of the RDF metadata file (see Figure 8). IM-AMD allows generating more correspondences where the main motivation is the alignment reuse. To achieve this, we propose the use of two novel properties named: *sowl:hasBigInstance* and *sow:hasBagClass* which are the sub properties of *owl:ObjectProperty* (as illustrated in Figure 3). They are defined as:

Figure 8. Metadata structure



```
<rdf:description rdf:about = “Cluster_ix” >
<sowl:hasBigInstance rdf:resource= “i_x“/>
<sowl:hasBagClass>
<rdf:bag>
<rdf:li rdf:resource= “i_y1 “ />
<.......................................  />
<rdf:li rdf:resource= “i_yn“ />
</rdf:bag>
</sowl:hasBagClass>
</rdf:description>
```

Definition 12 (hasBigInstance): *The hasBigInstance is defined to link each dominant cluster with the related Big instance. Given* $Cluster_{i_x}$ *a dominant cluster including the set of instances (* $i_x, i_{y1}, \ldots, i_{yn}$ *):*

$$(Cluster_{i_x} \quad sowl:hasBigInstance \quad i_x \quad) \tag{25}$$

Definition 13 (hasBagClass): *The hasBagClass is defined to link each dominant cluster with the related Bag Class. Given the dominant Cluster* $Cluster_{i_x}$ *including the set of instances (* $i_x, i_{y1}, \ldots, i_{yn}$ *):*

$$(Cluster_{i_x} \quad sowl:hasBagClass \quad Bag_{i_x}) \tag{26}$$

$$where:\ i_{y1}, \ldots, i_{yn} \in Bag_{i_x}.$$

The IM_AMD process is performed in three main steps: Big instance selection, Bag class verification and producing SameAs links. In the next subsections, we present these steps with an illustrated example.

Big Instance Selection

For each $i_x \in AlignVSA$; IM_AMD verifies if it is a big instance or not. If i_x is a Big instance in one of the clusters of the metadata file, the related target instance (s) will be then selected for the next steps. The choice of the Big instances is related to the vote shared with the instances in the Bag class. This characteristic is helpful in detecting other correspondences without passing by the IM_VSA process. In the example depicted in the Figure 9, the instance i_3 is a Big instance that belongs to the dominant cluster *Cluster2.*

Bag Class Verification

In this step, IM_AMD selects from *AlignVSA*; for each Big instance i_x ; the target instance i_{y1} . If this last is matched to an instance i_{y2} which belongs to Bag_{i_x} , then it is possible to link these three instances (i_x, i_{y1}, i_{y2}) by *sameAs* link.

Based on the previous example, the big instance i_3 is matched to two instances i_2 and i_5 :

- i_2 does not related to an instance that belongs to the Bag class of $i_3 \rightarrow i_2$ will be ignored.
- i_9 is related to an instance i_5 but it does not belong to the Bag class of $i_3 \rightarrow i_9$ will be ignored.
- i_5 is related to an instance i_{10} that is belongs to Bag class of $i_3 \rightarrow i_5$ will be selected to the next step.

Producing sameAs Link

In IM_VSA, the replacement of the *ViewSameAs* link by *sameAs* link depends on the *Vote* parameter. With the same method, IM_AMD verifies the vote between the instances pairs $\left(i_x, i_{y1}\right)$ and $\left(i_{y1}, i_{y2}\right)$. If $Vote \geq \partial$, then:

$\left(i_x \quad sowl : ViewSameAs \quad i_{y1}\right)$ becomes $\left(i_x \quad owl : sameAs \quad i_{y1}\right)$.
$\left(i_{y1} \quad sowl : ViewSameAs \quad i_{y2}\right)$ becomes $\left(i_{y1} \quad owl : sameAs \quad i_{y2}\right)$.

These results are saved on *AlignAMD* as follow:

$$(i_x, i_{y1}, conf_1, sameAs)$$
$$(i_x, i_{y2}, conf_2, sameAs)$$

Based on the previous example, the results in the metadata file are illustrated in Figure 10 while those in *AlignAMD* are presented as follow:

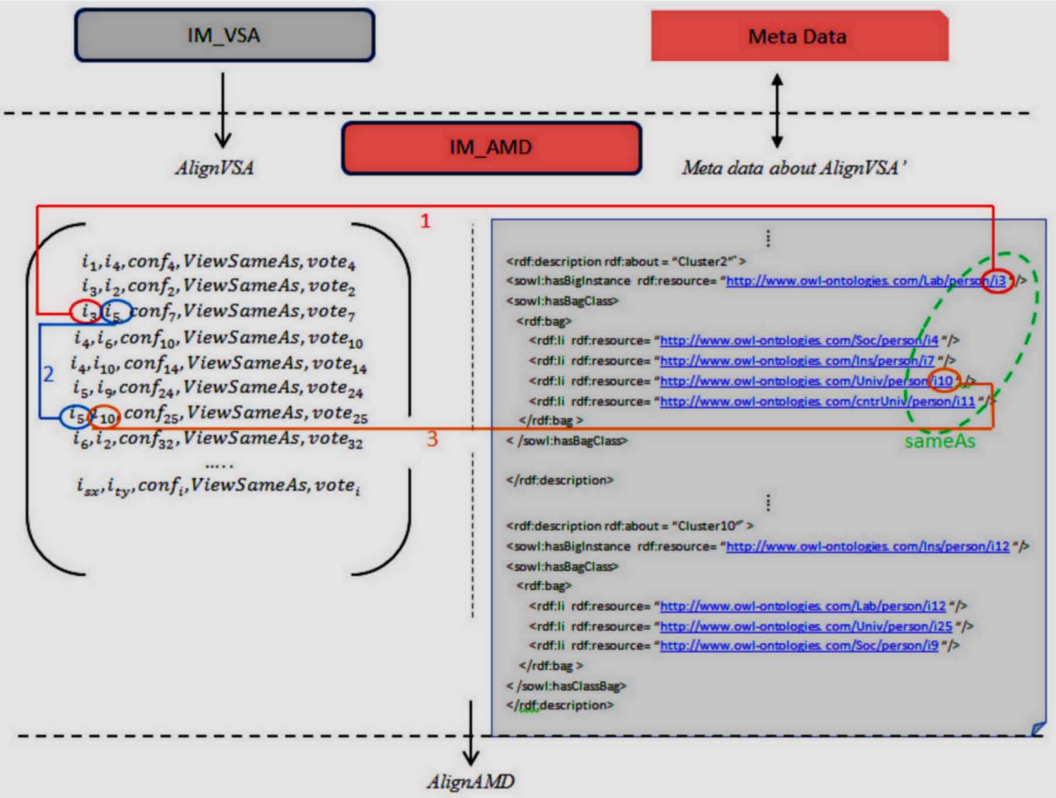
$$(i_3, i_5, conf_1, sameAs)$$
$$(i_5, i_{10'}, conf_2, sameAs)$$

EVALUATION

To validate the effectiveness of the proposed approach, several tests are conducted on both OAEI IM benchmarks and our datasets.

Datasets: OAEI[3] is an international ontology matching campaign that provides authoritative and reliable tests for ontology matching technologies and tools. Here, we use A-R-S and PR benchmarks OAEI 2009[4] and 2010[5] respectively to test IM_PC process. These datasets cannot be used for the experimentations of IM_VSA and IM-AMD processes, since these datasets didn't address the problem of multiple instances' descriptions. We decide therefore to use our own datasets.

