

Web datasets integration with RDF-AI

François Scharffe

joint work with

Yanbin Liu, Chunguang Zhou, Jilin University

INRIA Grenoble Rhone-Alpes, France

July 11, 2009

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Web Datasets

If ontologies are the backbone of the Semantic Web, Web datasets are its flesh !

Web Datasets

A Web dataset has the following characteristics:

- ▶ It is described in RDF

Web Datasets

A Web dataset has the following characteristics:

- ▶ It is described in RDF
- ▶ It is provided and maintained by a single entity

Web Datasets

A Web dataset has the following characteristics:

- ▶ It is described in RDF
- ▶ It is provided and maintained by a single entity
- ▶ Every resources are described according to a common URI scheme

Web Datasets

A Web dataset has the following characteristics:

- ▶ It is described in RDF
- ▶ It is provided and maintained by a single entity
- ▶ Every resources are described according to a common URI scheme
- ▶ Each resource is typed according to an ontology

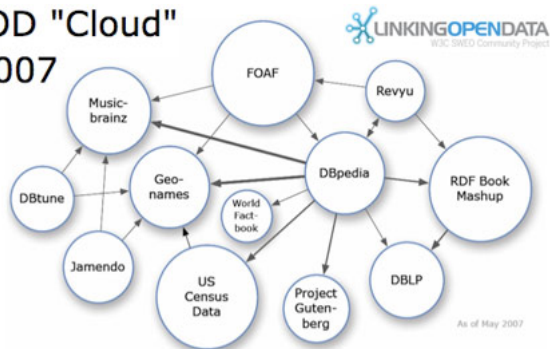
Web Datasets

A Web dataset has the following characteristics:

- ▶ It is described in RDF
- ▶ It is provided and maintained by a single entity
- ▶ Every resources are described according to a common URI scheme
- ▶ Each resource is typed according to an ontology
- ▶ Each resource URI is dereferenceable

Available datasets in the linked-data cloud

The LOD "Cloud" May 2007



Over 1 billion RDF triples served on the Web
Around 120,000 RDF links between data sources

Figure: Linked data cloud, May 2007

What's needed there

Constituting and maintaining datasets can hardly be done manually. Tools are needed to automatically:

- ▶ *Interlink* two datasets
- ▶ *Fusion* two datasets in order to extend an existing one

RDF-AI addresses these two problems.

Two approaches

- ▶ Equivalence lists (CRS) in RKB explorer
- ▶ Entity Name Servers in OKKAM Both approaches need to detect equivalent URIs.

Matching

Definition (Matching)

Given two datasets G_1 and G_2 find an *alignment* A between G_1 and G_2 .

Property

G_1 and G_2 are two Web datasets

Property

Similar resources in G_1 and G_2 are described according to the same ontologies

Web datasets illustration (1)

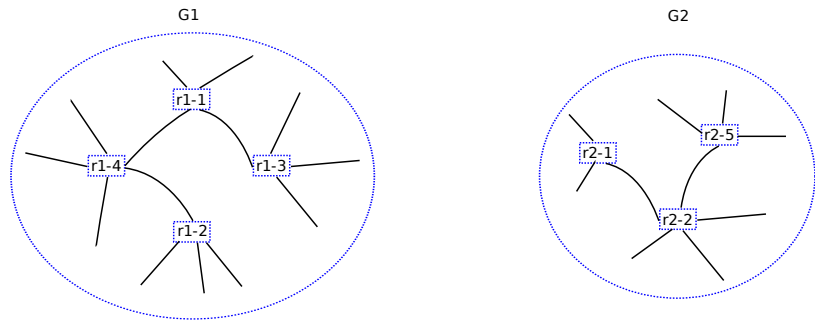


Figure: Two datasets

Web datasets illustration (2)

