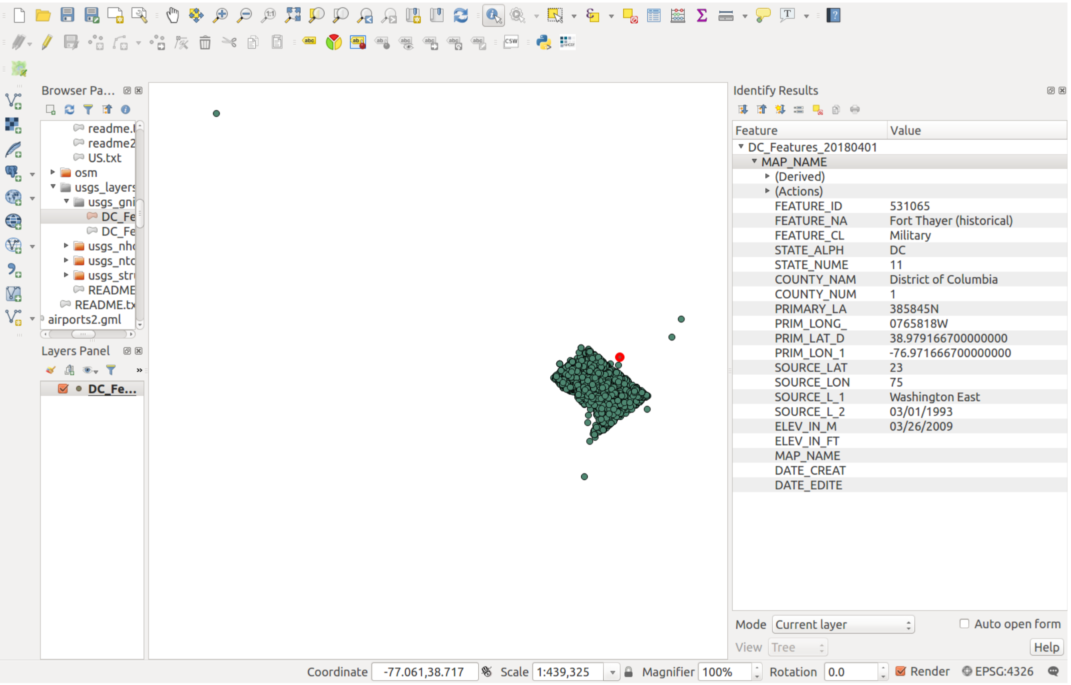
Advanced Feature Descriptions

User Manual v1.0

Part 2: LIMES Modeling

Description: This tutorial steps through using LIMES to create a Link Specification to create co-references between Geonames.org and GNIS datasets (over Washington D.C.).

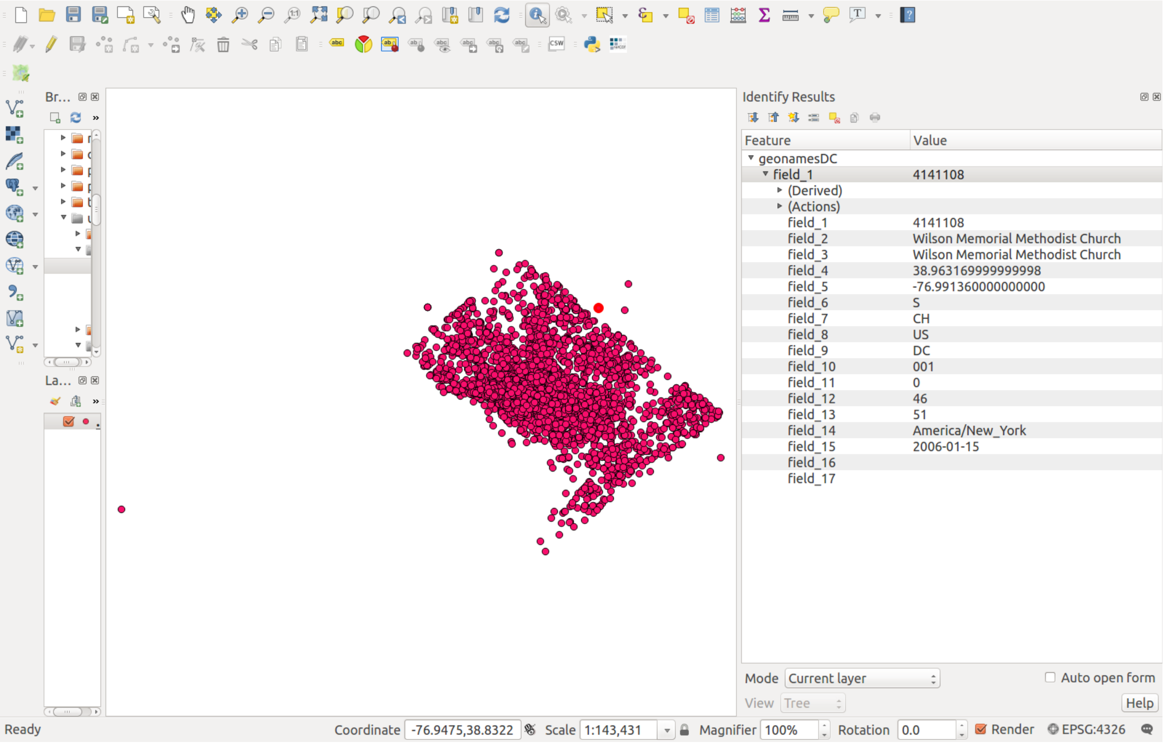
Example GNIS Schema and Values:



