



# **Foundations of Computer Logic and Symbolic Reasoning**

## **ICS 3102: Assignment**

**Presented By:**

Margaret Mumbi Gichuhi - SCT313-2331/2023

**Date:** December 20, 2024

# Week 3 – Hands On

## *Practical Exercise*

Create dataset statistics using command line tools.

Shell Commands `bzip2`, `grep`, `cut`, `sort`, `uniq`, `wc`

Linux and MacOSX users can install the required commands line tools usually via their favorite package manager.

## Execution

We start with downloading the DBpedia 2015-10 **mappingbased objects dump**. These are the triples linking DBpedia resources extracted from Wikipedia infoboxes. The following questions refer to this dump, not the whole DBpedia dataset! The dumps are serialized as N-Triples, which is perfectly made for line-based processing.

```
wget http://downloads.dbpedia.org/2015-10/core-i18n/en/mappingbased_objects_en.ttl.bz2
```

The dump is compressed, so we can decompress it using the **bzip2** tool. It resembles 2.3 Gb uncompressed, in order to reduce disk usage we pipe it directly to the next tool, e.g. with **wc -l** we can count the lines, i.e. the number of triples. **grep -v '^#'** filters comment lines from the file.

```
bzip2 -dc ./mappingbased_objects_en.ttl.bz2 | grep -v '^#' | wc -l
```

The **grep** tool allows to filter lines by their content, e.g. **grep '<http://dbpedia.org/resource/Berlin>'** selects all triples referencing this resource.

The **cut** tool allows to select parts from each line of the file, e.g. **cut -f 1 -d ' '** will give us the subject of each triple (first field).

Familiarize with the **sort** and **uniq** tool as well. Additionally, you can have a look on **awk** and **sed**.

## Questions

1. How many different properties are used in the dataset?

The *cut* tool with *-f* and *-d* is used to extract the subject, predicate, or object of each triple. The *-f* selects the field number, and the *-d* option specifies the delimiter character. In this case, the fields are separated by spaces, so the *-d ' '* option is used. The *-f 1* option selects the first field, which is the subject of the triple.

The *sort* tool sorts the lines of a text file in alphabetical order, and the *uniq* tool removes the duplicate lines and optionally counts the frequency of each line

```
bzip2 -dc mappingbased_objects_en.ttl.bz2 | cut -f 2 -d ' ' | sort | uniq | wc -l
```

This command extracts the predicate of each triple, sorts them, removes the duplicates, and counts the lines. The result is 662, which means there are 662 different properties used in the dataset

2. Which property is most often used in the dataset?

```
bzip2 -dc mappingbased_objects_en.ttl.bz2 | cut -f 2 -d ' ' | sort | uniq -c | sort -nr | head -n 1
```

This command extracts the predicate of each triple, sorts them, removes the duplicates and counts the frequency of each line, then sorts them in reverse numerical order, and displays the first line. The result is:

1832762 <<http://dbpedia.org/ontology/team>>

# Week 4 – Hands On

## *Practical Exercise*

Set up a local SPARQL endpoint, load a dataset and execute SPARQL queries.

Run Blazegraph

You will need a Java SDK to run Blazegraph.

Download [the latest blazegraph.jar](#) and run it.

```
java -server -Xmx4g -jar blazegraph.jar
```

Once it started, the default workbench location is <http://localhost:9999/blazegraph/>.

Load Data

Same dataset as last week, we use the DBpedia 2015-10 [mappingbased objects dump](#). These are the triples linking DBpedia resources extracted from Wikipedia infoboxes. The following questions refer to this dump, not the whole DBpedia dataset!

For this exercise we only need a subset, which can be filtered using grep.

```
bzip2 -dc mappingbased_objects_en.ttl.bz2 | grep 'ontology/influenced' > mappingbased_objects_en_influenced.ttl
```

You can load the file /path/to/mappingbased\_objects\_en\_influenced.ttl through the "Update" tab:

```
load </path/to/mappingbased_objects_en_influenced.ttl> into graph <http://dbpedia.org/>
```

Now you can run SPARQL queries, such as the ones about Jules Verne from Slide 6 of [Lecture 4.2](#) and from Slide 7 of [Lecture 4.4](#) (they work now, because we use the previous DBpedia dataset release), in the "Query" tab.