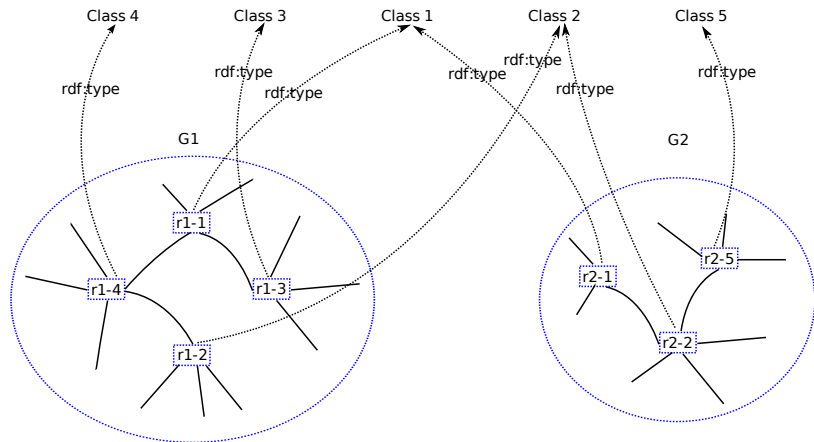


Figure: Two datasets and their ontological definitions

Aligned datasets illustration

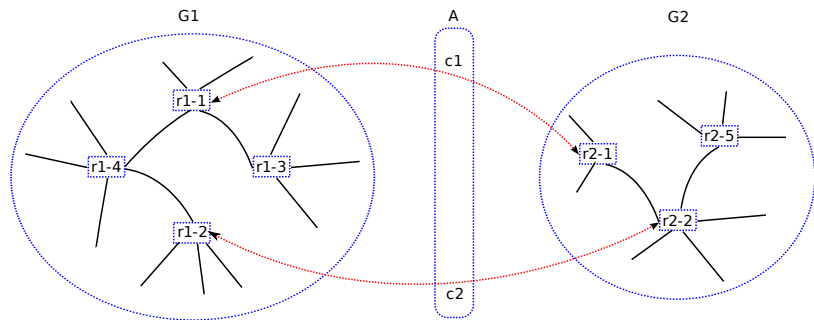


Figure: Alignment between two datasets

Interlink and Fusion

Definition (Interlink)

Given two datasets G_1 and G_2 and an alignment, construct the linkset G_3 a set of links between pairs of resources from G_1 and G_2

Definition (Fusion)

Given two datasets G_1 and G_2 and an alignment, construct the dataset G_3 resulting of fusing G_1 and G_2 according to the user input.

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Architecture Overview

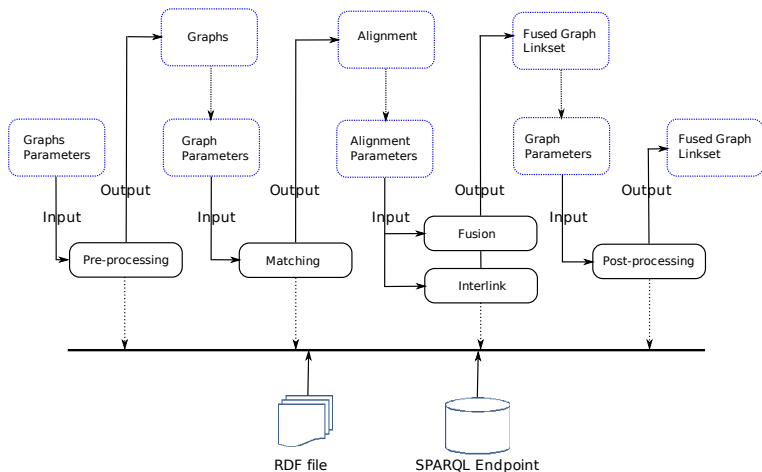


Figure: RDF-AI architecture

Pre-processing

I/O (Pre-processing)

Input: Two datasets G_1 and G_2 , a set of parameters P

Output: Two datasets G'_1 and G'_2

The pre-processing module is concern with checking and preparing the datasets for the rest of the process.

Example operations: Checking the datasets consistency wrt the ontologies, checking the datasets are described using the same version of the ontology, property values formatting.

Matching

I/O (Matching)

Input: Two datasets G_1' and G_2' , a set of parameters P

Output: An Alignment A between resources of G_1 and G_2

The matching module automatically detects equivalent resources. It can be interchanged through the use of the alignment format.

Example output:

```
<Alignment rdf:about="http://www.example.org/alignment/324684dd32">
  <map>
    <Cell>
      <entity1>
        <Instance rdf:resource="http://kmi.open.ac.uk/fusion/dblp#
          document163751_264"/>
      </entity1>
      <entity2>
        <Instance rdf:resource="http://kmi.open.ac.uk/fusion/dblp#
          document1fd88bff0db93"/>
      </entity2>
      <measure rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.620920502</
        measure>
    </Cell>
  </map>
</Alignment>
```

Interlink

I/O (Interlink)

Input: An Alignment A between resources of G_1 and G_2 , a set of parameters P

Output: A linkset L between resources of G_1 and G_2

The interlinking modules construct a linkset according to the alignment and the user input. Example linkset:

```
{
  <http://www.example.org/linkset/135erf65> a void:Linkset ;
  void:target <http://dataset1> ;
  void:target <http://dataset2> ;
  align:fromAlignment <http://www.example.org/alignment/324684dd32> ;
  align:threshold 0.5 .
}

<http://www.example.org/linkset/135erf65>
{
  <http://kmi.open.ac.uk/fusion/dblp#document163751_264>
  owl:same_as
  <http://kmi.open.ac.uk/fusion/dblp#document1fd88bff0db93> .
}
```

Fusion

I/O (Fusion)

Input: Two datasets G_1 and G_2 , an Alignment A between resources of G_1 and G_2 , a set of parameters P

Output: A dataset G_3

Details of the fusion algorithms are left open to the implementation.

Post-processing

I/O (Post-processing)

Input: A dataset G_3 , a set of parameters P

Output: A dataset G'_3

The post-processing module is concerned with checking and publishing the datasets resulting from the process. Example operations are checking the consistency of the dataset resulting from the fusion wrt the ontologies or including the linkset as part of one of the two linked datasets.

