The original publication is available http://www.springerlink.com/.
Albertoni R., De Martino M., SSONDE: Semantic Similarity on LiNked Data Entities,
MTSR 2012: Metadata and Semantics Research pp 25-36
Series: Communications in Computer and Information Science,

Dodero J.M., Palomo-Duarte M., Karampiperis P. (Ed.) vol 343
2012, Softcover
ISBN: 978-3-642-35232-4
Doi: 10.1007/978-3-642-35233-1_3

SSONDE: Semantic Similarity On liNked Data Entities

Riccardo Albertoni[1,2], Monica De Martino[2]

[1]OEG-DIA,
Facultad de Informática Universidad Politécnica de Madrid,
Boadilla del Monte, Madrid, Spain
ralbertoni@fi.upm.es
[2]CNR-IMATI,
Via De Marini, 6 – Torre di Francia - 16149 Genova, Italy
demartino@ge.imati.cnr.it

Abstract. The paper illustrates SSONDE, a framework to assess semantic similarity on linked data entities. It describes the framework architecture, its design assumptions and its configuration functionalities. SSONDE relies on an instance similarity in which asymmetricity and context dependence are specifically conceived to compare linked data resources according to their metadata. Two different applications to consume linked datasets are illustrated showing SSONDE as a building block technology to sift linked data resources.

Keywords: Instance Similarity, Linked Data, Metadata Analysis.

1 Introduction

Linked data provides a promising framework to encode, publish and share metadata of resources in scientific and industrial domains. Significant factors are enabling linked data as the ideal place where to share metadata: (i) linked data relies on light-weighed ontologies, which are encoded in Resource Description Framework (RDF) and can be exploited to provide ontology driven metadata. Such a kind of metadata takes advantage of the Open Word Assumption, enabling the adoption of domain specialized and independently developed metadata vocabularies which are pivotal to document resources produced in complex and loosely coupled pipelines; (ii) linked data is consistent with the current web architecture. It is not proposing a brand new platform replacing the existing technologies. It relies on content negotiation exploiting the standard HTTP protocol, so that, linked data solutions can be layered on existing domain-specific metadata architecture; (iii) linked data comprises a mature stack of frameworks to expose and manage metadata (e.g., D2R [1]), to retrieve non-authoritative RDF fragments published around the web (e.g., Sindice [2]), to consolidate metadata exposed in independently-provided datasets (e.g., SILK [3]), to search and navigate retrieved RDF fragments according to the entity oriented paradigm (e.g., SIGMA [4]), to query RDF fragments by appropriate query language (i.e., SPARQL), to store, manipulate and reason on these fragments once there are retrieved (e.g., Sesame, Virtuoso, Jena).

As a consequence of these enabling factors, linked data is adopted by data producers such as European Environment Agency, US and some EU Governs, whose first ambition is to share (meta)data making their processes more effective and transparent. Such as an increasing interest and involvement of data providers surely represents a genuine witness of the web of data success, but in a longer perspective, there will be a compelling need for frameworks supporting earlier linked data consumers in their decision making processes.

In this paper, we introduce SSONDE, a framework which enables a detailed comparison, ranking and selection of linked data resources through the analysis of their RDF ontology driven metadata. SSONDE implements the instance semantic similarity we presented in [5] under a linked data settings. SSONDE's similarity is especially designed to support in resource selection, namely the process stakeholders engage to choose a set of resources suitable for a given analysis purpose: (i) it deploys an *asymmetric* similarity assessment to emphasize containments between resource features, containment makes explicit information about *gains* and *losses* the stakeholders get adopting a resource in place of another; (ii) it relies on an explicit formalization of *contexts* to tailor the similarity assessment with respect to specific user-defined selection goals.

The crucial contribution of this paper is the SSONDE's JAVA open source framework, which is freely available for third parties usages. SSONDE pushes our instance similarity as a handy tool to analyze entities whose metadata are exposed as linked data. It has been designed to fit in the application layer of the *Crawling Architectural Pattern*, a linked data architectural pattern known as suitable for implementing applications on top of an open, growing set of sources [6]. SSONDE has been demonstrated in two scenarios related to the analysis of environmental and researcher metadata. Both the applications are introduced and in particular the latter is discussed in more details.

