

07.13.2012

6441 views |



Share



The [NoSQL Zone](#) is sponsored by [Neo4j](#), a major player in the open source graph database space.



See an [intro video](#) or check out a [free O'Reilly book](#) on Graph DB's.

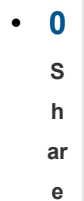
The content of this article was originally written by Tony Hirst on his blog, [OUseful.info](#)



Sometime last week, @mediaczar tipped me off to a neat recipe on the wonderfully named Drunks&Lampposts blog, [Graphing the history of philosophy](#), that uses Gephi to map an influence network in the world of philosophy. The data is based on the extraction of the “influencedBy”



relationship over philosophers referred to in Wikipedia using the machine readable, structured data view of Wikipedia that is DBpedia.



The recipe given hints at how to extract data from DBpedia, tidy it up and then import it into Gephi... but there is a quicker way: the Gephi [Semantic Web Import plugin](#). (If it's not already installed, you can install this plugin via the Tools -> Plugins menu, then look in the Available Plugin.)



To get DBpedia data into Gephi, we need to do three things:

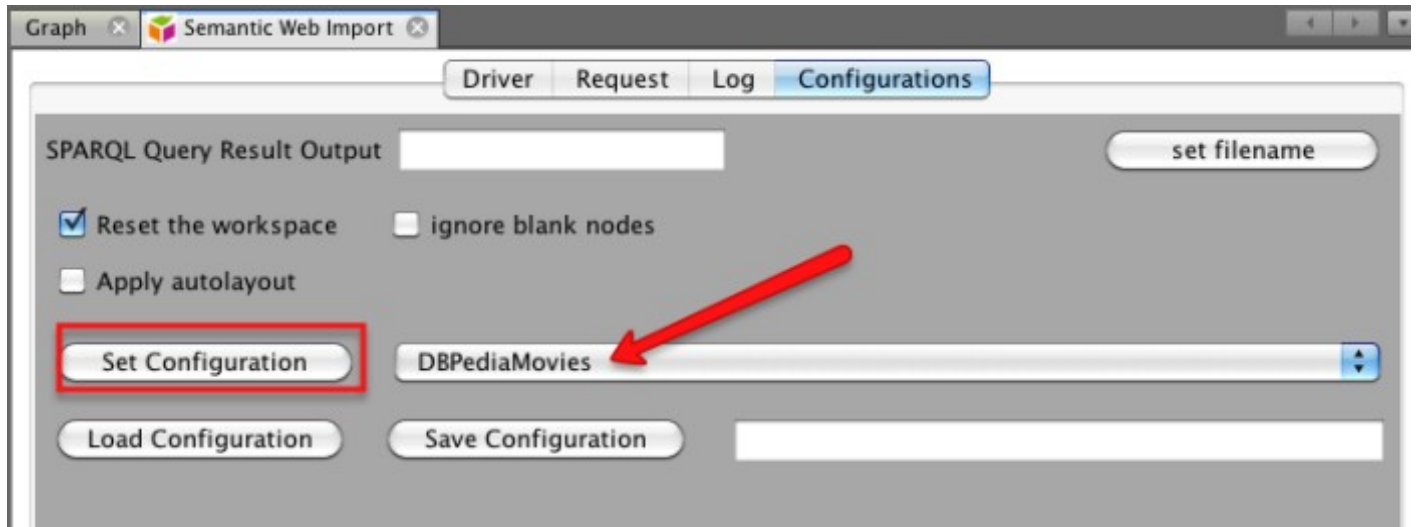
- tell the importer where to find the data by giving it a URL (the “Driver” configuration setting);

- tell the importer what data we want to get back, by specifying what is essentially a database query (the “Request” configuration setting);

