NEUROWIKI:

HOW WE INTEGRATED LARGE DATASETS INTO SMW WITH LDIF
LINKED DATA CHALLENGES

• obtaining the data and keeping it up to date
• data sources use a wide range of different RDF vocabularies to represent data about the same type of entity
• different URIs identify the same real-world entity
LDIF – LINKED DATA INTEGRATION FRAMEWORK

- manages data download and update
- translates heterogeneous data into a single local target vocabulary
- replaces URI aliases with a single target URI on the client side
- outputs the results to files or a quad store with provenance

- LDIF is available standalone and as part of Ontoprise TSC in conjunction with the SMW+ Linked Data Extension
- Supported in part by Vulcan Inc. as part of its Project Halo and by the EU FP7 project LOD2 - Creating Knowledge out of Interlinked Data (Grant No. 257943).
LINKED DATA APPLICATION ARCHITECTURE
CURRENT LDIF GOAL

Scalability!

<table>
<thead>
<tr>
<th></th>
<th>25M</th>
<th>50M</th>
<th>100M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load and build entities for R2R</td>
<td>128.1 sec</td>
<td>297.2 sec</td>
<td>1059.7 sec</td>
</tr>
<tr>
<td>R2R data translation</td>
<td>169.9 sec</td>
<td>515.0 sec</td>
<td>1109.2 sec</td>
</tr>
<tr>
<td>Build entities for Silk</td>
<td>15.3 sec</td>
<td>36.8 sec</td>
<td>107.4 sec</td>
</tr>
<tr>
<td>Silk Identity Resolution</td>
<td>103.0 sec</td>
<td>568.5 sec</td>
<td>2954.9 sec</td>
</tr>
<tr>
<td>Final URI rewriting</td>
<td>8.1 sec</td>
<td>27.0 sec</td>
<td>65.0 sec</td>
</tr>
<tr>
<td>Overall execution time</td>
<td>7.0 min</td>
<td>24.0 min</td>
<td>88.3 min</td>
</tr>
</tbody>
</table>
R2R INTRODUCTION
A Problem-Solution Approach
PROBLEM: NAME MISMATCH

Example: Music Artist

- Music Ontology: `mo:MusicArtist`
- DBpedia ontology: `dbpedia-owl:MusicalArtist`
SOLUTION TO NAME MISMATCH

SourcePattern

dbpedia-owl:MusicalArtist

TargetPattern

mo:MusicArtist
PROBLEM: STRUCTURE MISMATCH

• Example: DBpedia vs. Factbook
  
• DBpedia ontology:
    
    Property: dbpedia-owl:leaderName

    Property: foaf:name

• Factbook Ontology:
  
    Property: factbook:executivebranch_headofgovernment
SOLUTION TO STRUCTURE MISMATCH

SourcePattern

TargetPattern

- dbpedia-owl:leaderName
- foaf:name
- factbook:executivebranch_headofgovernment
MODIFIERS

• Datatype modifier: “120” => “120”^^xsd:double

• Language modifier
  • Used to specify a language tag for a literal

• URI modifier
  • To convert a value into a URI
  • Example: "http://dbpedia.org" to <http://dbpedia.org>

• Literal modifier
  • To convert a URI into a literal
PROBLEM: SCHEMA MISMATCH

• Example: Music Ontology vs. DBpedia

  • Music Ontology:
    
    Class: mo:Record
    Property: mo:release_type
    Instance: mo:single

  • DBpedia ontology:
    
    Class: dbpedia-owl:Single
SOLUTION: RESTRICTIONS

SourcePattern

TargetPattern

mo:release_type

mo:record

mo:single

rdf:type

dbpedia-owl:Single

rdf:type
PROBLEM: VALUE MISMATCH

• Example: PharmGKB vs. SMW

  • PharmGKB Ontology:
    Property: pharmgkb:Drugbank_Id
    Value example: <http://chem2biordf.org/.../DB00317>

  • SMW ontology:
    Property: smwprop:DrugBankId
    Value example: "DB00317"^^xsd:string
SOLUTION: TRANSFORMATIONS

SourcePattern

pharmgkb:DrugBankId

?x

TargetPattern

smwprop:DrugbankId

^^xsd:string

regexToList('http://chem2bio2rdf.org/drugbank/resource/drugbank_drug/(.+?)', ?x)
SILK INTRODUCTION

By Example
SILK: LINKAGE RULES

• Specify when two entities should be considered the same

• Elements of a linkage rule:
  • Restrict the set of entities you want to compare
  • Pick the relevant values
  • Compare the values
  • Aggregate results of different comparators
  • Results above a defined threshold are considered as matches
HOW TO CHOOSE AND COMPARE VALUES

• Example use case: Mainstream music bands
  • Band names are usually trademarked
  • Assumption: band names are unique
SOLUTION: PATH INPUT AND COMPARATORS

Entities represented by ?a and ?b are restricted to mo:MusicGroup
EXAMPLES OF OTHER COMPARATORS