The paper is organized as follows: Section 2 introduces the SSONDE framework, describing the design assumptions, framework's components and configuration; Section 3 presents two concrete scenarios in which SSONDE has been deployed analyzing RDF metadata exposed in real linked data datasets; Section 4 discusses instance similarity's related works; Section 5 provides conclusions and future works.

2 Framework Description

SSONDE moves our context depended and asymmetric instance similarity [5] from locally stored ontology driven repositories to a settings compatible with the linked data assumptions. In order to success in this transition, SSONDE

- extends the notion of context presented in [5], making explicit the reference to namespaces, so that, it is possible to exploit properties from distinct RDF schemas in the context formalization;
- deactivates the modules computing the similarity among instances on the bases of their class hierarchies, so that, poor structured hierarchies adopted in the current linked data do not negatively affect the similarity results;

- makes the similarity assessment independent from the existence of a least upper bound (*lub*), so that, instances from distinct class hierarchy can be compared;
- revises the similarity underneath data model, assuming the adoption of the RDF model and accessing data by SPARQL instead of by Protégé API, so that, consolidated RDF framework can be exploit dealing with crawled linked data.

SSONDE is an open source framework developed in JAVA and Jena. It is conceived as a command-line tool that can be configured through a JSON file and it can be downloaded[1], used and modified for free under the GNU GPL license.

SSONDE is designed coherently to *Crawling Architectural Pattern* [6]: RDF datasets are assumed to be crawled, cleaned, integrated and locally stored exploiting framework explicitly suited for that purpose (e.g., LDIF [7]). This pattern has been selected mainly for two reasons: (i) vocabulary mapping and entity consolidation deserve to be dealt with dedicated frameworks since they strongly affect the correctness of similarity assessment; (ii) on the fly dereferencing of large sets of entities is a slow process which is even quite inefficient in term of bandwidth. Especially when SSONDE explores thousands of entities belonging to few datasets, the construction of local stores built up by exploiting RDF dumps is preferable. In case RDF dump are not available for a dataset, LDSpider [8] and Jena Fuseki can be deployed to crawl and store linked data in local RDF stores.

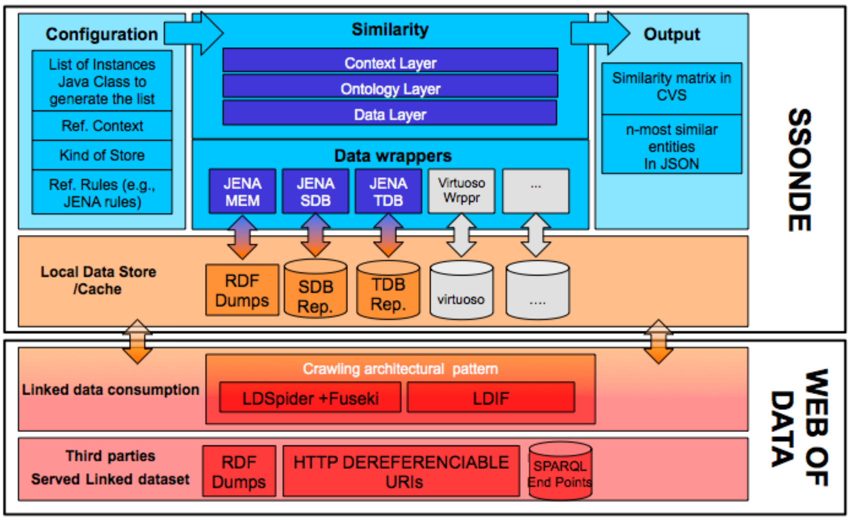

Fig. 1. Components of the semantic similarity framework SSONDE

The framework can be described in terms of different modules (see Fig. 1).