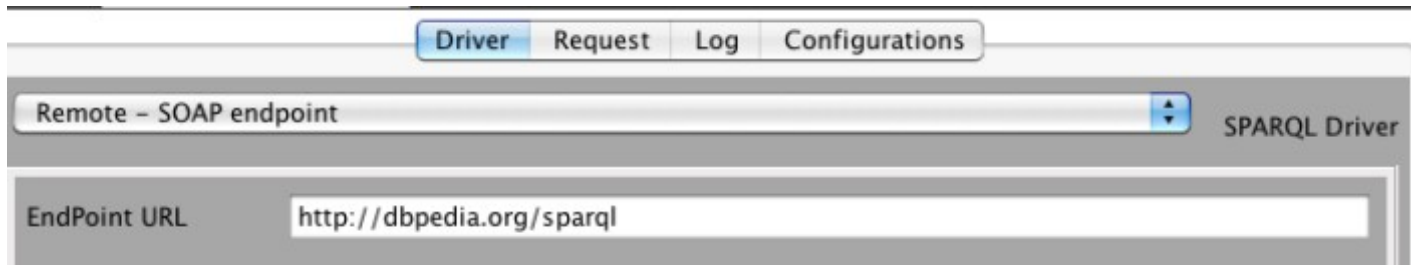
- tell Gephi how to create the network we want to visualise from the data returned from DBpedia (in the context of the “Request” configuration).

Fortunately, we don't have to work out how to do this from scratch - from the Semantic Web ImportConfiguration panel, configure the importer by setting the configuration

to DBpediaMovies.



Hitting “Set Configuration” sets up the Driver (Remote SOAP Endpoint with Endpoint URL <http://dbpedia.org/sparql>):



and provides a dummy, sample query Request:



We need to do some work creating our own query now, but not too much – we can use this DBpediaMovies example and the query given on the Drunks&Lampposts blog as a starting point:



```
1.SELECT *  
2.WHERE {  
3.?p a  
4.<http://dbpedia.org/ontology/Philosopher> .  
5.?p <http://dbpedia.org/ontology/influenced> ?influenced.  
6.}
```

This query essentially says: ‘give me all the pairs of people, (?p, ?influenced), where each person ?p is a philosopher, and each person ?influenced is influenced by ?p’.

We can replace the WHERE part of the query in the Semantic Web Importer with the WHERE part of this query, but what graph do we want to put together in the CONSTRUCT part of the Request?

The graph we are going to visualise will have nodes that are philosophers or the people who influenced them. The edges connecting the nodes will represent that one influenced the other, using a directed line (with an arrow) to show that A influenced B, for example.