Figure 9. An illustrated example of IM_AMD execution

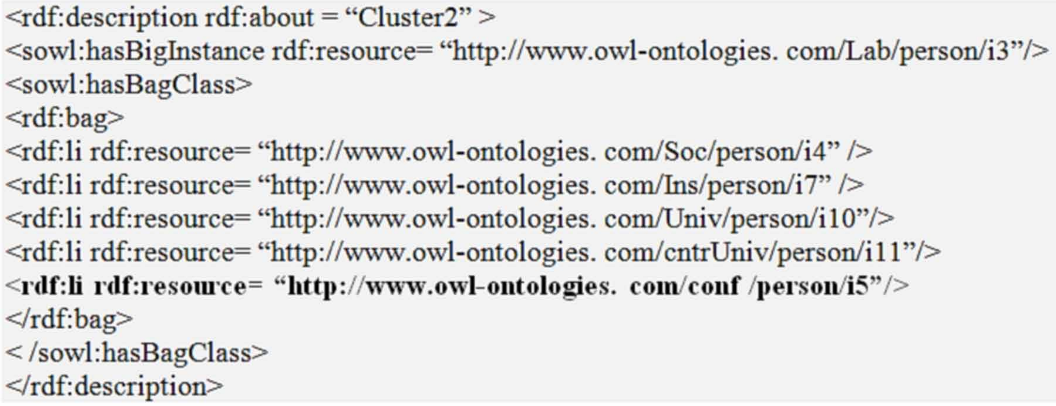

IM_VSA
Meta Data
IM_AMD
AlignVSA
Meta data about AlignVSA'
1
2
3
$i_1, i_4, conf_4, ViewSameAs, vote_4$
$i_3, i_2, conf_2, ViewSameAs, vote_2$
$i_3, i_5, conf_7, ViewSameAs, vote_7$
$i_4, i_6, conf_{10}, ViewSameAs, vote_{10}$
$i_4, i_{10}, conf_{14}, ViewSameAs, vote_{14}$
$i_5, i_9, conf_{24}, ViewSameAs, vote_{24}$
$i_5, i_{10}, conf_{25}, ViewSameAs, vote_{25}$
$i_6, i_2, conf_{32}, ViewSameAs, vote_{32}$
.....
$i_{sx}, i_{ty}, conf_i, ViewSameAs, vote_i$
sameAs
AlignAMD

```
                                   ⋮
<rdf:description rdf:about = "Cluster2"" >
<sowl:hasBigInstance  rdf:resource= "http://www.owl-ontologies. com/Lab/person/i3 "/>
<sowl:hasBagClass>
  <rdf:bag>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Soc/person/i4 "/>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Ins/person/i7 "/>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Univ/person/i10 "/>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/cntrUniv/person/i11 "/>
   </rdf:bag >
< /sowl:hasBagClass>

</rdf:description>
                                   ⋮
<rdf:description rdf:about = "Cluster10"" >
<sowl:hasBigInstance  rdf:resource= "http://www.owl-ontologies. com/Ins/person/i12 "/>
<sowl:hasBagClass>
  <rdf:bag>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Lab/person/i12 "/>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Univ/person/i25 "/>
    <rdf:li  rdf:resource= "http://www.owl-ontologies. com/Soc/person/i9 "/>
   </rdf:bag >
< /sowl:hasClassBag>
</rdf:description>
```

The A-R-S benchmark contains three datasets named dblp, rexa and eprints covering scientific publication field. PR benchmark is composed of three small datasets Person1, Person2 and Restaurant (here we use only Person' datasets). Each of them has two set of instances (Person11, Person12) and (Person21, Person22). Laboratory_onto (Lab), University_onto (Univ), Insurance_onto (Ins) and

Figure 10. Example of alignment in IM_AMD



```
<rdf:description rdf:about = "Cluster2" >
<sowl:hasBigInstance rdf:resource= "http://www.owl-ontologies. com/Lab/person/i3"/>
<sowl:hasBagClass>
<rdf:bag>
<rdf:li rdf:resource= "http://www.owl-ontologies. com/Soc/person/i4" />
<rdf:li rdf:resource= "http://www.owl-ontologies. com/Ins/person/i7" />
<rdf:li rdf:resource= "http://www.owl-ontologies. com/Univ/person/i10"/>
<rdf:li rdf:resource= "http://www.owl-ontologies. com/cntrUniv/person/i11"/>
<rdf:li rdf:resource= "http://www.owl-ontologies. com/conf /person/i5"/>
</rdf:bag>
< /sowl:hasBagClass>
</rdf:description>
```

Table 2. The size of dataset (# I denotes the number of instances)

A-R-S	#I	PR	#I	Datasets	#I
Dblp	1642945	Person1	~ 500	Laboratory_onto	350
Rexa	14771	Person2	~ 600	University_onto	5612
Eprints	847	-	-	Insurance_onto	4789
				Social_onto	9458

Social_onto (Soc) are four datasets including people holding different descriptions. The number of instances contained in each dataset is showing in Table 2.

Performance metrics: We use the three standard retrieval information metrics: precision, recall and F1-Measure where:

$$Precision\left(P\right) = \frac{\#correctly\_discovered\_results}{\#discovered\_results} \tag{27}$$

$$Recall\left(R\right) = \frac{\#correctly\_discovered\_results}{\#Correctly\_results} \tag{28}$$

$$\text{F1} - \text{Measure}\left(F1\right) = \frac{2 * P * R}{P + R} \tag{29}$$

Parameters: γ and ∂ are the two parameters used in our approach. In a first test (as illustrated in Figure 11), we check the performance of IM_PC with different γ on Eprints and Rexa datasets. In fact, the more γ is low, chances to find more correspondences are high (the recall R gets to its peaks), but the execution time increases. In the performance evaluation, it is preferable to take into account the two measurements P and R via F. So, when $\gamma = 0.7$, the F1 measure gets to its peaks. Therefore, we complete our tests with $\gamma = 0.7$ as a default value.

In the second test, we put $\gamma = 0.7$ and we check the performance of IM_VSA with different values of ∂ on the three datasets Lab, Univ and Ins. Indeed, the value of the parameter ∂ has also an effect in improving results.

- When we choose $\partial \leq 1$, a significant number of correct alignments is obtained but it is accompanied with many false results. The reason is that in some cases one property value in common is insufficient in the matching task.
- When $\partial \geq 3$, many correct alignments were eliminated.
- When $\partial = 2$, IM_VSA achieved better result (best F1 measure) in which it recuperated the eliminated alignments.

However, the IM_VSA proves the effect of proposed technique that based on the *ViewSameAs* link in improving matching results. The figure 12(a), (b) and (c) presents the effects of ∂ on (Univ-Lab), (Ins-Lab) and (Univ-Ins) respectively.

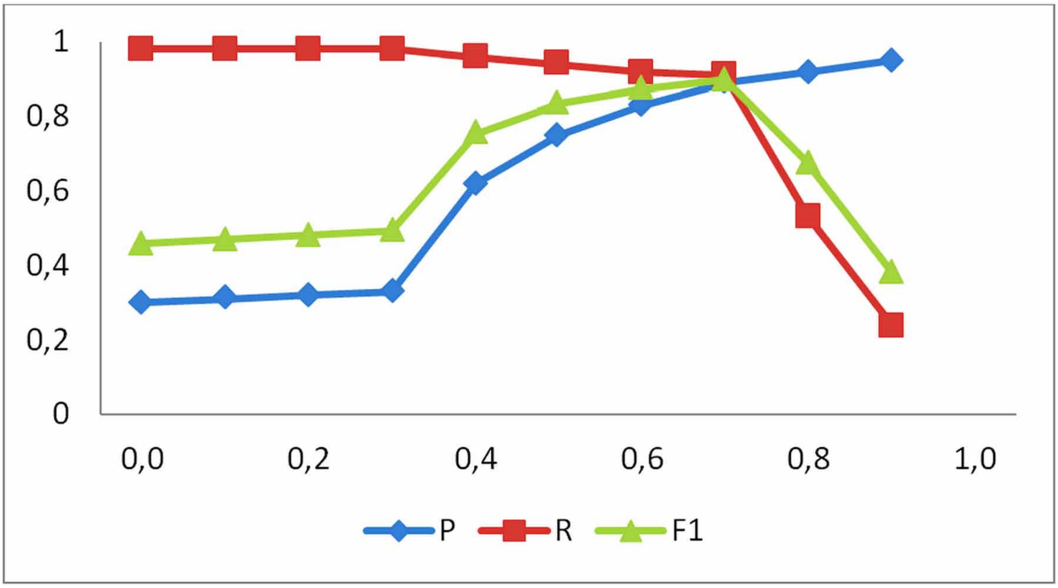
In this paper, we compare the IM_PC results with some existing systems. Then, we evaluate the performance of IM_VSA and IM_AMD processes. The results are illustrated and discussed in the following sub sections.