The *similarity module* deploys the semantic similarity algorithm. It is structured in: (i) a *context layer*, which provides the formalism to parameterize the similarity assessment by specifying criteria induced by application contexts. Criteria are specified in terms of features and operations to be applied comparing those features.

[1] Source code can be downloaded at http://purl.oclc.org/NET/SSONDE

The features correspond to RDF properties, which can be data properties or object properties depending on whether their values are RDF literals or instances themselves, whilst operations are functions determining how to compare the selected RDF properties; (ii) an *ontology layer* which interprets the criteria induced by the application context and compares instances related by the object properties involved in the context criteria; (iii) a *data layer* which provides similarity functions for data types and is activated by the ontology layer when data properties are involved in the context criteria. Given two resources x, y the similarity value sim(x,y) ranges between 0 and 1. The asymmetry of semantic similarity is designed to highlight the containment among resource features, which is particularly useful to interpret the resource dependencies: (i) if sim(x,y)=1 and sim(y,x)=1 then x and y have the same features; (ii) if sim(x,y)=1 and sim(y,x)<1 then the feature of x are contained in the features of y but the vice versa doesn't hold; in any case, sim(x,y) is proportional to the percentage of features that x shares with y.

The *data wrapper* module enables the access to different kinds of stores. Currently, in-memory stores as well as Jena SDB and Jena TDB stores are supported. Further RDF stores (e.g., Virtuoso, Sesame) can be included by re-implementing the OntologyModel Java Interface. Analogously, a direct access to SPARQL end points can be provided even in federated-like form, but keeping in mind that the similarity assessment is query-intensive, thus SSONDE applied on complex contexts and big set of entities might seriously affect the efficiency of the less robust SPARQL endpoints.

The *output module* provides different encodings for the results of similarity, currently Common Separated Value (CVS) and JSON encoding are supported. The CVS is used to represent the results as a similarity matrix, whilst the JSON encoding is employed to represent the first N-most similar entities for each target entities.

The c*onfiguration module* customizes the similarity assessment defining data wrappers, context, URIs of resources to be compared and output format that must be adopted in a SSONDE execution.

More details about SSONDE configuration and how to specify the context are provided in the following section.

2.1 Configuring SSONDE

Every similarity assessment performed by SSONDE must be configured providing a JSON file with the following JSON Objects:

- *StoreConfiguration*, which specifies the kind of wrapper adopted for reading RDF data, and all information related to wrapper configuration. For example, it is possible to specify the directory of store; a list of Jena rules if the wrapper provides a Jena reasoner; some URLs referring at additional RDF documents that must be dereferenced and included in the RDF data collection;
- *ContextConfiguration*, which specifies the context to be applied in the similarity assessment. Currently, it is a path referring to a text file in which the context formalization is encoded in an in-house format, but we are considering to encode context in JSON as well;

- *InstanceConfiguration*, which specifies on which instances' URIs the similarity must be worked out. A list of the URIs or a reference to a JAVA class generating on-the-fly the list of URIs can be provided. The latter option is useful when the list of instances to be compared is large and can be generated by querying the wrapped repository. In that case, the JAVA class must implement the ListOfInputInstances interface and the abstract method ArrayList getListOfInstanceURIs();
- *OutputConfiguration*, which specifies where and how the semantic similarity results must be written. Two options are supported: (i) similarity matrix encoded as a CVS file; (ii) a JSON file, in which for each of the instances included in the analysis, the similarity values with their n-most similar instances are reported.