Feature	Value
DC_Features_20180401	
MAP_NAME	
(Derived)	
(Actions)	
FEATURE_ID	531065
FEATURE_NA	Fort Thayer (historical)
FEATURE_CL	Military
STATE_ALPH	DC
STATE_NUME	11
COUNTY_NAM	District of Columbia
COUNTY_NUM	1
PRIMARY_LA	385845N
PRIM_LONG_	0765818W
PRIM_LAT_D	38.979166700000000
PRIM_LON_1	-76.971666700000000
SOURCE_LAT	23
SOURCE_LON	75
SOURCE_L_1	Washington East
SOURCE_L_2	03/01/1993
ELEV_IN_M	03/26/2009
ELEV_IN_FT	
MAP_NAME	
DATE_CREAT	
DATE_EDITE	

Coordinate -77.061,38.717 Scale 1:439,325 Magnifier 100% Rotation 0.0 ☑ Render EPSG:4326

Example Geonames.org Schema and Values:



Feature	Value
geonamesDC	
field_1	4141108
(Derived)	
(Actions)	
field_1	4141108
field_2	Wilson Memorial Methodist Church
field_3	Wilson Memorial Methodist Church
field_4	38.963169999999998
field_5	-76.991360000000000
field_6	S
field_7	CH
field_8	US
field_9	DC
field_10	001
field_11	0
field_12	46
field_13	51
field_14	America/New_York
field_15	2006-01-15
field_16	
field_17	

Ready Coordinate -76.9475,38.8322 Scale 1:143,431 Magnifier 100% Rotation 0.0 ☑ Render EPSG:4326

Generate RDF from Geoserver data via Karma As A Service:

(NOTE: LIMES needs input datasets in RDF; these steps will generate the needed RDF; however, **if using the example datasets, the example dataset RDF has been provided in afd/limes/input folder.**)

1. **To generate the SOURCE dataset (geonamesDC.shp) as RDF, execute:**

```
curl --request POST --data
'R2rmlURI=http://localhost:8080/examples/usgs/r2rml/geonamesDC2-
model.ttl&ContentType=XML&DataURL=http://localhost:8080/geoserver/usgsns/wfs?service
%3Dwfs%26version%3D2.0.0%26request%3DGetFeature%26typeNames%3Dusgsns
%3AgeonamesDC' http://localhost:8080/web-services-rdf-0.0.1-SNAPSHOT/rdf/r2rml/rdf >
geonamesDC.nt
```

IMPORTANT: The “DataURL” parameter is a WFS URL that contains ampersands within another URL parameter. Therefore, make sure to ***URL encode*** everything after the “?” in the WFS URL. For the example above, the resulting URL encoding is:

```
service%3Dwfs%26version%3D2.0.0%26request%3DGetFeature%26typeNames
%3Dusgsns%3AgeonamesDC
```

The resulting output file, geonamesDC.nt, will be written to the current directory.

2. **To generate the TARGET dataset (DC_Features_20180401.shp) as RDF, execute:**

```
curl --request POST --data
'R2rmlURI=http://localhost:8080/examples/usgs/r2rml/usgs_gnis3-
model.ttl&ContentType=XML&DataURL=http://localhost:8080/geoserver/usgsns/wfs?service
%3Dwfs%26version%3D2.0.0%26request%3DGetFeature%26typeNames%3Dusgsns
%3AGNIS_DC_Features_20180401' http://localhost:8080/web-services-rdf-0.0.1-
SNAPSHOT/rdf/r2rml/rdf > gnisDC.nt
```

IMPORTANT: The “DataURL” parameter is a WFS URL that contains ampersands within another URL parameter. Therefore, make sure to ***URL encode*** everything after the “?” in the WFS URL. For the example above, the resulting URL encoding is:

```
service%3Dwfs%26version%3D2.0.0%26request%3DGetFeature%26typeNames
%3Dusgsns%3AGNIS_DC_Features_20180401
```

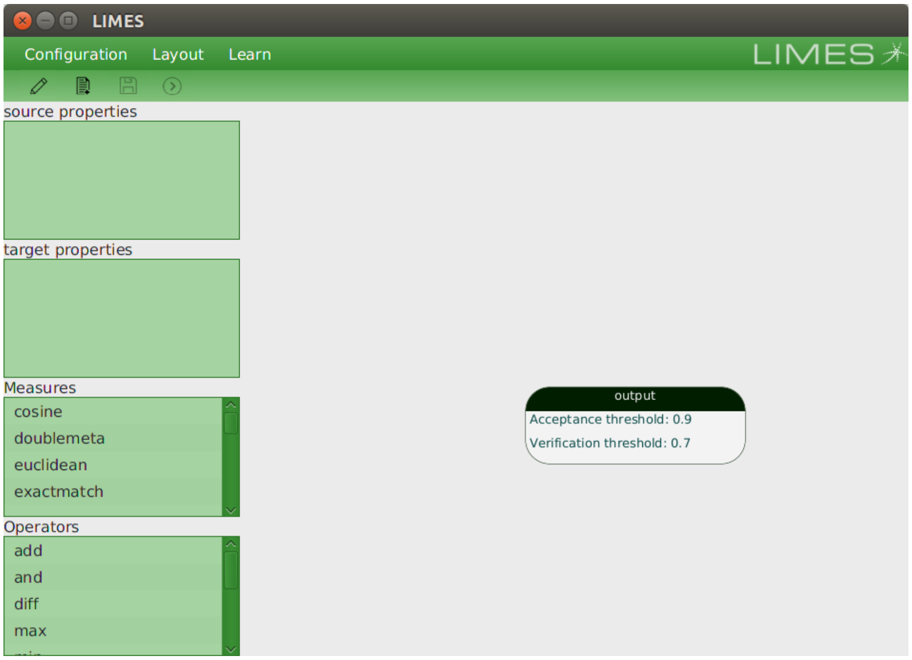
The resulting output file, gnisDC.nt, will be written to the current directory.