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Quick overview

Pre-processing The pre-processing module prepare the input graphs in order to: adapt the datasets to a same ontology version, translate selected properties using Google translate API, harmonize names.

Quick overview

- Pre-processing** The pre-processing module prepare the input graphs in order to: adapt the datasets to a same ontology version, translate selected properties using Google translate API, harmonize names.
- Matching** The matching modules matches the datasets and output an alignment. It uses a user configuration to select the most relevant properties for matching two resources. It uses a sequence alignment algorithm to match strings. It uses wordnet for computing a semantic similarity between words.

Quick overview

- Pre-processing** The pre-processing module prepare the input graphs in order to: adapt the datasets to a same ontology version, translate selected properties using Google translate API, harmonize names.
- Matching** The matching modules matches the datasets and output an alignment. It uses a user configuration to select the most relevant properties for matching two resources. It uses a sequence alignment algorithm to match strings. It uses wordnet for computing a semantic similarity between words.
- Interlinking** The interlinking module produces a linkset according to the alignment and a threshold provided by the user.

Quick overview

- Pre-processing** The pre-processing module prepare the input graphs in order to: adapt the datasets to a same ontology version, translate selected properties using Google translate API, harmonize names.
- Matching** The matching modules matches the datasets and output an alignment. It uses a user configuration to select the most relevant properties for matching two resources. It uses a sequence alignment algorithm to match strings. It uses wordnet for computing a semantic similarity between words.
- Interlinking** The interlinking module produces a linkset according to the alignment and a threshold provided by the user.
- Fusion** The fusion module produces a new graph. The user can select the fusion strategy: source graph, extension graph, merging or duplication of similar properties.

Quick overview

- Pre-processing** The pre-processing module prepare the input graphs in order to: adapt the datasets to a same ontology version, translate selected properties using Google translate API, harmonize names.
- Matching** The matching modules matches the datasets and output an alignment. It uses a user configuration to select the most relevant properties for matching two resources. It uses a sequence alignment algorithm to match strings. It uses wordnet for computing a semantic similarity between words.
- Interlinking** The interlinking module produces a linkset according to the alignment and a threshold provided by the user.
- Fusion** The fusion module produces a new graph. The user can select the fusion strategy: source graph, extension graph, merging or duplication of similar properties.
- Post-processing** The post processing module actually does not

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Tests on three datasets

1. Publication datasets AKT EPrints archive ¹ and Rexa ². 314 and 2103 resources.
2. Large music datasets Jamendo ³ and Musicbrainz ⁴.
3. Two small datasets about Johann Sebastian Bach works.⁵ ⁶
771 and 800 resources.

¹<http://eprints.aktors.org>

²<http://www.rexa.info>

³<http://www.jamendo.com>

⁴<http://www.musicbrainz.org>

⁵<http://www.scharffe.fr/pub/dist2008/bach-1.rdf.xml>

⁶<http://www.scharffe.fr/pub/dist2008/bach-2.rdf.xml>

Outline

Web Datasets Integration

RDF-AI Architecture

System Implementation

Experimental Results

Conclusion

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure
- ▶ Usage of the ontologies axioms to derive matches

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure
- ▶ Usage of the ontologies axioms to derive matches
- ▶ Usage of other thesauri like SKOS

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure
- ▶ Usage of the ontologies axioms to derive matches
- ▶ Usage of other thesauri like SKOS
- ▶ Implementation of the consistency checking functionality

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure
- ▶ Usage of the ontologies axioms to derive matches
- ▶ Usage of other thesauri like SKOS
- ▶ Implementation of the consistency checking functionality
- ▶ Dealing with large datasets by dynamically querying resources

Conclusion

- ▶ RDF-AI is an architecture and implementation for Web datasets alignment, interlink and Fusion
- ▶ Initial framework and prototypes have been presented here but there is still a lot to do:
- ▶ Formalizing the problem using graph grammars
- ▶ Automatic acquisition of the datasets structure
- ▶ Usage of the ontologies axioms to derive matches
- ▶ Usage of other thesauri like SKOS
- ▶ Implementation of the consistency checking functionality
- ▶ Dealing with large datasets by dynamically querying resources
- ▶ Multi ontologies, usage of ontology alignments

2 ongoing activities

- ▶ Data matching at the Ontology Alignment Evaluation Initiative (OAEI)

<http://oaei.ontologymatching.org/2009/>

2 ongoing activities

- ▶ Data matching at the Ontology Alignment Evaluation Initiative (OAEI)
<http://oaei.ontologymatching.org/2009/>
- ▶ MeLinDa, finding commonalities in data matchers input descriptions. <http://melinda.inrialpes.fr>

Questions

- ▶ What level of automation is feasible ?

Questions

- ▶ What level of automation is feasible ?
- ▶ Links quality ?

Thank you !

<http://code.google.com/p/rdfai/>

This presentation is available at

<http://www.scharffe.fr/presentations/>

[rdf-ai-presentation-STI-research-seminar-01-12-2008.pdf](http://www.scharffe.fr/presentations/rdf-ai-presentation-STI-research-seminar-01-12-2008.pdf)