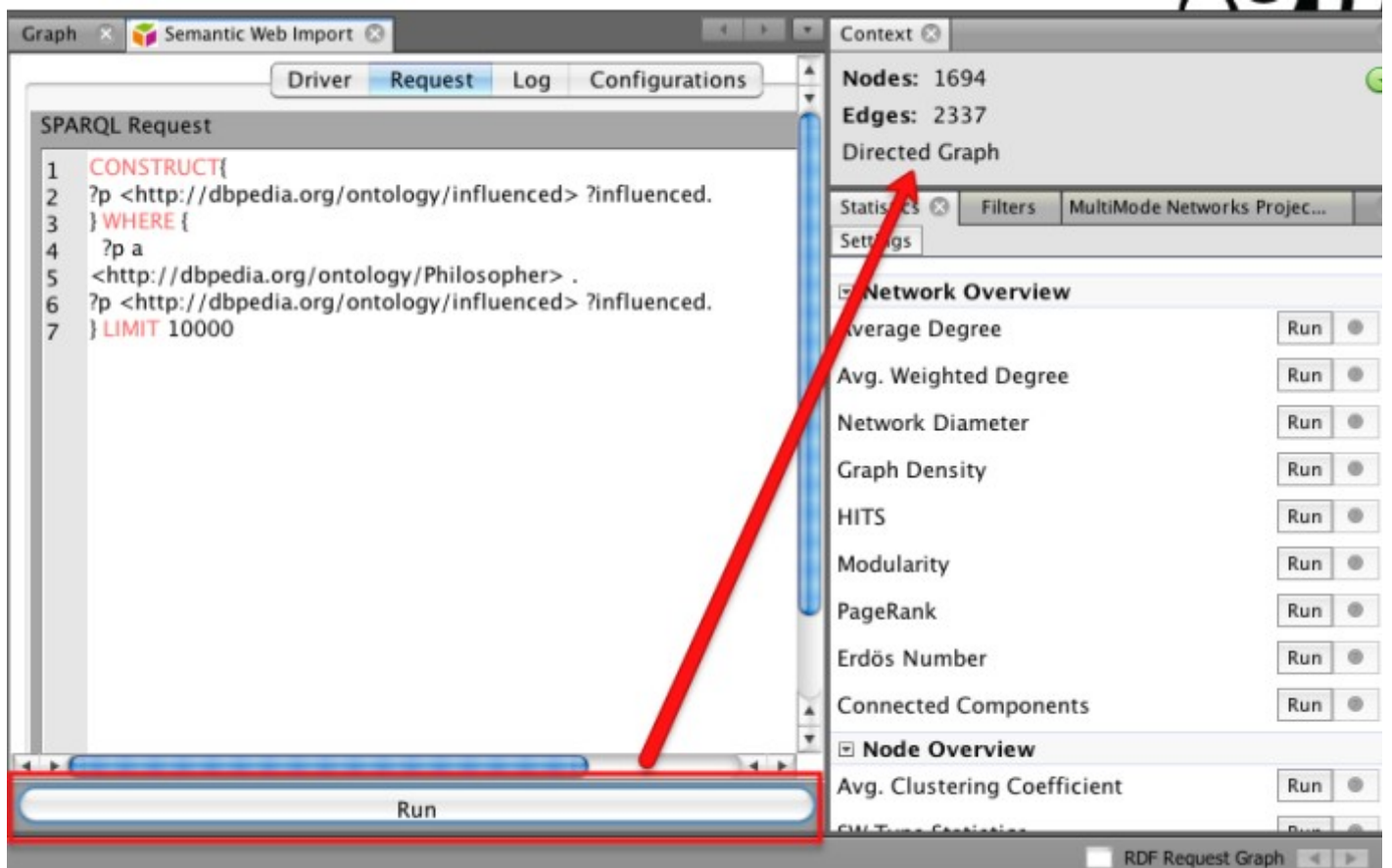
The following construction should achieve this:



```
1.CONSTRUCT{
2.?p <http://dbpedia.org/ontology/influenced> ?influenced.
3.} WHERE {
4.?p a
5.<http://dbpedia.org/ontology/Philosopher> .
6.?p <http://dbpedia.org/ontology/influenced> ?influenced.
7.} LIMIT 10000
```

(The LIMIT argument limits the number of rows of data we're going to get back. It's often good practice to set this quite low when you're trying out a new query!)