Example 1: a JSON configuration file

The following example shows a JSON configuration file in which SSONDE reads triples from a TDB store (i.e., CNRR/data/), it dereferences a RDF schema, (i.e., "http://www.w3.org/2004/02/skos/core#"), and it assesses the similarity according to the context formalization specified in "CNRR/CCRIPubIntCoa.ctx". The similarity is worked out on resources returned by a JAVA procedure (i.e., "application.Data CNRIt.GetResearcherIMATIplusCoauthor"), and results are written as a similarity matrix encoded in the CVS file (i.e., "CNRR/CCRIPubIntCoa.res.cvs").

```
{ "StoreConfiguration":{
      "KindOfStore":"JENATDB",
      "RDFDocumentURIs":[
         "http://www.w3.org/2004/02/skos/core#"
      ],
      "TDBDirectory":"CNRR/data/"
   },
   "ContextConfiguration":{
      "ContextFilePath":" CNRR/CCRIPubIntCoa.ctx"
   },
   "InstanceConfiguration":{
      "InstanceURIsClass":"application.dataCNRIt.
GetResearcherIMATIplusCoauthor"
   },
   "OutputConfiguration":{
      "KindOfOutput":"CVSFile",
      "FilePath":" CNRR/CCRIPubIntCoa.res.cvs"}}
```

Further details pertaining to SSONDE configuration are discussed in the framework documentation. After preparing the configuration file is always advisable to validate its syntactical correctness by using one of the JSON checking services[2].

[2] e.g., JSON Formatter & Validator http://jsonformatter.curiousconcept.com

2.2 How to specify a context

In the real world, the same bunch of linked data resources can be analyzed having in mind quite different target applications, so it is very important to put in place flexible mechanisms for fine-grain customizations. In SSONDE, this kind of flexibility is provided by specifying a context for each similarity assessment. Users specify the application context indicating the resource features and operations to be considered in the similarity assessment. Resource features correspond to RDF properties, and the operations can be *Count*, *Inter* and *Simil*, which compare property values respectively according to their cardinality, their intersection and their recursive similarity. Contexts are defined as text files according to the format introduced in [5] with minor modification to consider the namespaces deriving from the adoption of multiples RDF/OWL vocabularies:

```
PREFIX namespaceA: <urlA>
PREFIX namespaceB: <urlB>
[owl:Thing]->{
 {(namespaceA:attribute1,operationForAttribute1),…
  …(namespaceB:attributeN, operationForAttributeN)},
{(namespaceA:relation1,operationForRelation1),…
  …(namespaceB:relationM, operationForRelationM)}}
```

In particular, when the operation *Simil* is applied to properties whose values are RDF literals (e.g., strings, numbers) then the values are compared considering data type similarity functions served by the *data layer*; when *Simil* is applied to properties whose values are resources themselves (aka, object properties), values are compared recursively by following the criteria specified in the context for that recursion. So, in the case the operation *Simil* is selected, i.e., (xxx:yyy, *Simil*) occurs in one of the previous pairs, the context must include what criteria to apply when the object property xxx:yyy is reached. That is done by adding the recursive path [owl:Thing, xxx:yyy] and listing its criteria as shown in the following excerpt:

```
[owl:Thing, xxx:yyy]->{
{(namespaceA:attribute1, operationForAttribute1),…
…(namespaceA:attributeN, operationForAttributeN)},
{(namespaceB:relation1, operationForRelation1),…,(
namespaceB:relationM, operationForRelationM)}}
```

Example 2: Context 1 " researcher's comparison"
This example shows a context specification defined to compare linked data resources representing researchers. It compares researchers considering the publications they share (via *pub:autoreCNRDi* property) and similarities in their research interests (via *dc:subject* property). The similarity on research interests is worked out "recursively", assuming two topics are as similar as they share *skos:broader* topics.