Evaluation of IM_PC With Existing Systems

In this test, we evaluate IM_PC process by using A-R-S and PR benchmarks:

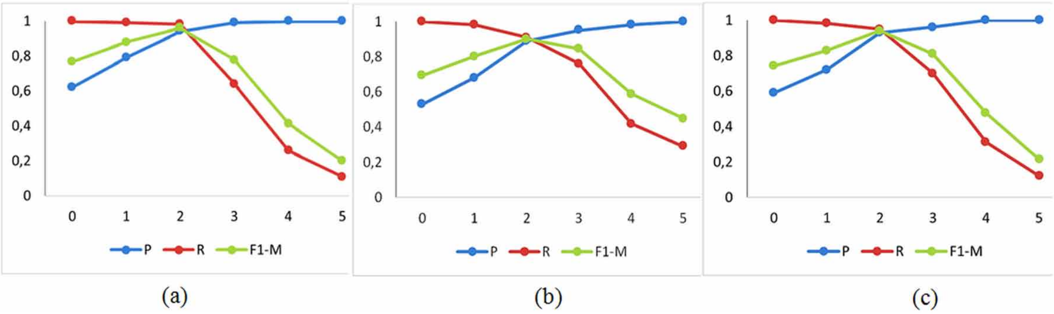
1) On A-R-S benchmark:
   - IM-PC gets good results compared with the other systems according to the precision metric as showing in Figure 13 (third higher P on rexa-dblp and dblp-eprints).

Figure 11. The effect of the threshold γ on Eprints-Rexa by IM-PC.



   - By comparing the Recall results illustrated in Figure 14, IM_PC gets the best Recall on rexa-dblp, eprints-rexa and dblp-eprints
   - In Figure 15, the results show that IM_PC generates the best F1 score on rexa-dblp and eprints-rexa, it gets the second higher F1 on dblp-eprints.
2) On PR benchmarks, it gets also good results as RIMOM and SERIMI:
   - On Person11-Person12, it gets the higher precision, recall and F1 like RIMOM and SERIMI (as presented in Figures 16, 17 and 18 respectively).
   - On Person 21-person 22, it gets the second higher precision after RIMOM as presented in Figure 16.
   - On Person 21-person 22, it gets the second higher recall while RIMOM has always the best results as showing in Figure 17.
   - On person 21-person 22, it gets the second higher F1 score after RIMOM as illustrated in the Figure 18.

Figure 12. The effect of $vote$ in IM_VSA on: (a) Univ-Lab, (b) Ins-Lab and (c) Univ-Ins



To conclude, IM_PC generates results as good as VMI and RIMOM on OAEI 2009 Benchmarks. On rexa-dblp and eprints-rexa, it gets the best recall and F1-score. In PR benchmark, IM_PC generates the same results as SERIMI and RIMOM on Person11-Person12. On Person21-Person22, it gets best results than SERIMI.

After analyzing results, we can say that:

- The good selection of discriminative information implies a good matching result and vice versa.
- Moreover, the existence of isolated instances without any descriptions or information has a negative effect on the matching process in which false alignments are produced (The results of IM systems are obtained from (Araujo et al., 2011, Li et al., 2013)).

Evaluation of IM_VSA

In this test, we evaluate the performance of IM_VSA with IM_PC to prove the efficiency of the proposed technique which is based on the *ViewSameAs* link. This test is realized on Lab-Univ, Ins-Lab and Univ-Ins. The results are illustrated in Figure 19, 20 and 21 respectively. The IM_VSA gets always the best results (Precision, Recall and F1-measure) than IM_PC which confirms the efficiency of the ViewSameAs-based clustering method in our proposal.

In other test and in order to prove the efficiency of the *ViewSameAs* link in the case of updating datasets, we add new information to Lab, Univ and Ins datasets. Then, we re-process IM_PC in two different ways: firstly, we re-start it from the beginning i.e. we compare all existing instance pairs within additional information (in this case IM_PC repeats only the comparison of the instances that will updated with the datasets instances). Secondly, we detect only instances matched by *ViewSameAs* to compare them with the updating information.

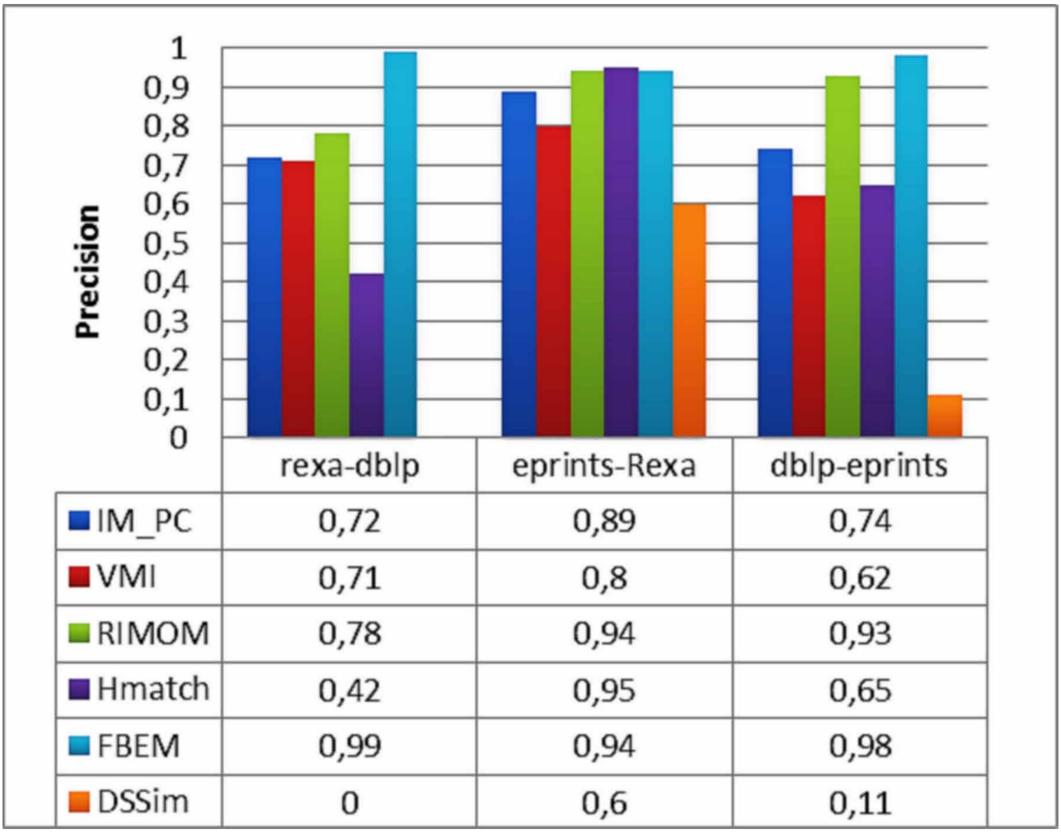
As showing in Table 3, the second method performs much better and faster than the first one. This improvement comes from the using of the first alignment results that we saved by the *ViewSameAs* link. Here, some false alignments were produced. This is due to the type of the additional information.

Moreover, we validate our proposal in number of extracted *sameAs* links. Table 4 presents the number of discovered *sameAs* links by using and without using the *ViewSameAs* link. Between the three datasets (Univ_Lab_Ins), IM_VSA produces 315 more correct links compared to IM_PC.

Evaluation of IM_AMD

In this test, we evaluate the performance of IM_AMD process. For this, we first compare the dataset pairs (Soc_Univ) and (Soc_Lab) using IM_PC and IM_VSA processes. Then, we try to detect correspondences between (Soc_Ins) by using the metadata file.

Figure 13. Comparison of the precision metric on A-R-S Benchmark



	rexa-dblp	eprints-Rexa	dblp-eprints
🟦 IM_PC	0,72	0,89	0,74
🟥 VMI	0,71	0,8	0,62
🟩 RIMOM	0,78	0,94	0,93
🟪 Hmatch	0,42	0,95	0,65
🟦 FBEM	0,99	0,94	0,98
🟧 DSSim	0	0,6	0,11

IM_AMD performs very well (P = 0.85, R = 0.91, F1 =0.88) and the more important very fast. This is due to the important information within the Big instances and the Bag instances which proved their effective in discovering correspondences.