• String based similarities like Edit Distance
• Token based similarities like Jaccard's Coefficient
• Data type specific: geo-coordinates, date types etc.
PROBLEM: NON-UNIQUE NAMES

• Example: Local music bands
  • Band names are usually not trademarked
  • Assumption: Band names are NOT unique
→ Not enough to only compare band names!
SOLUTION: COMBINE COMPARATORS WITH AGGREGATION

Output: Between 0 and 1

Entities represented by ?a and ?b are restricted to mo:MusicGroup
LDIF PIPELINE

1. Collect data
2. Map to Schema
3. Resolve Identities
4. Output

- Parallelized on one machine
- Next release will add Hadoop support
LDIF PIPELINE

1. Collect data
2. Map to Schema
3. Resolve Identities
4. Output

Supported data sources:
- RDF dumps (various formats)
- SPARQL Endpoints
- Crawling Linked Data
<table>
<thead>
<tr>
<th>Name</th>
<th>Data source</th>
<th>Last import</th>
<th>Change frequency</th>
<th>Imported</th>
<th>Status message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiderOriginal</td>
<td><a href="http://mes.smw-lde.eu.s3.amazonaws.com/sider_dump_fixed.nt.bz2">http://mes.smw-lde.eu.s3.amazonaws.com/sider_dump_fixed.nt.bz2</a></td>
<td>9/15/11 8:50 AM</td>
<td>-</td>
<td>yes</td>
<td>Unable to find a matching description that maps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.example.org/smw-lde/smwDatasources/SiderOriginal">http://www.example.org/smw-lde/smwDatasources/SiderOriginal</a> to wiki</td>
</tr>
<tr>
<td>KEGGGeneOriginal</td>
<td><a href="http://mes.smw-lde.eu.s3.amazonaws.com/kegg_genes_20101018_100.nt.bz2">http://mes.smw-lde.eu.s3.amazonaws.com/kegg_genes_20101018_100.nt.bz2</a></td>
<td>9/15/11 9:13 AM</td>
<td>-</td>
<td>yes</td>
<td>Unable to find a matching description that maps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.example.org/smw-lde/smwDatasources/KEGGGeneOriginal">http://www.example.org/smw-lde/smwDatasources/KEGGGeneOriginal</a> to wiki</td>
</tr>
<tr>
<td>DrugbankOriginal</td>
<td><a href="http://mes.smw-lde.eu.s3.amazonaws.com/drugbank_dump_fixed.nt.bz2">http://mes.smw-lde.eu.s3.amazonaws.com/drugbank_dump_fixed.nt.bz2</a></td>
<td>9/15/11 6:29 AM</td>
<td>-</td>
<td>yes</td>
<td>Unable to find a matching description that maps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><a href="http://www.example.org/smw-lde/smwDatasources/DrugbankOriginal">http://www.example.org/smw-lde/smwDatasources/DrugbankOriginal</a> to wiki</td>
</tr>
</tbody>
</table>
Data source definition for DiseasomeOriginal

The Linked Data source definition was parsed successfully. The following values will be stored:

<table>
<thead>
<tr>
<th>ID</th>
<th>DiseasomeOriginal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>DiseasomeOriginal</td>
</tr>
<tr>
<td>Data dump location</td>
<td><a href="http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2">http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2</a></td>
</tr>
</tbody>
</table>

Data source definition for DiseasomeMapped

The Linked Data source definition was parsed successfully. The following values will be stored:

<table>
<thead>
<tr>
<th>ID</th>
<th>DiseasomeMapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>DiseasomeMapped</td>
</tr>
<tr>
<td>Data dump location</td>
<td><a href="http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2">http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2</a></td>
</tr>
</tbody>
</table>

Data source definition for DBpediaToDiseasome

The Linked Data source definition was parsed successfully. The following values will be stored:

<table>
<thead>
<tr>
<th>ID</th>
<th>DBpediaToDiseasome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>DBpediaToDiseasome</td>
</tr>
<tr>
<td>Data dump location</td>
<td><a href="http://mes.smw-lde-eu.s3.amazonaws.com/dbpedia_diseasome.nt.bz2">http://mes.smw-lde-eu.s3.amazonaws.com/dbpedia_diseasome.nt.bz2</a></td>
</tr>
</tbody>
</table>

Drugbank

Note: drugbank_dump_fixed.nt has the following two lines were commented out from http://www4.wiwi.fu-berlin.de/drugbank/drugbank_dump.nt.bz2:

- <http://www4.wiwi.fu-berlin.de/drugbank/resource/drugs/DB00013>
- <http://www4.wiwi.fu-berlin.de/drugbank/resource/drugbank/rxlistLink>
  <http://www.rxlist.com/cgi/pharmclips2.cgi?keyword=%20Abbokinase%20AE>
Disease

{{#sourcedefinition:
  id = DiseasomeOriginal
  Label = DiseasomeOriginal
  DataDumpLocation = <http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2>
}}