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX dc:  <http://purl.org/dc/terms/>
PREFIX pub:
<http://www.cnr.it/ontology/cnr/pubblicazioni.owl#>
[owl:Thing]->{{},{ (pub:autoreCNRDi, Inter),(dc:subject,
Simil)}}
[owl:Thing, dc:subject]-> {{},{(skos:broader, Inter)}}
```

Ongoing work foresees the specification of contexts encoded into JSON, besides, the set of operations specified will be made extensible so that SSONDE users can define their own operations and associate them their own JAVA implementation.

3 SSONDE Applications

This section presents two SSONDE applications on real linked data. These applications do not aim at demonstrating the correctness of our semantic similarity algorithm, because that has been already shown in [5]. Rather, they provide illustrative examples attesting SSONDE as a building block for deploying new analysis services on linked data resources.

The first application has been developed within NatureSDIplus (ECP-2007-GEO-317007), a European project aimed at developing a Spatial Data Infrastructure for Nature Conservation. We have first exposed and interlinked EUNIS Habitats and Species as linked data and successively applied SSONDE to analyze them. In this application, SSONDE is demonstrated as a new way for exploiting geographical (meta)data exposed as linked data. In particular, it is shown (i) to provide useful insight among habitats and species dependencies, enabling domain experts in ranking habitats according to the species the habitats host; (ii) to exploit different context formalizations as a mean to rank and browse habitats with respect to specific user's views.

The second application considers a linked dataset exposing metadata of large organizations, and in particular providing information pertaining to researchers and research competencies at the Italian National Research Council (CNR). Third parties have created this linked dataset at data.cnr.it as part of SemanticScout framework [9]. SSONDE is applied on such as a dataset in order to determine how researches are scientifically related. That application shows that our framework can be exploited to analyze data provided by third parties. The similarity results obtained could be eventually exploited to extend the browsing functionalities granted by SemanticScout, for example, by qualifying the researcher's coauthor according to the discovered relatedness or by providing an entity-based retrieval among the researchers.

Due to space limitation, only the results pertaining to the second application are discussed in paper. For a more detailed discussion of the first application on Habitats and Species we remand to the experiments presented in [10].

3.1 Application: comparison of researchers

SSONDE has been applied to analyze metadata exposed as linked data at data.cnr.it: the researchers pertaining to our institute CNR-IMATI including visitors and research associates. RDF fragments about CNR-IMATI researchers have been crawled from data.cnr.it by applying LDSpider and Fuseki, starting from a list of seed URIs and limiting the follow-your-nose at the solely relations mentioned in the context presented in Section 2.2. RDF fragments pertaining to researcher's interests have been transparently downloaded from dbpedia.org (i.e., the linked data version of Wikipedia) as a consequence of *dc:subject* interlinks between the researchers provided by data.cnr.it and the scientific interests provided by dbpedia.org.

SSONDE has been applied considering the JSON parameterization described in the Example 1 of Section 2.1 and the context described in Example 2 of Section 2.2. According to that context, researchers are compared with respect to their publications and scientific interests: the more two researchers share publications and have related research interests, the more SSONDE assesses them as similar. The obtained results are illustrated in the similarity matrix depicted in Fig 2. SSONDE analysis empowers users in extracting knowledge from (meta)data exposed as linked data at data.cnr.it. In particular, it supports in:

- *Discovering the linked data resource dependencies*: Fig. 2 provides information pertaining to the relatedness and the containment among researchers. The grey level of the pixel (i,j) represents the similarity value between the two researchers located at row j and column i: the darker is the pixel, the more similar are the two researchers. If a maximum similarity value (i.e., full black pixel) does not appear in the matrix diagonal, it represents a relation of *containment* among researchers. In this application, the containment emphasizes when a researcher has always produced research in collaboration with another. For example, according to Fig. 2, "Bertone" is contained in "Albertoni" and "De Martino" which means that "Bertone" during his research activity at IMATI-CNR has always performed his research in collaboration with "Albertoni" and "De Martino";