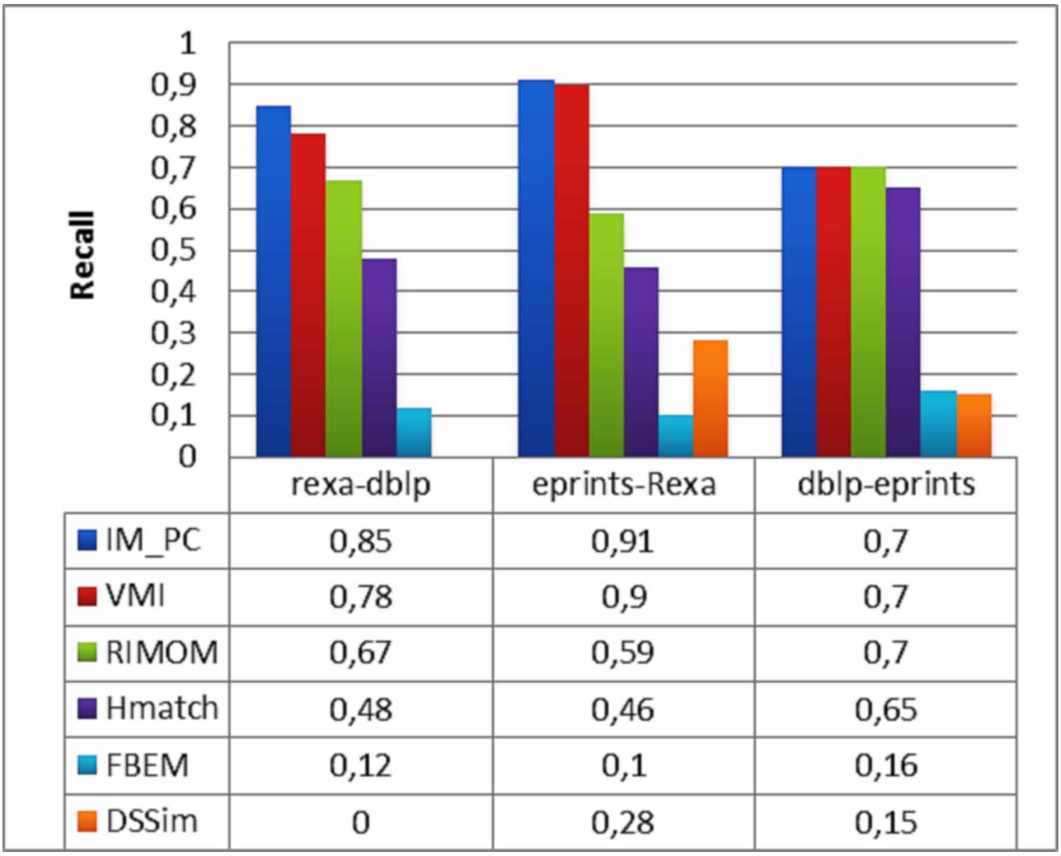
After analyzing the results, we observed that in some cases we can insert an instance in the Bag class while it should be represented as Big instance. Moreover, some alignment duplications are produced between the IM_VSA and IM_AMD processes. Therefore, we decide to add in our future works:

1. A component allowing the verification of the metadata file.
2. An intermediate component between the three proposed processes.

In summary, we have the following conclusion of our proposal:

1. Our approach explores the semantic information within instances effectively in identifying correspondences.
2. We have demonstrated the efficiency of the proposed methods: ViewSameAs-based clustering and the metadata file in discovering correspondences.
3. Using OAEI benchmarks, IM_PC achieves good results compared to the existing systems. It gets the best recall and F1-score on rexa-dblp and eprints-Rexa. It gets also good results on the Person11-Person12 datasets.

Figure 14. Comparison of the Recall metric on A-R-S Benchmark



	rexa-dblp	eprints-Rexa	dblp-eprints
IM_PC	0,85	0,91	0,7
VMI	0,78	0,9	0,7
RIMOM	0,67	0,59	0,7
Hmatch	0,48	0,46	0,65
FBEM	0,12	0,1	0,16
DSSim	0	0,28	0,15

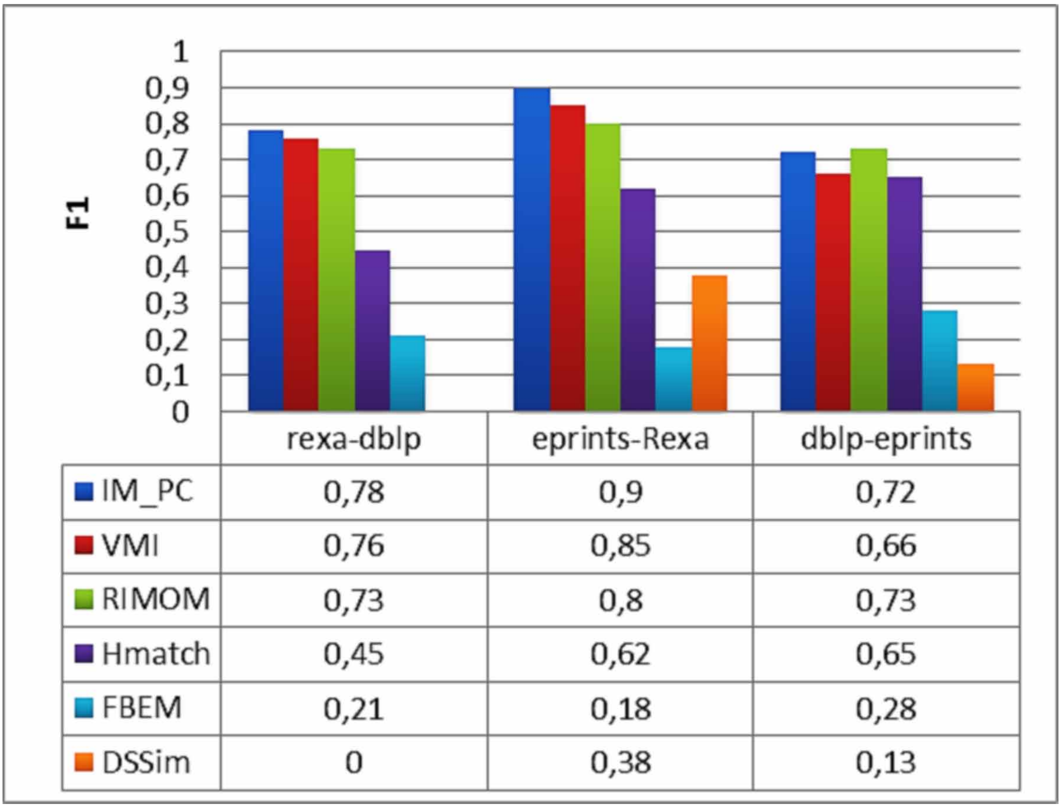
4. The proposed link *ViewSameAs* plays an important role in finding correspondences by keeping the track of the partially similar instances. It shortens the running time in updating datasets.
5. The results show that the matching task with *ViewSameAs* achieves better result than without it.
6. The proposed predicates *hasBigInstance* and *hasBagClass*; allowing saving helpful information; proved their effective in discovering correspondences.

RELATED WORK

Several IM approaches have been proposed over the last decade. Here, we refer to works mostly related to our study.

- ASL (Nguyen & Ichise, 2018): is a schema-independent system that performs IM on repositories without prior knowledge about their schemata. It bases on three main steps. At first, it allows the detection of property mappings to eliminate dissimilar instances. Then, it uses a token-based blocking procedure that discards dissimilar candidate pairs. Finally, it verifies the remaining pairs by estimating their similarity.
- AIM-PC (Lu et al, 2018), is a novel approach which can bring the human into the loop of IM. For candidate selection, it uses existing blocking techniques. For results refine, the authors propose to use a set of pairwise constraints and active learning.
- VMI (Li et al., 2013) is an IM approach proposed to handle the large scale ontology. It classifies the instances information in six categories: URI, name, meta, descriptive property values,