LIMES Modeling Instructions:

1. **Start LIMES GUI:**

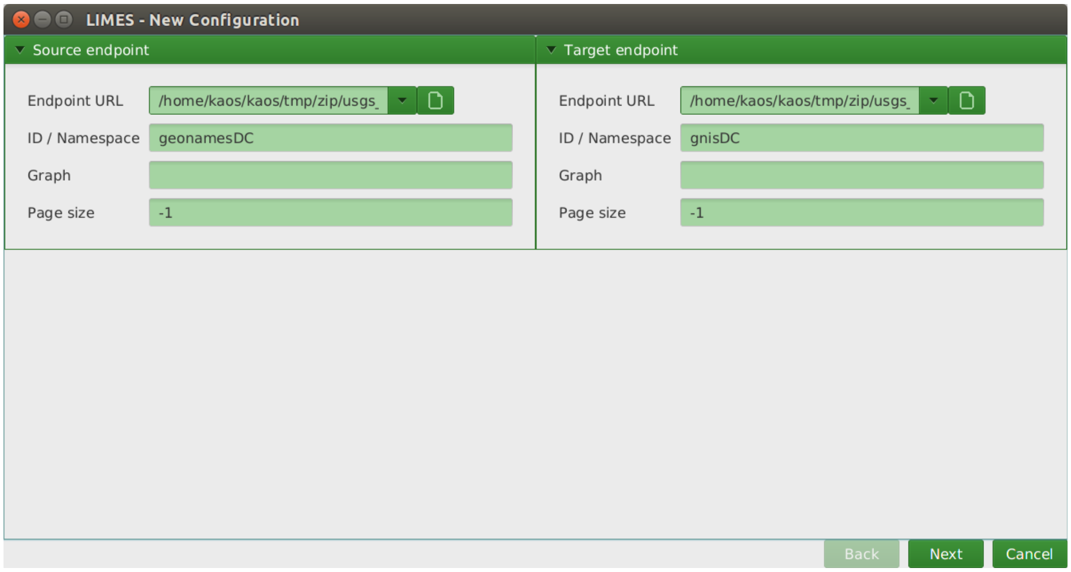
   ```
   LIMES_ROOT_DIR/limes-gui/target/jfx/app$ java -jar limes-gui-1.5.5-jfx.jar
   ```

   The following screen should display:



2. **Configure the SOURCE and TARGET datasets:**

   (a) On the top menu, go to: Configuration → New, to create a new configuration
   (b) Browse to and select the geonamesDC.nt file for the Source Endpoint URL
   (c) Browse to and select the gnisDC.nt file for the Target Endpoint URL
   (d) Set the Source ID to: geonamesDC
   (e) Set the Target ID to: gnisDC
   (f) The populated form should look like the following image:



Source endpoint

Endpoint URL /home/kaos/kaos/tmp/zip/usgs_

ID / Namespace geonamesDC

Graph

Page size -1

Target endpoint

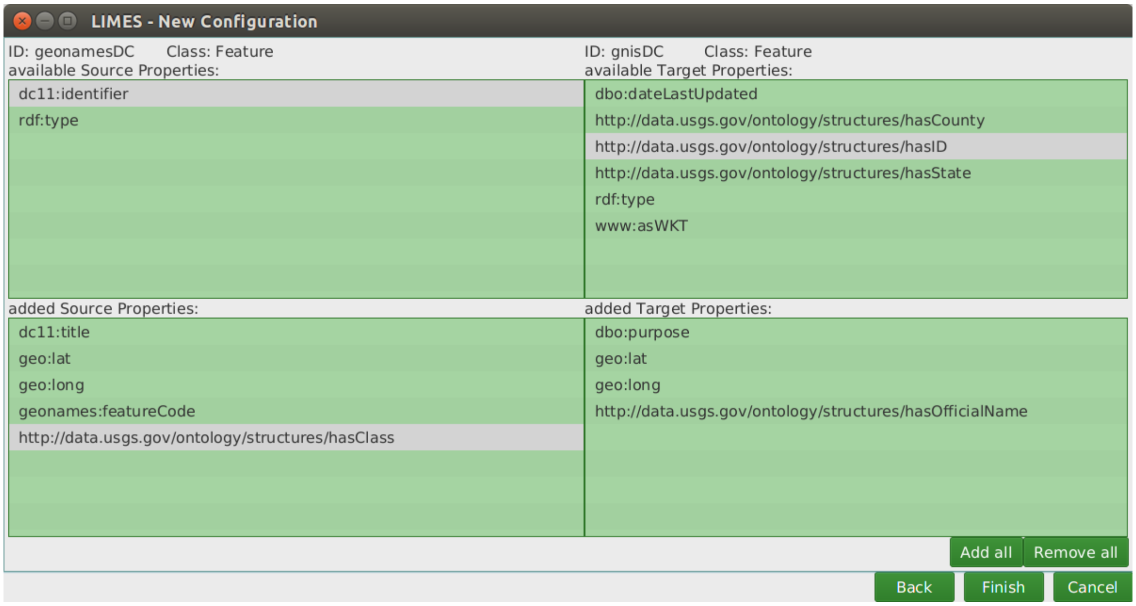
Endpoint URL /home/kaos/kaos/tmp/zip/usgs_