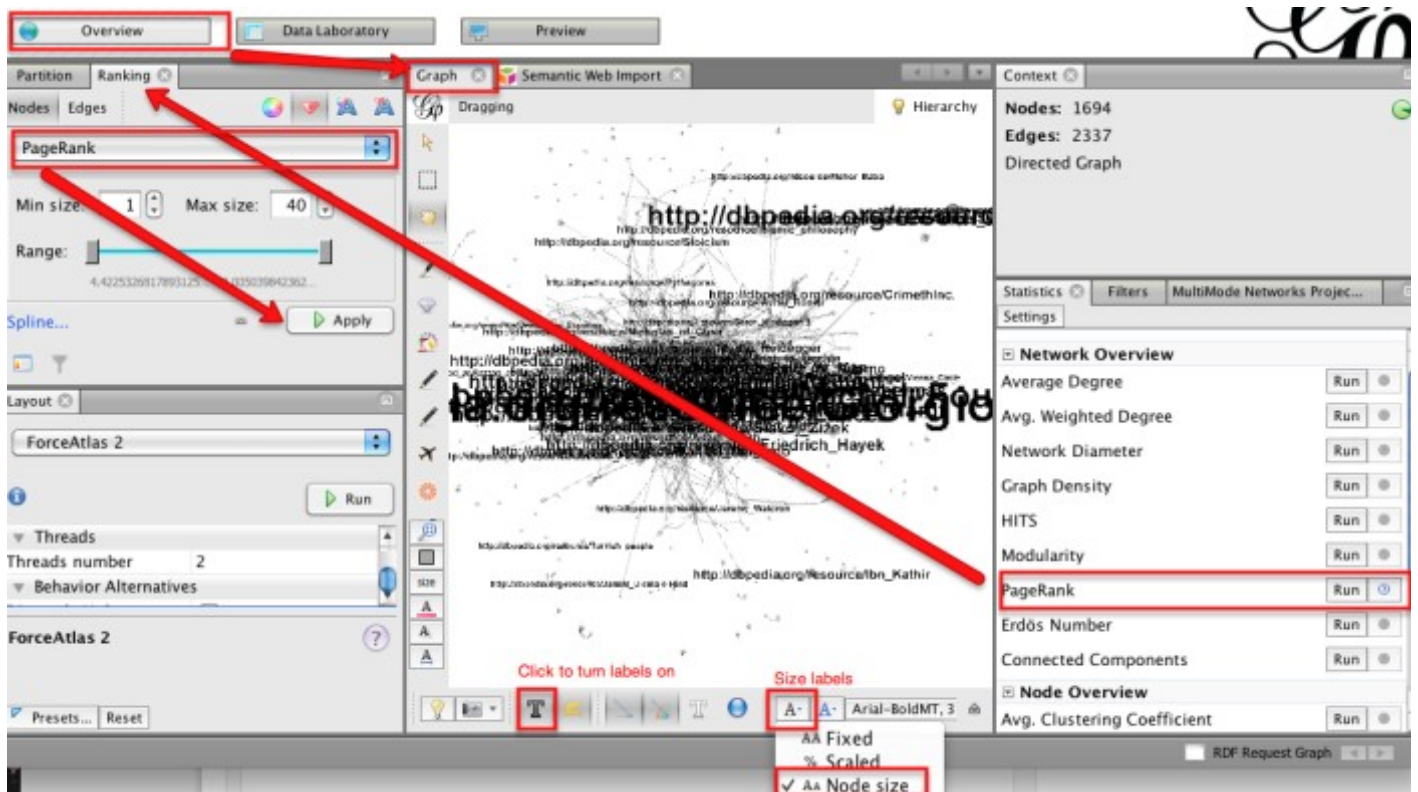
Hit Run and a graph should be imported:



If you click on the Graph panel (in the main Overview view of the Gephi tool), you should see the graph:



If we run the PageRank or EigenVector centrality statistic, size the nodes according to that value, and lay out the graph using a force directed or Fruchterman-Rheingold layout algorithm, we get something like this:



The nodes are labelled in a rather clumsy way

- http://dbpedia.org/page/Martin_Heidegger - for example, but we can tidy this up.

Going to one of the DPpedia pages, such as http://dbpedia.org/page/Martin_Heidegger, we find what else DBpedia knows about this person:



In particular, we see we can get hold of the name of the philosopher using the foaf:nameproperty/relation. If you look back to the original DBpediaMovies example, we can start to pick it apart. It looks as if there are a set of gephi properties we can use to create our network, including a “label” property. Maybe this will help us label our nodes more clearly, using the actual name of a philosopher for example? You may also notice the declaration of a gephi “prefix”, which appears in various constructions (such as gephi:label). Hmm.. Maybe gephi:label is to prefix gephi:<http://gephi.org/> as foaf:name is to something? If we do a web search for the phrase foaf:name prefix, we turn up several results that contain the phrase prefix foaf:<http://xmlns.com/foaf/0.1/>, so maybe we need one of those to get the foaf:name out of DBpedia....?