Figure 15.Comparison of the F1 metric on A-R-S Benchmark



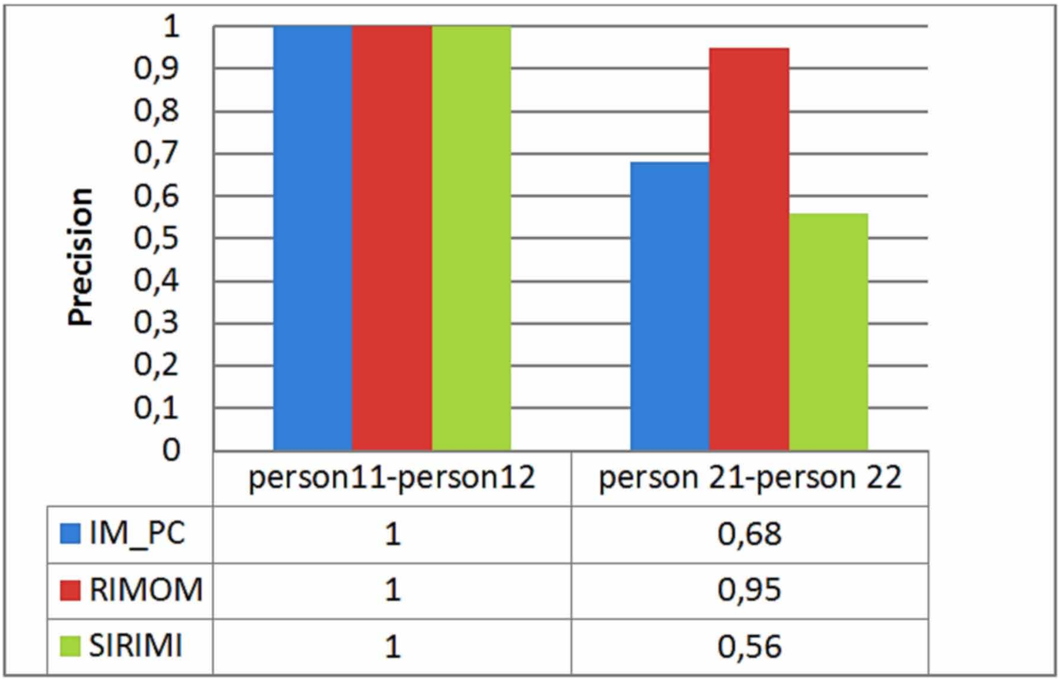
	rexa-dblp	eprints-Rexa	dblp-eprints
🟦 IM_PC	0,78	0,9	0,72
🟥 VMI	0,76	0,85	0,66
🟩 RIMOM	0,73	0,8	0,73
🟪 Hmatch	0,45	0,62	0,65
🟦 FBEM	0,21	0,18	0,28
🟧 DSSim	0	0,38	0,13

discriminative property values, and neighbors. For each instance, it generates two types of vector: name vector including name information and virtual document containing the descriptive and neighboring information. Then, the primary matching candidates are generated by applying some selection rules based on the vectors and indexes. To refine the results, VMI uses the discriminative property values.

◦ In (Wang et al., 2013), authors classify the instances information in lexical information, including: label, comments and data-type property values, and structural information hidden in links between entities (concepts and properties) and object-type property values. To find matching candidates, this approach uses the lexical information. It applies three matching strategies on this information including: Named-based strategy, Meta-based strategy and Instance-based strategy for schema matching; and Property-based strategy for IM task. Then, a voted-based method is employed to combine the results of these strategies. The additional correspondences are identified using the structural information.

◦ RIMOM (Li et al., 2009; Tang et al., 2006) is a dynamic multistrategy ontology alignment framework. For IM, a new framework is integrated to RIMOM called RIMOM-IM (Shao et al., 2016). It bases on two main instances' characteristics in its matching steps. For candidates' selection, it utilizes a blocking technique based on predicates and their distinctive object features. It applies three matching strategies: unique subject matching, one-left object matching and Score Matching which utilize the aligned instances for matching the remaining ones.

Figure 16. Comparison of the precision metric on PR Benchmark

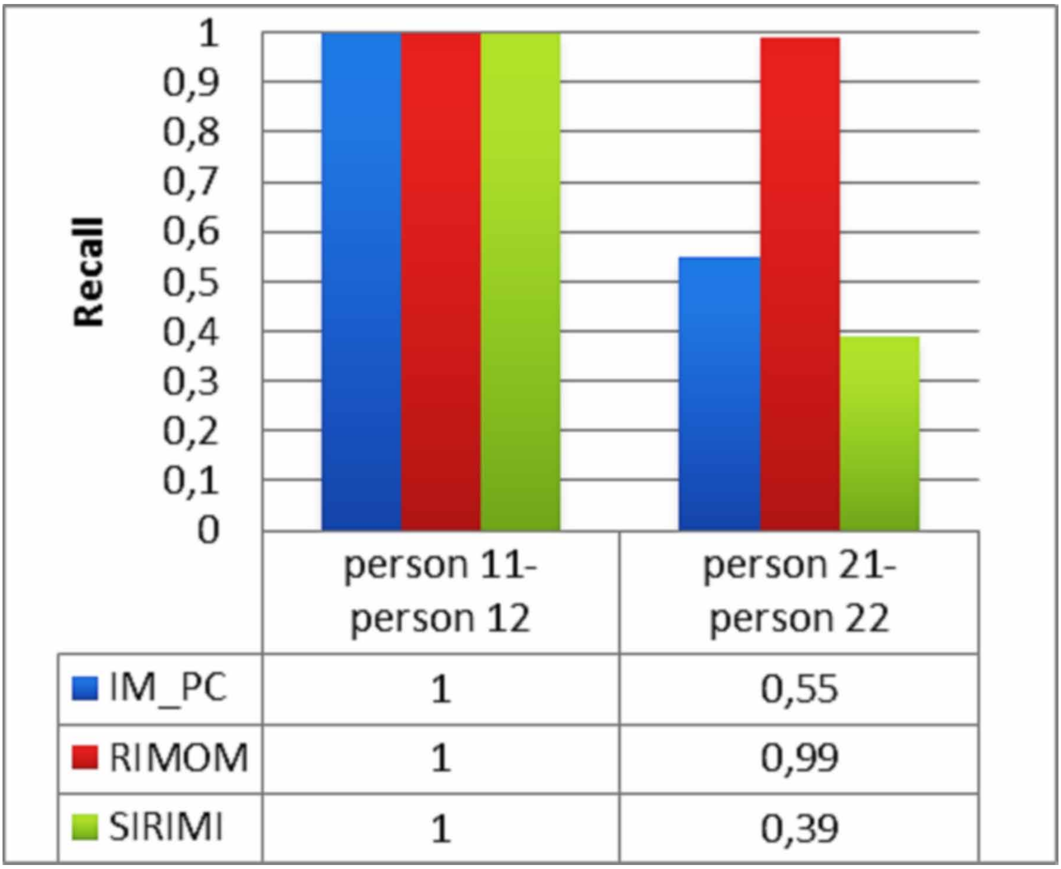


	person11-person12	person 21-person 22
🟦 IM_PC	1	0,68
🟥 RIMOM	1	0,95
🟩 SIRIMI	1	0,56

- SERIMI (Araujo et al., 2011; 2015) is a matching approach in which a new paradigm called class-based matching is proposed. This latter is combined with a direct matching to infer *sameAs* relation over heterogeneous data. It constructs boolean queries using tokens extracted from instances labels to select the matching candidates. In SERIMI, the paradigm of class-based matching is proposed to refine the candidates' selection by eliminating those that do not belong to the class of interest. However, using labels to generate matching candidates may produce many incorrect candidates and also may filter out many correct ones.
- FBEM (Stoermer & Rassadko, 2009) is a feature based instance matching system. It first computes the Levenstein similarity between all the features of instances, and then calculates the combined similarity score by summing all the maximum similarity feature combinations. FBEM also implemented a ''brute-force'' matching, similarity to get the matching results.
- DSSim (Nagy et al, 2008) is an ontology mapping system which incorporates the Dempster Shafer theory of evidence into the mapping process. It assesses similarity of all the entities from two ontologies. Therefore, it employs a multi-agent architecture to enable distributed execution of the approach.
- HMatch (Castano et al., 2008) is an ontology matching suite that provides a component for IM task called HMatch(*I*). The matching task in HMatch(*I*) is based on the comparison of instance properties also called roles and corresponding property values called role fillers. For computing similarity, HMatch(*I*) proposes two main functions: *Instance affinity* and *Filler similarity*. The former is calculated by taking into account all the properties; of each instance; together with their corresponding property fillers, and the latter is defined in order to adopt the matching technique more suitable for a given pair of fillers.