ID / Namespace gnisDC

Graph

Page size -1

(g) Click on “Next”
(h) LIMES will find the classes it can detect in both data sources and allow you to select a specific class type in each. In this case, there are only geosparql:Feature classes for both, so select Feature and press “Next”
(i) LIMES then finds the available properties for the selected classes from the previous step. In this case, most of the properties have semantic value that can be used to help determine if a given pair of instances from the source and target datasets reference the same data entity. Thus, select all of the *source* properties except the dc11:identifier and rdf:type properties. Next, select the *target* properties: dbo:purpose, geo:lat, geo:lon, usgs:hasOfficialName.
(j) The resulting populated form should look like the following:



ID: geonamesDC Class: Feature

available Source Properties:

dc11:identifier
rdf:type

added Source Properties:

dc11:title
geo:lat
geo:long
geonames:featureCode
http://data.usgs.gov/ontology/structures/hasClass

ID: gnisDC Class: Feature

available Target Properties:

dbo:dateLastUpdated
http://data.usgs.gov/ontology/structures/hasCounty
http://data.usgs.gov/ontology/structures/hasID
http://data.usgs.gov/ontology/structures/hasState
rdf:type
www:asWKT

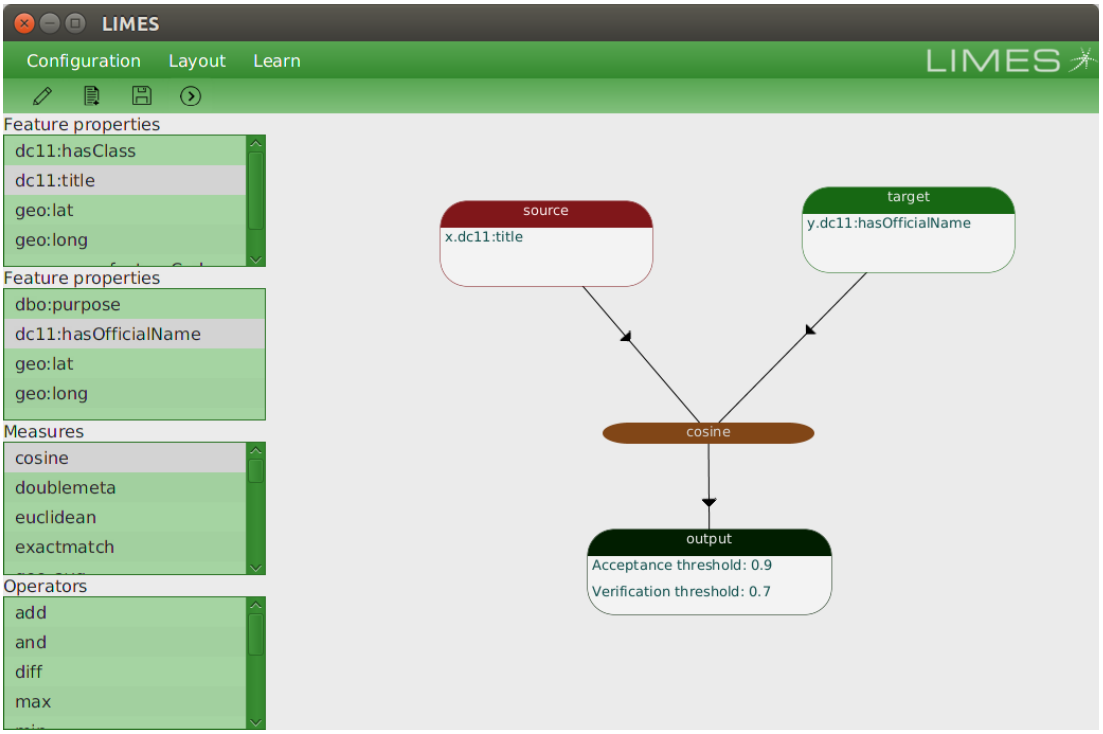
added Target Properties:

dbo:purpose
geo:lat
geo:long
http://data.usgs.gov/ontology/structures/hasOfficialName

(k) Press “Finish”

3. **Create a Link Specification which defines the algorithms and workflow used to compare pairs of entities from the source and target datasets.**

(a) To start, compare pairs of entities by name matching only. On the main screen, drag and drop "dc11:title" property from the source properties on the left side of the screen.
(b) Similarly, drag and drop "dc11:hasOfficialName" property from the target properties.
(c) Next, drag and drop "cosine" from the available Measures. This will perform a cosine distance comparison between the source and target "name" fields.
(d) In the workspace, "right click → Link To" and connect Source node to the Cosine node.
(e) Similarly, right click → Link To on the Target node and connect it to the Cosine node.
(f) Lastly, right click → Link To on the Cosine node and connect it to the Output node.
(g) The Output node has default value of 0.9 Acceptance and 0.7 Verification. If a comparison results in a score >= 0.9, then an owl:sameAs triple is output. These values can be modified by double clicking on the Output node. It's useful to adjust these values to test how the precision and recall of the Link Specification varies as changes are made to the specification.
(h) At this point the Link Specification should look like the following image:



x.dc11:title

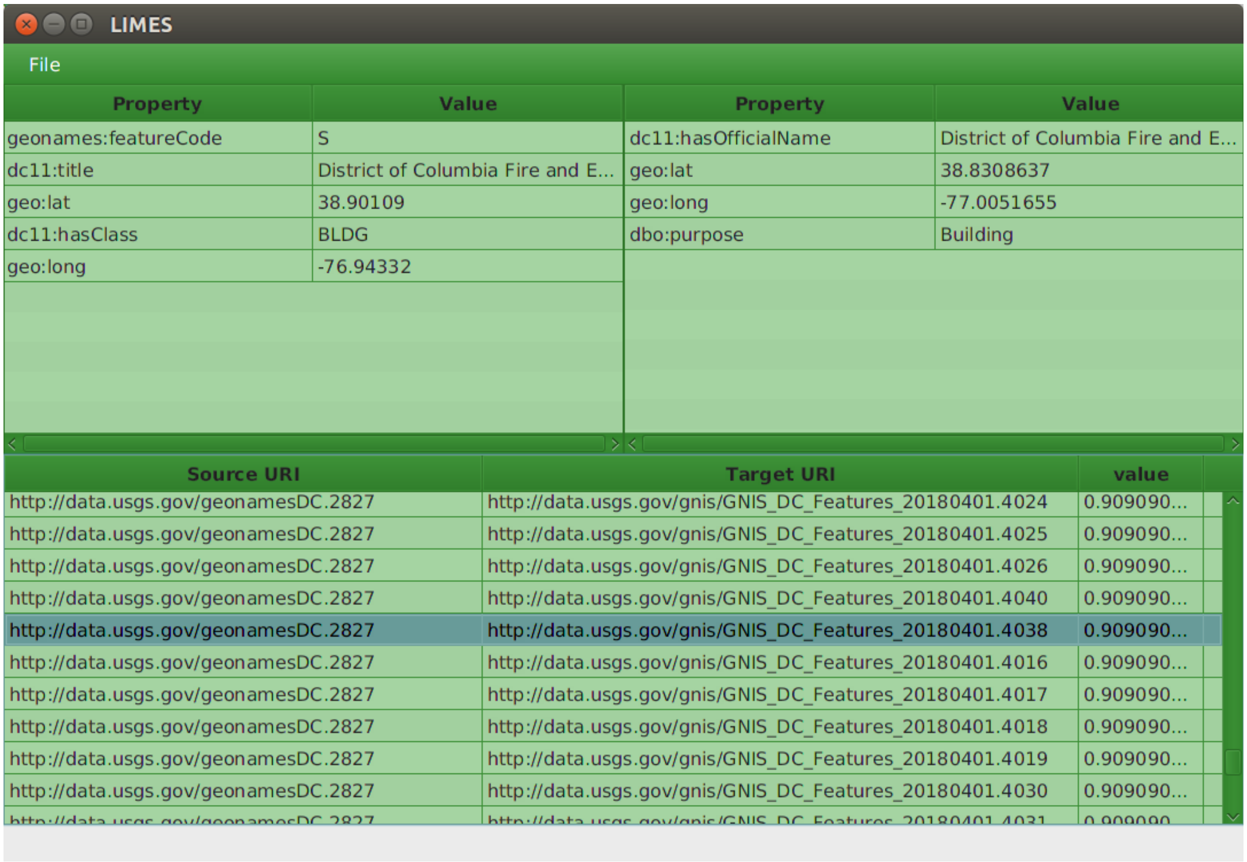
y.dc11:hasOfficialName

Acceptance threshold: 0.9
Verification threshold: 0.7

4. **Execute the Link Specification:**

(a) On the toolbar near the top press the ">" button. This will execute the matching process.
(b) Once the calculations are complete, a form displaying the co-references (near the bottom in a grid) is displayed.

(c) The value on the right is the output confidence value. Values over 0.9 in this case will be asserted as owl:sameAs triples in the output.
(d) Click on any of the records in the grid to see the attributes and values. This is a good way to sample the results and validate if comparison calculations are working or not.
(e) The form/window should look similar to the following image:



Property	Value	Property	Value
geonames:featureCode	S	dc11:hasOfficialName	District of Columbia Fire and E...
dc11:title	District of Columbia Fire and E...	geo:lat	38.8308637
geo:lat	38.90109	geo:long	-77.0051655
dc11:hasClass	BLDG	dbo:purpose	Building
geo:long	-76.94332		

Source URI	Target URI	value
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4024	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4025	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4026	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4040	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4038	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4016	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4017	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4018	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4019	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4030	0.909090...
http://data.usgs.gov/geonamesDC.2827	http://data.usgs.gov/gnis/GNIS_DC_Features_20180401.4031	0.909090

NOTE: This Link Specification generated 5009 owl:sameAs coreferences.