But how do we get it out? We’ve already seen that we can get the name of a person who was influenced by a philosopher by asking for results where this relation holds: ?p <http://dbpedia.org/ontology/influenced> ?influenced. So it follows we can get the name of a philosopher (?pname) by asking for the foaf:name in the WHEER part of the query:



```
1.?p <foaf:name> ?pname.
```

and then using this name as a label in the CONSTRUCTION:



```
1.?p gephi:label ?pname.
```

We can also do a similar exercise for the person who is influenced.

looking through the DBpedia record, I notice that as well as an influenced relation, there is aninfluencedBy relation (I think this is the one that was actually used in the Drunks&Lampposts blog?). So let’s use that in this final version of the query:



```
01.prefix gephi:<http://gephi.org/>
02.prefix foaf: <http://xmlns.com/foaf/0.1/>
03.CONSTRUCT{
04.?philosopher gephi:label ?philosopherName .
05.?influence gephi:label ?influenceName .
06.?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence
07.} WHERE {
```

```
08.?philosopher a
```

```
09.<http://dbpedia.org/ontology/Philosopher> .
```

```
10.?philosopher <http://dbpedia.org/ontology/influencedBy> ?influence.
```

```
11.?philosopher foaf:name ?philosopherName.
```

```
12.?influence foaf:name ?influenceName.
```

```
13.}
```

If you've already run a query to load in a graph, if you run this query it may appear on top of the previous one, so it's best to clear the workspace first. At the bottom right of the screen is a list of workspaces - click on the RDF Request Graph label to pop up a list of workspaces, and close the RDF Request Graph one by clicking on the x.