Compared to the approaches studied above, the proposed matching process starts by the discriminative property values contrary to VMI that uses descriptive ones. Discriminative properties are more relevant than descriptive ones. They are used to (i) quickly find matching candidates and (ii)

Figure 17. Comparison of the Recall metric on PR Benchmark



	person 11-person 12	person 21-person 22
🟦 IM_PC	1	0,55
🟥 RIMOM	1	0,99
🟩 SIRIMI	1	0,39

achieve the matching results (high recall and precision). In a second step, we use descriptive property values to refine the results. The classification of these properties is based on data semantic. It is more efficient than data structure classification as used in Z.Wang et al approach. As RIMOM_IM, we introduce also novel methods allowing the alignment reuse.

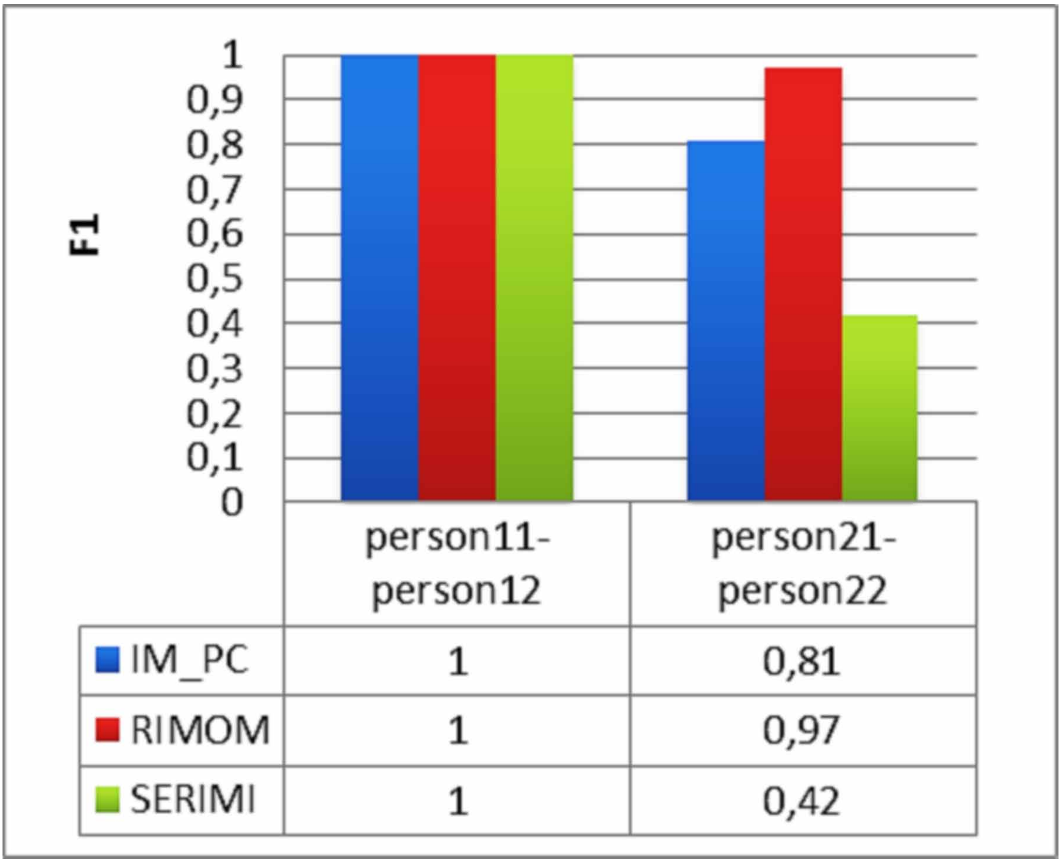
In Table 5, a comparative study of the above approaches and the proposed one is provided. This study is based on six criteria: Category, Candidate selection, result refinement, similarity methods, final link and alignment reusing.

CONCLUSION AND FUTURE WORK

In this paper, we discussed the IM problem which is considered as one of the main challenges in data integration field. We have proposed a novel IM approach based on the following characteristics:

- Classification of instances properties.
- Proposition of ViewSameAs link to connect partially similar instances.
- Detection of correspondences between instances having diverse descriptions.
- Alignment reuse via ViewSameAs-based clustering method and metadata.

Figure 18. Comparison of the F1 metric on PR Benchmark

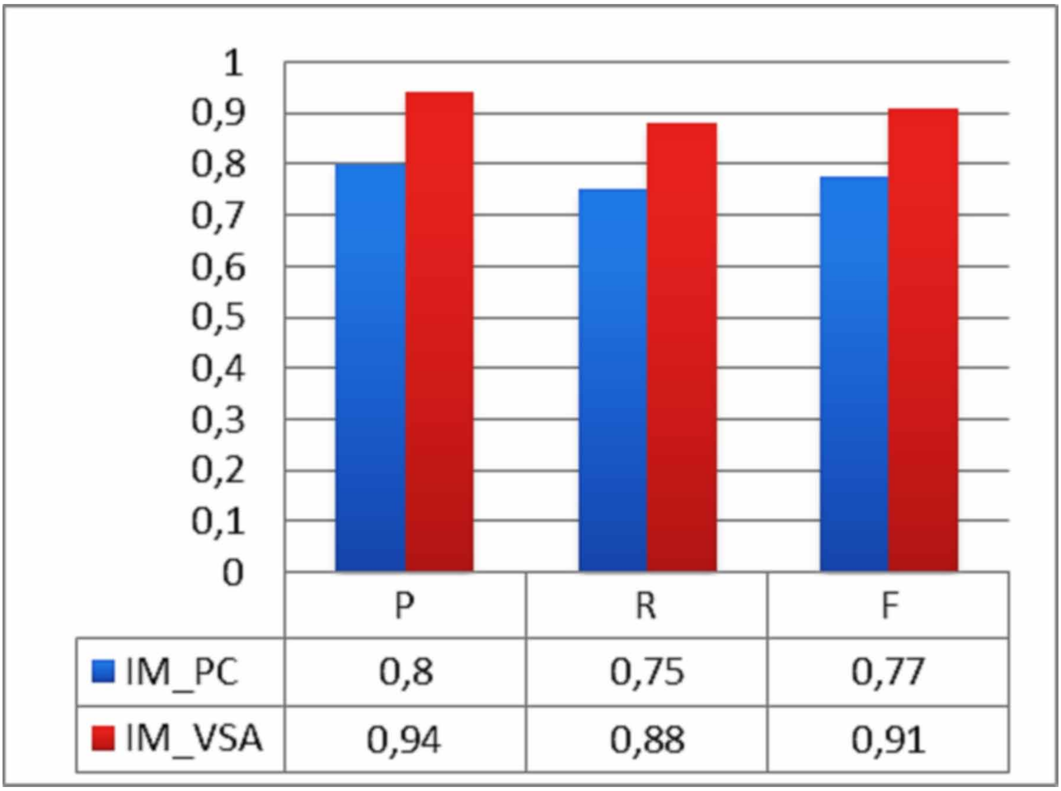


	person11-person12	person21-person22
🟦 IM_PC	1	0,81
🟥 RIMOM	1	0,97
🟩 SERIMI	1	0,42

The proposed approach includes three main processes IM_PC, IM_VSA and IM_AMD. The IM_PC uses the discriminative information to reduce the number of comparison pairs and find the matching candidates, while the descriptive one is used for result refinement. The result of this process is a set of instances matched by two links: *owl:sameAs* or *sowl:ViewSameAs*. The *owl:sameAs* link is utilized to match the similar instances while the proposed one is introduced to match the partially similar instances. In the second process IM_VSA, a novel clustering method is proposed aiming to detect correspondences between partially similar instances basing on the *ViewSameAs* link. In parallel, we propose to use two novel predicates *sowl:hasBigInstance* and *sowl:hasBagClass* as metadata. In the third process IM_AMD, we use this metadata for detecting corresponding instances. Compared to existing systems, the proposed one achieves a good result. The IM task; in which the *ViewSameAs* link is used, performs better than without it. Moreover, the metadata file is very helpful and useful in detecting correspondences. As presented in the motivating example, our approach could be suitable for geographical and bibliographical applications.