- *Discovering dataset inconsistency*: different kinds of inconsistencies can be identified by investigating unexpected similarly values. For example, in this application, from unexpected intermediate results, we have found out erroneous instance compilation (e.g., "Albertoni" was erroneously indicated as coauthor of "Guglielmi"), data missing (e.g., scientific interest for "Falcidieno" was missing), distinct resources representing the same real entity (e.g., "D'Agostino" is presented as two distinct resources);
- *Performing a cluster analysis:* starting from the similarity matrix and exploiting the framework HCE3.5[3], a cluster analysis has been performed. The resulting dendrogram illustrated in Fig. 3 closely recalls the structure in research groups at our institute CNR-IMATI: from the right to the left, we find the research group on "Computer Graphic", "Distributed Computing", "Data Semantics", "E-learning".

---

[3] http://www.cs.umd.edu/hcil/multi-cluster/

The last cluster represents the group of collaborators and host researchers that have more sporadic collaboration or visit.

Fig. 2. Similarity matrix of data retrieved from data.cnr.it

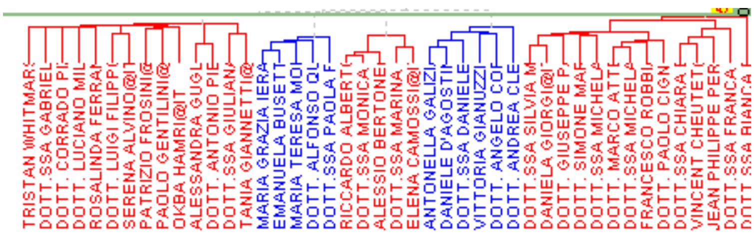

Fig. 3. Cluster analysis of data retrieved from data.cnr.it

4 Related work and discussion

The term “semantic similarity” has been used with different meanings in the literature. It sometimes refers to ontology alignment, where it enables the matching of distinct ontologies by comparing the names of the classes, attributes, relations, and instances [11]. Semantic similarity can also refer to concept similarity where it assesses the similarity among terms by considering their distinguishing features [12], their encoding in lexicographic databases [13], their encoding in conceptual spaces [14], mixing features and information theoretic approach [15]. In SSONDE, however, semantic similarity is meant as an instance similarity since this is the kind of similarity which is pivotal to support detailed comparison, ranking and selection of entities that are exposed in the web of data.

Different methods to assess instance similarity have been proposed. Some methods rely on description logics [16]; some have been applied in the context of web services [17]; some others have been applied to cluster ontology driven metadata [18, 19]. Surprisingly, none of these methods supports recognition in the case of those instances, albeit different, have effectively the same informative content: they either lack of an explicit formalization of the role of context in the entity comparison, or they fail identifying and measuring if the informative content of one overlaps or is contained in the other. Thus, the similarity results are not easily driven by explicit parameterizations or are not interpretable in terms of gain and loss the users get adopting a resource in place of another.

In the context of linked data, instance similarity is usually related to the discovering of interlinks among datasets. For example, SILK [3] is a very advanced and well-engineered tool exploiting similarity for determining *owl:sameAs* interlinks. However, it is worth noting that SSONDE and SILK deal with two different objectives: SILK compares resources assuming they might represent the same real entity, and exploits similarity to verify if they are actually the same, whereas SSONDE compares resources assuming they are different real entities and measuring at what extent they have commonalities. Even assuming SILK can be set to pursue the SSONDE’s goals, (i) SILK’s formalization of context relies on Link Specification Language (Silk-LSL) which doesn’t explicitly support the notion of recursive similarity assessment; (ii) SILK’s combines data layer similarities which are symmetric and do not explicitly support the notion of containment.

At the best of our knowledge, SSONDE is the only framework providing an instance similarity which is linked data compatible and deploys the notions of context and containment. The combination of these two notions has been shown in our past research as extremely useful when analysing metadata for comparing researchers [5], 3D objects, environment linked data [10].

5 Conclusions and future work

This paper illustrates SSONDE, an open source framework supporting in the comparison of linked data resources. SSONDE is implemented in accordance to the crawling architectural pattern, and it pushes our instance similarity as a ready-to-go tool for the analysis of linked data. SSONDE is demonstrated in two applications where metadata is analysed to enable domain experts in their decision-making processes.