5. Iterate and enhance the Link Specification to produce better results:

(a) Performing only name matching can yield many false positives and false negatives. To showcase an example from the previous matching:

geonames.4016:

"District of Columbia Fire and Emergency Medical Services Engine Company **6**"

gnis.2827:

"District of Columbia Fire and Emergency Medical Services Engine Company **27**"

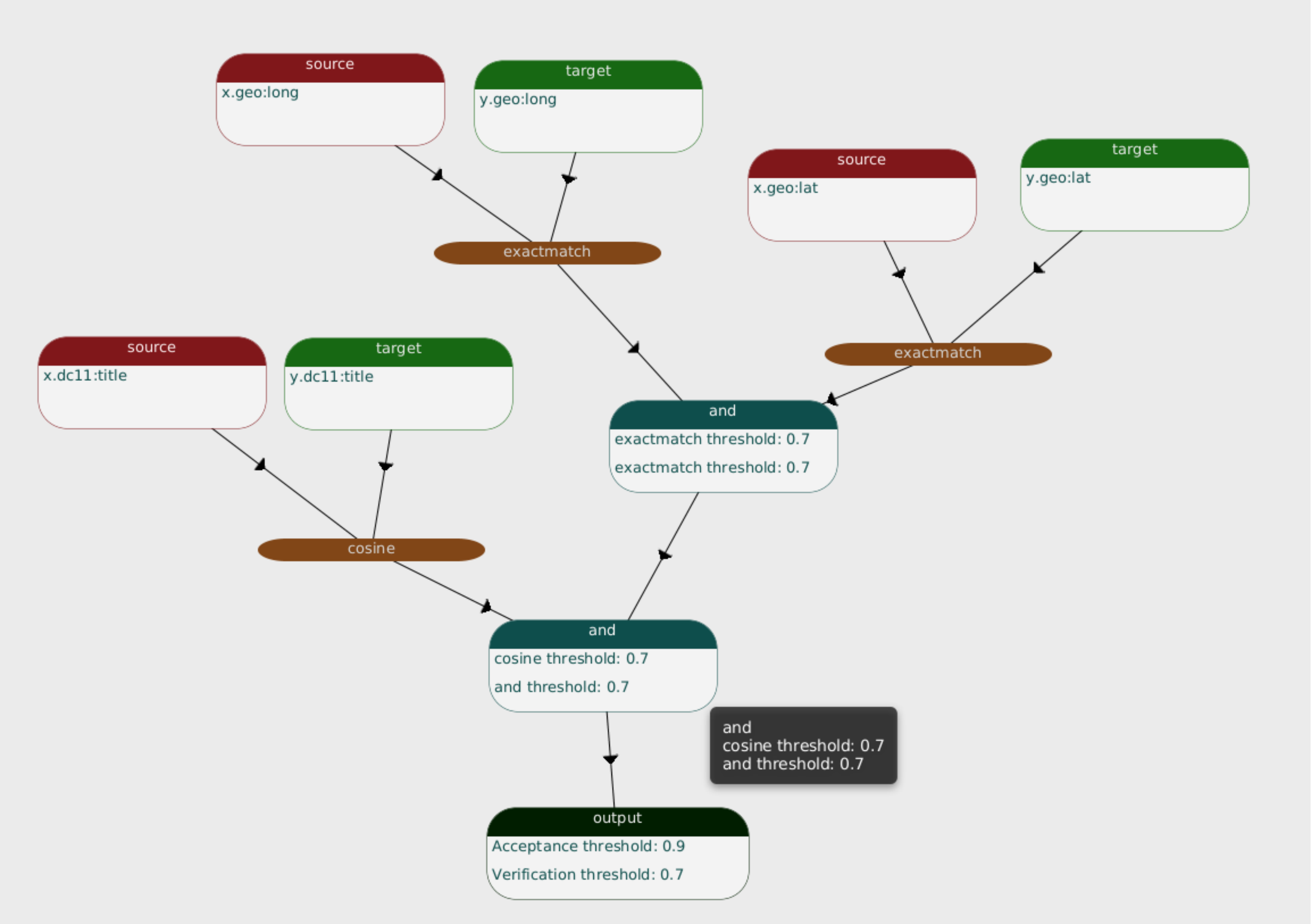
(b) When looking at the available attribution, both are asserted as a “Building” which wouldn’t help in this case, but potentially would help in other comparisons. However, for this prototype, the feature type mappings between Geonames.org and USGS GNIS will take additional effort to complete. These mappings could be performed upstream during the Karma mapping process so that both the source and target feature types are normalized and can then subsequently be compared via LIMES.

(c) For this case (and in many cases), the geographic coordinates can help discern whether a pair of entities are co-references.

(d) Normally, for simple (separate) WGS84 latitude and longitude coordinate fields, a simple Euclidean distance measure can be used to compare if two geographic features have coordinates within a given threshold. However, it’s unknown yet how to implement the Euclidean distance measure available in LIMES.

(e) However, an even simpler technique to compare lat/lon coordinates is to truncate the coordinates to a given precision. For comparing placenames in Washington D.C. it was chosen to truncate the coordinates to 0.001 degree and then do an exact string match. This equates roughly to about 100 meters precision. For this context it is sufficient.
For other datasets it may need to be adjusted. A side benefit of this technique is that exact string matching is very fast computationally.

(f) One way to create the truncated coordinates is to perform the truncation during the Karma schema mapping process. This is the technique used here. To do this in Karma, click on the column drop down and select “Python Transform”. Then create a new column using the “get_coord3()” Python function provided in KARMA_HOME/python/usgs_utils.py. Once the R2RML file has been produced, generate the RDF using curl as specified in the section “Generate RDF from Geoserver Data...” at the beginning of this document.



Steps (g) – (k) below to generate a Link Specification that includes geo coordinate comparisons.

(g) Next, to enhance the Link Specification in LIMES, begin by dragging and dropping both the source and target geo:lat properties onto the workspace. Do the same for the geo:lon properties from both the source and target.

(h) Then drag and drop an “Exact Match” Measure near the geo:lat properties on the workspace. Repeat this process for the geo:lon properties as well.

(i) Next, connect the source and target geo:lat fields to the “Exact Match” Measure (right click → Link To). Do the same for the source and target geo:lon fields.