For future research, we propose the investigation of adopting an automatic method that aims at configuring the vote parameter to enhance the effectiveness of our proposal. Additionally, researchers could improve the efficiency of the IM_AMD process by adding new components that allow the verification of the metadata file and the synchronization of the matching processes. On the other hand, we would like to formalize our approach using FCA (Formal Concept Analysis). Moreover, we propose to create a novel ontology by integrating our ontology (*sowl*) with existing vocabularies

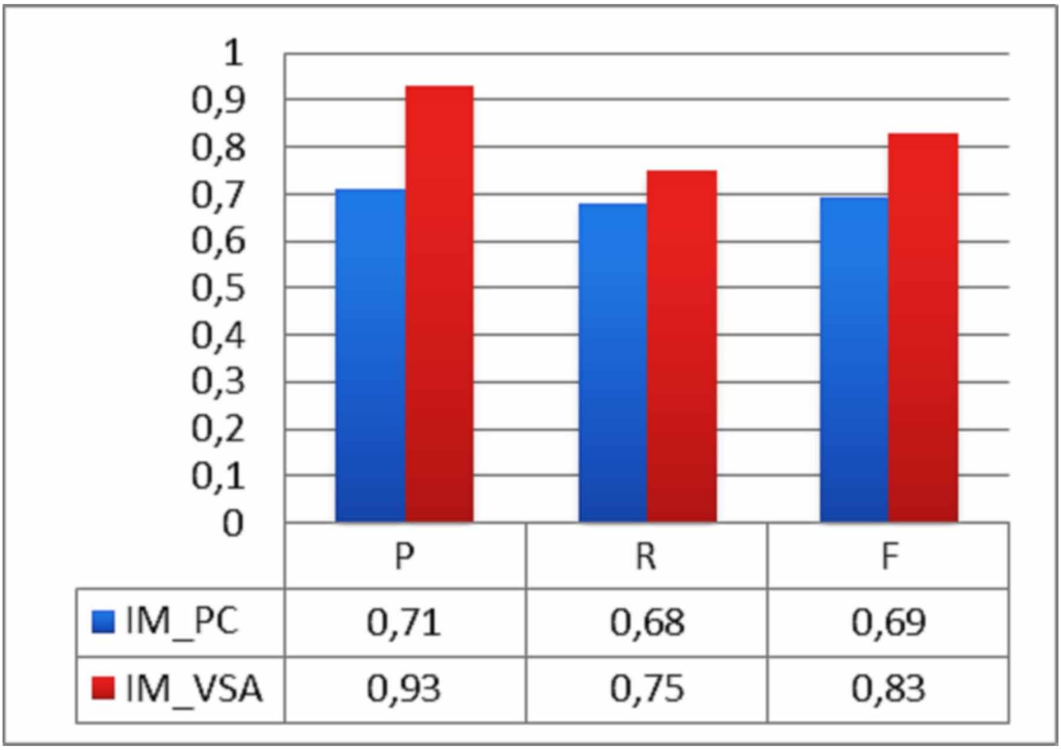
Figure 19. IM_PC performance VS IM-PC with IM_VSA Performance on Lab-Univ



	P	R	F
🟦 IM_PC	0,8	0,75	0,77
🟥 IM_VSA	0,94	0,88	0,91

using the algorithms and rules provided in (Mishra & Jain, 2016; 2015). To verify and validate the novel ontology, we propose also to use "QueryOnto" tool (Mishra & Jain, 2018).

Figure 20. IM-PC performance VS IM-PC with IM-VSA performance on Ins-Lab.



	P	R	F
🟦 IM_PC	0,71	0,68	0,69
🟥 IM_VSA	0,93	0,75	0,83

Figure 21. IM-PC performance VS IM-PC with IM-VSA performance on Univ-Ins

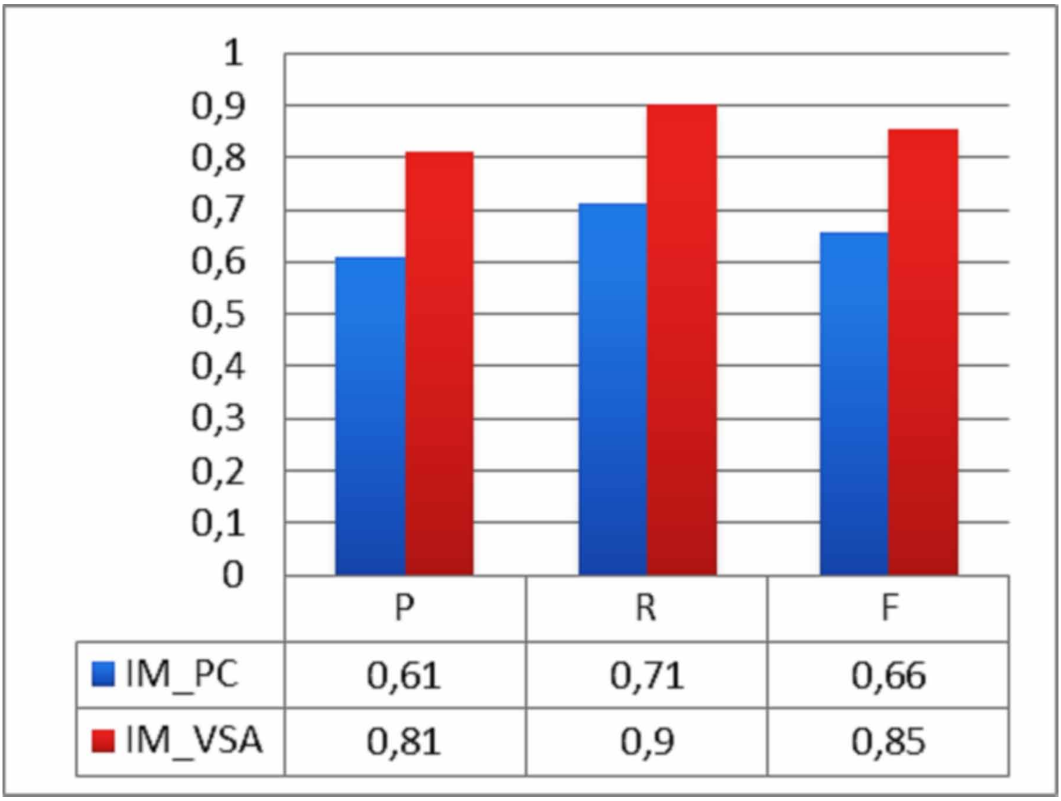


Table 3. The alignment time in the case of updating data

Datasets	Without ViewSameAs	With ViewSameAs
Lab-Univ	5min 45s	1min 15s
Ins-Lab	6min 2s	2min 46s
Univ-Ins	15min 35s	7min 08s

Table 4. The number of discovered sameAs links with and without using ViewSameAs link.

Datasets	Without ViewSameAs	With ViewSameAs
Univ_Lab_Ins	1071	1386
Lab-Univ	213	250
Ins-Lab	112	146
Univ-Ins	746	990

Table 5. IM Approaches

Approaches	Category	Candidate selection	Result refinement	Similarity methods	Final link	Alignment Reuse
ASL (2018)	Approach based on Interpretation of Instance Information	token-based blocking procedure	Similarity computation	• Modification of existing string similarities	sameAs	/
AIM-PC (2018)	Approach based on Interpretation of Instance Information	Existing blocking techniques	• Pairwise constraints • Active learning	• Mixed integer programming method • Jaccard • Vector-based similarity	sameAs	/
RIMOM-IM (2016)	Approach based on Instance Properties Classification	Distinctive information based blocking method	Matching Score Calculation	• Jaccard • Cosine • ExpAgg	sameAs	• Unique subject matching. • One-left object matching • Score Matching
SERIMI (2011; 2015)	Approach based on Interpretation of Instance Information	Existing blocking techniques	• Direct matching • Class_based matching	• Set_based similarity	sameAs	/
VMI (2013)	Approach based on Instance Properties Classification	Descriptive properties, neighboring information, meta, Name, URI	Discriminative properties	• Edit_dist-anc • Vector-based space • Cosine	sameAs	/
Wang et al approach (2013)	Approach based on Instance Properties Classification	Lexical information	Structural information	• Edit_disatnce • Vector-based space • Cosine	sameAs	/
DSSim (2008)	Approach based on Interpretation of Instance Information	/	/	• Similarity and semantic similarity algorithms • Dempster's rule • belief mass functions	sameAs	/
FBEM (2009)	Approach based on Interpretation of Instance Information	/	/	• Similarity score using sum and max. • "brute-force" matching, • similarity	sameAs	/
Hmatch (2008)	Approach based on Interpretation of Instance Information	/	/	• Instance affinity • Filler similarity	sameAs	/
Proposed approach	Approach based on Instance Properties Classification	Discriminative property values	Descriptive property values	• Vector based space • Cosine similarity	sameAs and ViewSa-meAs	• ViewSameAs based Clustering • Metadata

REFERENCES

Araujo, S., Hidders, J., Schwabe, D., & De Vries, A. P. (2011). *Serimi-resource description similarity, rdf instance matching and interlinking*. arXiv preprint arXiv:1107.1104.