Future extensions will consider new measures especially suited for geo-referenced entities, the provision of interfaces sifting entities according to their similarity (e.g., by exploiting existing visualization frameworks such as Exibit, Google visualization and JavaScript InfoVis Toolkit to support in complex information searches [20]), and the adoption of MapReduce paradigm to parallelize the similarity assessment.

Acknowledgements Part of this research activity has been performed within the project Nature-SDIplus is co-funded by the Community Programme eContentplus ECP-2007-GEO-317007. Albertoni's contribution to this work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission".

References

1. Bizer, C., Cyganiak, R.: D2R Server – Publishing Relational Databases on the Semantic Web. Poster at the 5th International Semantic Web Conference. , Athens, USA (2006).
2. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: a document-oriented lookup index for open linked data. International Journal of Metadata Semantics and Ontologies. 3, 37 (2008).
3. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and Maintaining Links on the Web of Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., and Krishnaprasad, T. (eds.) International Semantic Web Conference, LNCS 5823. Springer (2009).
4. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: Live views on the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web. 8, 8–17 (2010).
5. Albertoni, R., De Martino, M.: Asymmetric and context-dependent semantic similarity among ontology instances. J. Data Semantics. LNCS 4900, 1–30 (2008).
6. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology. 1, 1–136 (2011).
7. Schultz, A., Matteini, A., Isele, R., Mendes, P.N., Bizer, C., Becker, C.: LDIF - A Framework for Large-Scale Linked Data Integration. 21st International World Wide Web Conference (WWW2012), Developers Track. , Lyon, France (2012).
8. Isele, R., Umbrich, J., Bizer, C., Harth, A.: LDspider: An open-source crawling framework for the Web of Linked Data. Proceedings of 9th International Semantic Web Conference (ISWC 2010) Posters and Demos. pp. 6–9 (2010).
9. Baldassarre, C., Daga, E., Gangemi, A.: Semantic scout: Making sense of organizational knowledge. In: Cimiano, P. and Pinto, H.S. (eds.) Knowledge

Engineering and Management by the Masses - 17th International Conference, EKAW 2010, LNAI 6317. pp. 272–286. Springer (2010).
10. Albertoni, R., De Martino, M.: Semantic Technology to Exploit Digital Content Exposed as Linked Data. In: Cunningham, P. and Cunningham, M. (eds.) eChallenges e-2011. pp. 1–8. IIMC International Information Management Corporation, 2011 ISBN: 978-1-905824-27-4 (2011).
11. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag (2007).
12. Janowicz, K., Keßler, C., Schwarz, M., Wilkes, M., Panov, I., Espeter, M., B, B.: Algorithm , Implementation and Application of the SIM-DL Similarity Server. In: Fonseca, F., Rodríguez, M.A., and Levashkin, S. (eds.) GeoS 2007. LNCS 4853. pp. 128–145. Springer (2007).
13. Rada R Mili H Bicknell E Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions On Systems Man And Cybernetics. 19, 17–30 (1989).
14. Schwering, A.: Hybrid Model for Semantic Similarity Measurement. In: Meersman, R., Tari, Z., and Herrero, P. (eds.) On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. LNCS 3761. pp. 1449–1465. Springer (2005).
15. Pirrò, G., Euzenat, J.: A feature and information theoretic framework for semantic similarity and relatedness. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., and Glimm, B. (eds.) International Semantic Web Conference (1). LNCS 6496. pp. 615–630. Springer (2010).
16. d' Amato, C., Fanizzi, N., Esposito, F.: A dissimilarity measure for ALC concept descriptions. Proceedings of the 2006 ACM symposium on Applied computing, SAC 2006. pp. 1695–1699. ACM Press (2006).
17. Hau, J., Lee, W., Darlington, J.: A Semantic Similarity Measure for Semantic Web Services. Web Service Semantics Workshop at WWW. (2005).
18. Maedche, A., Zacharias, V.: Clustering Ontology-Based Metadata in the Semantic Web. Proceedings of Principles of Data Mining and Knowledge Discovery: 6th European Conference, PKDD 2002. LNCS 2431. pp. 348–360. Springer, Helsinki, Finland (2002).
19. Grimnes, G.A., Edwards, P., Preece, A.D.: Instance Based Clustering of Semantic Web Resources. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M. (eds.) The Semantic Web: Research and Applications, ESWC 2008. LNCS 5021. pp. 303–317. Springer (2008).
20. Albertoni, R., Bertone, A., De Martino, M.: Information Search: The Challenge of Integrating Information Visualization and Semantic Web. 16th International Workshop on Database and Expert Systems Applications DEXA05. pp. 529–533. IEEE Computer Society (2005).