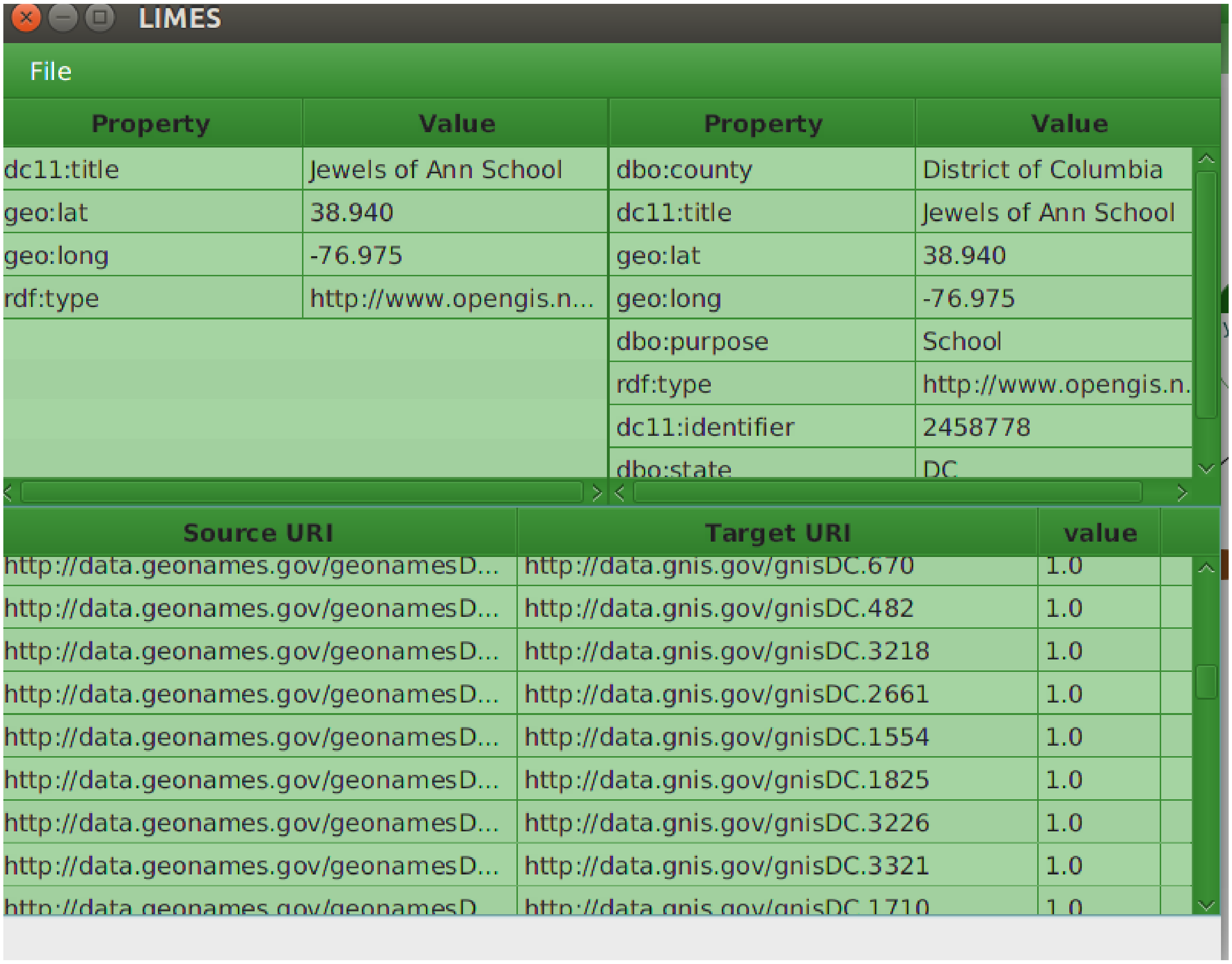
(j) Then drag and drop an “And” Operator. Connect each of the “Exact Match” Measures from the previous step to this “And” Operator.

(k) Drag and drop another “And” Operator onto the workspace. Connect the “And” Operator from the previous step. Also connect the “Cosine” Measure that was used to compare the source and target “dc11:title” properties.

(l) Save the configuration by clicking on the floppy disk in the top toolbar.

(m) Execute the Link Specification by clicking on the “>” button also on the top toolbar.

(n) This time the Link Specification not only took into account “name matching” via the cosine measure, but also geospatial coordinate distance similarity. As seen in the image below, not only does the titles match between the two entities, but also the latitude and longitude values of the source and target entities. See the image below:



Property	Value
dc11:title	Jewels of Ann School
geo:lat	38.940
geo:long	-76.975
rdf:type	http://www.opengis.n...

Property	Value
dbo:county	District of Columbia
dc11:title	Jewels of Ann School
geo:lat	38.940
geo:long	-76.975
dbo:purpose	School
rdf:type	http://www.opengis.n.
dc11:identifier	2458778
dbo:state	DC

Source URI	Target URI	value
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.670	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.482	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.3218	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.2661	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.1554	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.1825	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.3226	1.0
http://data.geonames.gov/geonamesD...	http://data.gnis.gov/gnisDC.3321	1.0
http://data.geonames.gov/geonamesD	http://data.gnis.gov/gnisDC.1710	1.0

(o) Lastly, save the coreferences as a .TTL file by clicking on File → Save Results near the top of the form.

(p) Note, entity resolution is a very iterative process and may take several iterations to fine tune the Link Specification to the desired precision and recall desired.