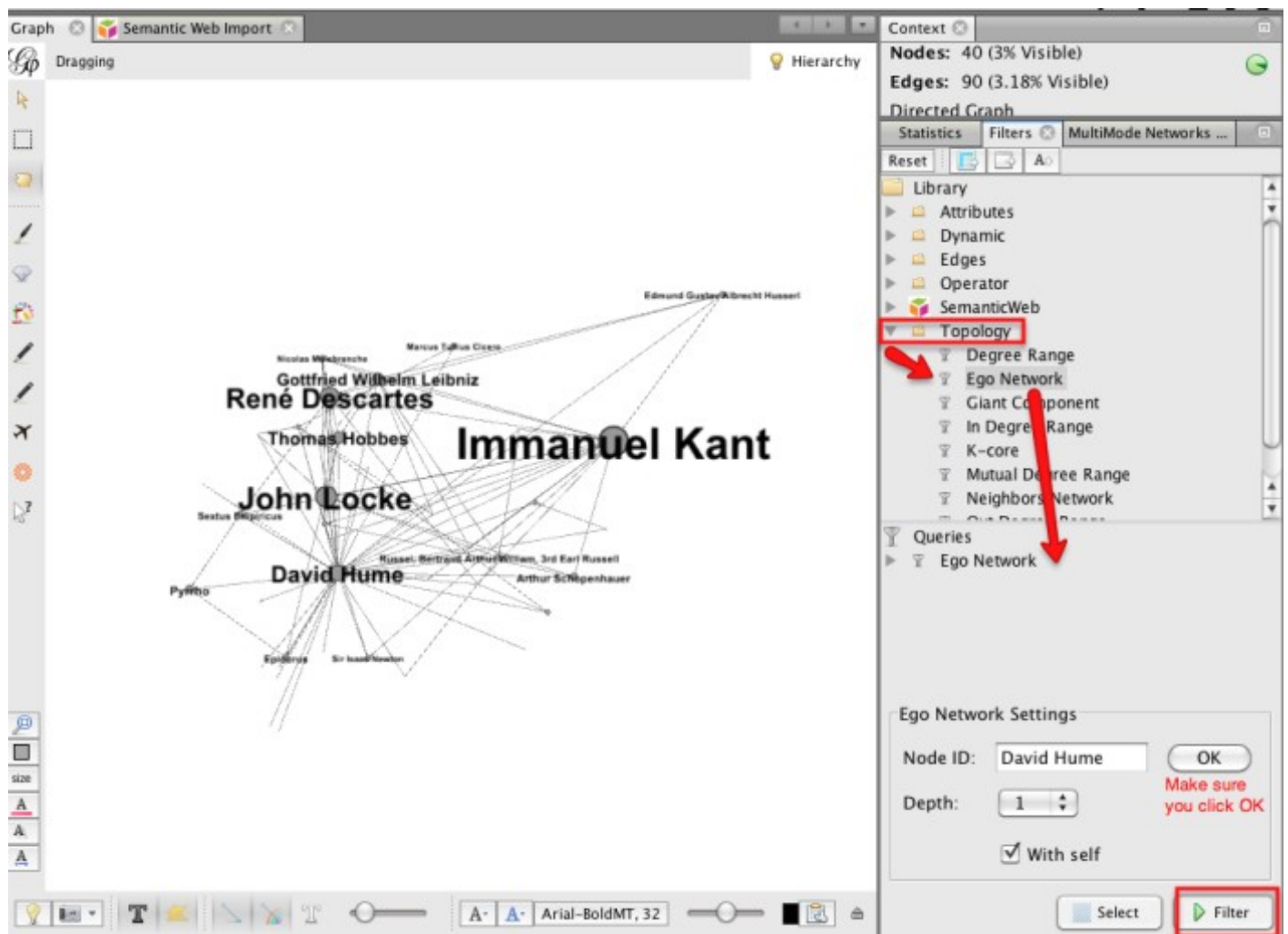


Now run the query into a newly launched, pristine workspace, and play with the graph to your heart's content...:-) [I'll maybe post more on this later - in the meantime, if you're new to Gephi, here are [some Gephi tutorials](#)]

Here's what I get sizing nodes and labels by PageRank, and laying out the graph by using a combination of Force Atlas2, Expansion and Label Adjust (to stop labels overlapping) layout tools:



Using the Ego Network filter, we can then focus on the immediate influence network (influencers and influenced) of an individual philosopher:



What this recipe hopefully shows is how you can directly load data from DBpedia into Gephi. The two tricks you need to learn to do this for other data sets are:

- 1) figuring out how to get data out of DBpedia (the WHERE part of the Request);
- 2) figuring out how to get that data into shape for Gephi (the CONSTRUCT part of the request).

If you come up with any other interesting graphs, please post Request fragments in the comments below:-)

[See also: [Graphing Every* Idea In History](#)]

PS via @sciencebase ([Mapping research on Wikipedia with Wikimaps](#)), there's this related tool: [WikiMaps](#), an online (and desktop?) tool for visualising various Wikipedia powered graphs, such as, erm, Justin Bieber's network...

www.ickn.org/wikimaps/

- Classical Musicians
- Classical- , Modern-Musicians and Bands
- Modern Musicians
- Modern Musicians and Bands
- ✓ People Born in the 80ies
- Justin Bieber
- Dominique Strauss-Kahn
- Lady Gaga
- Barack Obama

Read more about

Zoom-Controls:

Change-Log:

Rafael Nadal
 Richard Gasquet
 Lleyton Hewitt
 Novak Djokovic
 David Nalbandian
 Marat Safin
 Roger Federer
 Andy Roddick
 Tommy Robredo
 Maria Sharapova
 Vera Zvonareva
 Anastasia Myskina
 Alicia Molik
 Dinara Safina
 Elena Dementieva
 Justine Henin
 Serena Williams
 Nadia Petrova
 Venus Williams
 Svetlana Kuznetsova
 Marion Bartoli
 Francesca Schiavone
 Sania Mirza
 Gleesa Dulko
 Flavia Pennetta
 Alicia Keys

Any other related tools out there for constructing and visualising Wikipedia powered network maps? Please add a link via the comments if you know of any...