Araujo, S., Tran, D. T., de Vries, A. P., & Schwabe, D. (2015). SERIMI: Class-based matching for instance matching across heterogeneous datasets. *IEEE Transactions on Knowledge and Data Engineering*, *27*(5), 1397–1440. doi:10.1109/TKDE.2014.2365779

Bizer, C., Cyganiak, R., & Heath, T. (2007). *How to publish linked data on the web*. Retrieved from http://wifo5-03.informatik.uni-mannheim.de/bizer/HowtoPublishLinkedData.htm

Castano, S., Ferrara, A., Montanelli, S., & Lorusso, D. (2008, June). Instance Matching for Ontology Population. In SEBD (pp. 121-132). Academic Press.

Cukier, K., & Mayer-Schoenberger, V. (2013). The rise of big data: How it's changing the way we think about the world. *Foreign Affairs*, *92*, 28.

De Melo, G. (2013, June). Not Quite the Same: Identity Constraints for the Web of Linked Data. AAAI.

Ding, L., Shinavier, J., Finin, T., & McGuinness, D. L. (2010). An empirical study of owl: sameAs use in Linked Data. *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*.

Dong, X., Halevy, A., & Madhavan, J. (2005, June). Reference reconciliation in complex information spaces. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 85-96). ACM. doi:10.1145/1066157.1066168

Ehrig, M. (2007). *Ontology Alignment: Bridging the Semantic Gap*. Springer.

Euzenat, J., & Shvaiko, P. (2013). *Ontology matching* (2nd ed.). Springer. doi:10.1007/978-3-642-38721-0

Ghemmaz, W., & Benchikha, F. (2016). ViewSameAs: A Novel Link in Instance Matching Process. In WEBIST (pp. 274-279). doi:10.5220/0005908102740279

Halpin, H., Hayes, P. J., McCusker, J. P., McGuinness, D. L., & Thompson, H. S. (2010, November). When owl: sameas isn't the same: An analysis of identity in linked data. In *International Semantic Web Conference* (pp. 305-320). Springer.

Hernández, M. A., & Stolfo, S. J. (1995, June). The merge/purge problem for large databases. *SIGMOD Record*, *24*(2), 127–138. doi:10.1145/568271.223807

Idrissou, A. K., Hoekstra, R., van Harmelen, F., Khalili, A., & van den Besselaar, P. (2017). Is my:sameAs the same as your:sameAs? *The ninth international conference on knowledge capture: k-cap 2017* doi:10.1145/3148011.3148029

Li, J., Tang, J., Li, Y., & Luo, Q. (2009). RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, *21*(8), 1218–1232. doi:10.1109/TKDE.2008.202

Li, J., Wang, Z., Zhang, X., & Tang, J. (2013). Large scale instance matching via multiple indexes and candidate selection. *Knowledge-Based Systems*, *50*, 112–120. doi:10.1016/j.knosys.2013.06.004

Lu, W., Dai, H., Zhang, Z., Wu, C., & Zhuang, Y. (2018). Active instance matching with pairwise constraints and its application to Chinese knowledge base construction. *Knowledge and Information Systems*, *55*(1), 171–214. doi:10.1007/s10115-017-1076-7

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60). doi:10.3115/v1/P14-5010

McCusker, J. P., & McGuinness, D. L. (2010, June). Towards Identity in Linked Data. OWLED.

Mishra, S., & Jain, S. (2016). A Unified Approach for OWL Ontologies. *International Journal of Computer Science and Information Security*, *14*(11), 747.

Mishra, S., & Jain, S. (2018). Ontologies as Semantic Model in IoT. *International Journal of Computers and Applications*, 40.

Mishra, S., Malik, S., Jain, N. K., & Jain, S. (2015). A Realist Framework for Ontologies and the Semantic Web. *Procedia Computer Science*, *70*, 483–490. doi:10.1016/j.procs.2015.10.087

Nagy, M., Vargas-Vera, M., & Stolarski, P. (2008, October). DSSim results for OAEI 2008. In *Proceedings of the 3rd International Conference on Ontology Matching-Volume 431* (pp. 147-159). CEUR-WS.org.

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic linkage of vital records. *Science*, *130*(3381), 954–959. doi:10.1126/science.130.3381.954 PMID:14426783

Nguyen, K., & Ichise, R. (2018). Automatic schema-independent linked data instance matching system. In Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications (pp. 1446-1469). IGI Global. doi:10.4018/978-1-5225-5191-1.ch065

Papaleo, L., Pernelle, N., Saïs, F., & Dumont, C. (2014, November). Logical detection of invalid sameas statements in RDF data. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 373-384). Springer. doi:10.1007/978-3-319-13704-9_29

Raad, J., Pernelle, N., & Saïs, F. (2017). Detection of Contextual Identity Links in a Knowledge Base. In *Proceedings of the Knowledge Capture Conference* (p. 8).ACM. doi:10.1145/3148011.3148032

Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. McGraw-Hill Inc.

Shao, C., Hu, L. M., Li, J. Z., Wang, Z. C., Chung, T., & Xia, J. B. (2016). RiMOM-IM: A novel iterative framework for instance matching. *Journal of Computer Science and Technology*, *31*(1), 185–197. doi:10.1007/s11390-016-1620-z

Stoermer, H., & Rassadko, N. (2009). Results of OKKAM feature based entity matching algorithm for instance matching contest of OAEI 2009. *OM, CEUR-WS. org*, 200-207.

Tang, J., Li, J., Liang, B., Huang, X., Li, Y., & Wang, K. (2006). Using Bayesian decision for ontology mapping. *Journal of Web Semantics*, *4*(4), 243–262. doi:10.1016/j.websem.2006.06.001

Wang, Z., Li, J., Zhao, Y., Setchi, R., & Tang, J. (2013). A unified approach to matching semantic data on the Web. *Knowledge-Based Systems*, *39*, 173–184. doi:10.1016/j.knosys.2012.10.015

Zhang, T., Xu, D., & Chen, J. (2008). Application-oriented purely semantic precision and recall for ontology mapping evaluation. *Knowledge-Based Systems*, *21*(8), 794–799. doi:10.1016/j.knosys.2008.03.060

ENDNOTES

[1] A Uniform Resource Identifier (URI) is a sequence of characters that identifies a logical or physical resource. A URI-reference is used to determine common usage for a URI, namely as a web address (https://www.w3.org/2001/03/identification-problem/rfc2396-uri-references.html).

[2] The Resource Description Framework (RDF) is a framework for representing information in the Web (https://www.w3.org/TR/rdf11-concepts/).

[3] http://oaei.ontologymatching.org

[4] http://oaei.ontologymatching.org/2009/

[5] http://oaei.ontologymatching.org/2010/im/

Wafa Ghemmaz has got her PhD in Computer Science at Constantine 2 University Abdelhamid Mehri, Algeria in 2019. She is currently a lecturer in the department of Software and Information Systems Technologies at Constantine 2 University . Her current research activities are conducted at the LIRE Laboratory. Her research interests include data integration, data linking, semantic web & linked data, ontology matching, and Web-based information systems.

Fouzia Benchikha is a Professor in Department of Software Technologies and Information Systems at Constantine 2 University Abdelhamid Mehri, Algeria. Her current research activities are conducted at the LIRE Laboratory from the same university. Her research interests are data modeling and integration, information systems interoperability, multi-viewpoints representation, and semantic web technology.

Maroua Bouzid received the diploma of "Ingénieur d'état" in computer science from the university of Constantine (Algeria) in 1990 and her M.Sc. degree from the university on Nancy 1 (France) in 1991 as well as her PhD degree in 1995 in temporal reasoning. She obtained the HDR degree (habilitation to supervise research) from the university of Caen (France) in 2006 in spatio-temporal reasoning. From 1996 to 2002, she was Associate Professor at The University of Artois in Lens (France) and from 2002 to 2009, she was Associate Professor at The university of Caen. Since 2009, she is Professor at the university of Caen in the computer science department.