6. **Load the coreferences into Marmotta**

(a) Now that co-references have been generated as owl:sameAs triples, the co-references can be inserted into Marmotta.

(b) Go to the Marmotta UI (http://localhost:8080/marmotta) and login if needed.

(c) Click on "Core Services". Then click on "Import".

(d) Click on "File" within the Import form and select the co-references RDF generated in the previous step 5(p).

(e) Select Relation = "meta".
(f) Select Mime = "text/turtle".
(g) Select Context = "new".

(h) Set the defined context (i.e. - named graph) URL to be:

http://localhost:8080/marmotta/context/Geonames-GNIS

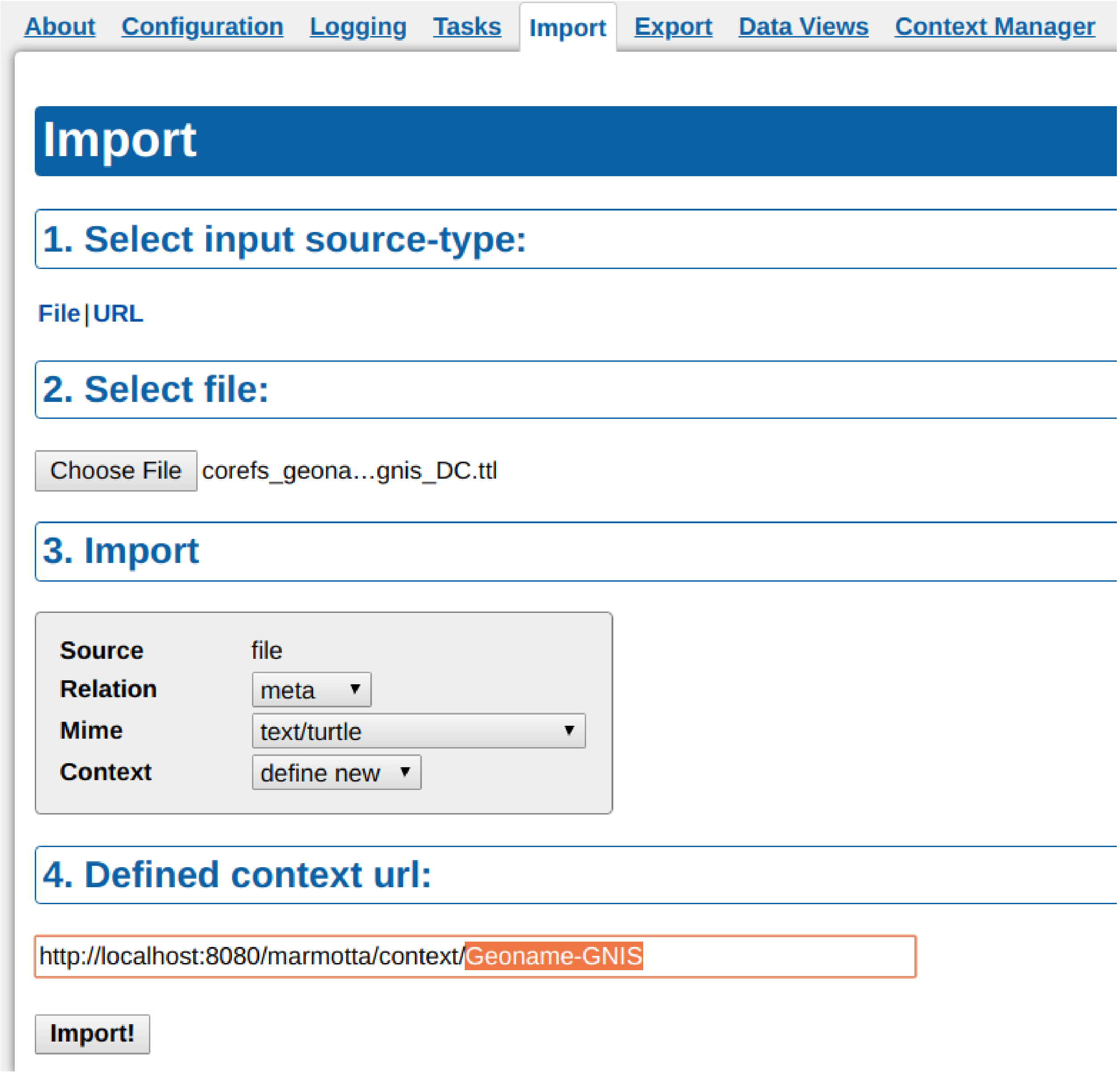
IMPORTANT: This naming scheme is important for the Leaflet UI to work. This is the simplest way to detect which dataset a coreference is coming from (without adding additional triples to Marmotta). This is used to produce the name of the dataset displayed at the top of the "tabs" in the Leaflet Map UI.

Therefore, for each coreference file generated between a source and target dataset, use the following context (named graph) naming scheme:

http://localhost:8080/marmotta/context/SOURCE-TARGET

In the example above "Geonames" is the source dataset and "GNIS" is the target dataset. Don't forget the hyphen between the two!

The resulting completed Import form should look like the following image:



Import

1. Select input source-type:

2. Select file:

corefs_geona…gnis_DC.ttl

3. Import

Source	file
Relation	meta
Mime	text/turtle
Context	define new

4. Defined context url:

http://localhost:8080/marmotta/context/Geoname-GNIS

DONE!!!

At this point all configuration for using co-references is complete and can be used in the Advanced Feature Description System.