{{#sourcedefinition:
  id = DiseasomeMapped
  Label = DiseasomeMapped
  DataDumpLocation = <http://mes.smw-lde-eu.s3.amazonaws.com/diseasome_dump_fixed.nt.bz2>
}}

{{#sourcedefinition:
  id = DBpediaToDiseasome
  Label = DBpediaToDiseasome
  DataDumpLocation = <http://mes.smw-lde-eu.s3.amazonaws.com/dbpedia_diseasome.nt.bz2>
}}

Drugbank

Note: drugbank_dump_fixed.nt has the following two lines were commented out from [http://www4.wiwiss.fu-berlin.de/drugBank/drugbank_dump.nt.bz2 http://www4.wiwiss.fu-berlin.de/drugBank/drugbank_dump.nt.bz2]

Press Ctrl+Alt+Space to use auto-completion. (Ctrl+Space in IE)

Please note that all contributions to Neuro Wiki may be edited, altered, or removed by other contributors. If you do not want your writing to be edited mercilessly, then do not submit it here.

You are also promising us that you wrote this yourself, or copied it from a public domain or similar free resource (see Neuro Wiki:Copyrights for details).

Do not submit copyrighted work without permission!
LDIF PIPELINE

1. Collect data
2. Map to Schema
3. Resolve Identities
4. Output

Using R2R

KEGG:gene      wiki:Gene
KEGG:hasPathway wiki:IsInvolvedIn
## All R2R Mappings

<table>
<thead>
<tr>
<th>ID</th>
<th>From</th>
<th>To</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiseasesMapped_to_Wiki_Mapping_1</td>
<td>DiseasesMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>DrugbankMapped_to_Wiki_Mapping_1</td>
<td>DrugbankMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>DrugbankMapped_to_Wiki_Mapping_2</td>
<td>DrugbankMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>KEGGGeneMapped_to_Wiki_Mapping_1</td>
<td>KEGGGeneMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>KEGGGeneMapped_to_Wiki_Mapping_4</td>
<td>KEGGGeneMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>KEGGPathwayMapped_to_Wiki_Mapping_1</td>
<td>KEGGPathwayMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>KEGGPathwayMapped_to_Wiki_Mapping_5</td>
<td>KEGGPathwayMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>SiderMapped_to_Wiki_Mapping_1</td>
<td>SiderMapped</td>
<td>Wiki</td>
<td></td>
</tr>
<tr>
<td>SiderMapped_to_Wiki_Mapping_3</td>
<td>SiderMapped</td>
<td>Wiki</td>
<td></td>
</tr>
</tbody>
</table>
### DiseasesMAPPED_to_Wiki_Mapping_1

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Target</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp:Disease</td>
<td>diseasesome:diseases</td>
<td>smwcat:Disease</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
<tr>
<td>mp:associatedGene</td>
<td>diseaseome:associatedGene</td>
<td>smwprop:associatedGene</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
<tr>
<td>mp:diseaseLabel</td>
<td>rdfs:label</td>
<td>rdfs:label</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
<tr>
<td>mp:possibleDrug</td>
<td>diseaseome:possibleDrug</td>
<td>smwprop:possibleDrug</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
</tbody>
</table>

**New Property Mapping**

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Target</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp:Gene</td>
<td>genes</td>
<td>smwcat:Gene</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
<tr>
<td>mp:geneLabel</td>
<td>rdfs:label</td>
<td>rdfs:label</td>
<td><img src="edit-icon" alt="Edit" /></td>
</tr>
</tbody>
</table>

**New Property Mapping**

**New Class Mapping**

**Back to Overview**
**Remove Mapping**
Edit Property Mapping

mp:diseaseLabel

?SUBJ rdfs:label ?label

?SUBJ rdfs:label ?labelTransformed

?labelTransformed = regexToList('(.+?)(?:, [0-9]+)?', ?label)
Edit Property Mapping

Transformation

```
?labelTransformed = regexToList('(.+)?(\?;\: [0-9]+)?', ?label)
```

String functions
- `split(regex, stringarg)`
- `replaceAll(thisRegex, withString, inThisString)`

Arithmetic functions

List functions
- `xpath_matches(s, pattern)`
- `xpath_replace(s, pattern, replacement)`

XPath functions
- `regexToList(regex, stringarg)`

Returns a list of strings as specified by the regex

Save  Cancel
LDIF PIPELINE

1. Collect data
2. Map to Schema
3. Resolve Identities
4. Output

Using Silk

Berlin

Berlin, Germany
Berlin, CT
Berlin, MD
Berlin, NJ
Berlin, MA

52° 31' N, 13° 24' O
LDIF PIPELINE

1. Collect data
2. Map to Schema
3. Resolve Identities
4. Output

Output options:
- N-Quads
- N-Triples
- SPARQL Update Stream
- Includes provenance
Q & A
THANKS!

• We’re looking for first adopters!
• Website: http://bit.ly/ldifweb