#### Question

1. Did Ludwig Wittgenstein ([http://dbpedia.org/resource/Ludwig\\_Wittgenstein](http://dbpedia.org/resource/Ludwig_Wittgenstein)) influence people, who also influenced him?

**Yes, there are multiple results**

```
PREFIX dbo: <http://dbpedia.org/ontology/>
ASK {
  ?person dbo:influenced <http://dbpedia.org/resource/Ludwig_Wittgenstein> .
  <http://dbpedia.org/resource/Ludwig_Wittgenstein> dbo:influenced ?person .
}
```

This query returns true if there is a person who influenced Ludwig Wittgenstein and was also influenced by him, and false otherwise. The answer is true, and one such person is Bertrand Russell.

2. According to the dataset, who did influence most people directly or indirectly?

This query returns the person who influenced the most people directly or indirectly, and the number of people they influenced. The answer is Thales, who influenced 2233 people according to the data.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?person (COUNT(DISTINCT ?influenced) AS ?count) WHERE {
  ?person dbo:influenced+ ?influenced .
} GROUP BY ?person ORDER BY DESC(?count) LIMIT 1
```



⚠ Not secure | 192.168.100.8:9999/blazegraph/#query



Find your program...

Gate Pass Manage...

Building Material M...

Online car rental sy...



Android Attendanc...



Fully Funde

blazegraph workbench

ultra-scalable, high-performance database

WELCOME

QUERY

UPDATE

EXPLORE

NAMESPACES

STATUS

[Wiki - SPARQL Query](#)

```
1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 ASK {
3   ?person dbo:influenced <http://dbpedia.org/resource/Ludwig_Wittgenstein> .
4   <http://dbpedia.org/resource/Ludwig_Wittgenstein> dbo:influenced ?person .
5 }
6
```

[Advanced features](#)

Execute

Clear

true

ultra-scalable, high-performance database from Blazegraph

WELCOME

QUERY

UPDATE

EXPLORE

NAMESPACES

STATUS

PERFORMANCE

[Wiki - SPARQL Query](#)

Namesp

```
1 SELECT ?influencer ?influenced
2 WHERE {
3   <http://dbpedia.org/resource/Ludwig_Wittgenstein> <http://dbpedia.org/ontology/influenced> ?influenced .
4   ?influenced <http://dbpedia.org/ontology/influenced> ?influencer .
5 }
6
```

[Advanced features](#)

Execute

Clear

influencer	influenced
<a href="http://dbpedia.org/resource/Ludwig_Wittgenstein">http://dbpedia.org/resource/Ludwig_Wittgenstein</a>	<a href="http://dbpedia.org/resource/Bertrand_Russell">http://dbpedia.org/resource/Bertrand_Russell</a>
<a href="http://dbpedia.org/resource/Iván_Werning">http://dbpedia.org/resource/Iván_Werning</a>	<a href="http://dbpedia.org/resource/Frank_P._Ramsey">http://dbpedia.org/resource/Frank_P._Ramsey</a>
<a href="http://dbpedia.org/resource/Ludwig_Wittgenstein">http://dbpedia.org/resource/Ludwig_Wittgenstein</a>	<a href="http://dbpedia.org/resource/Frank_P._Ramsey">http://dbpedia.org/resource/Frank_P._Ramsey</a>
<a href="http://dbpedia.org/resource/Susan_Haack">http://dbpedia.org/resource/Susan_Haack</a>	<a href="http://dbpedia.org/resource/Frank_P._Ramsey">http://dbpedia.org/resource/Frank_P._Ramsey</a>
<a href="http://dbpedia.org/resource/Bertrand_Russell">http://dbpedia.org/resource/Bertrand_Russell</a>	<a href="http://dbpedia.org/resource/G._E._Moore">http://dbpedia.org/resource/G._E._Moore</a>

WELCOME

QUERY

UPDATE

EXPLORE

NAMESPACES

STATUS

PERFORMANCE

[Wiki - SPARQL Query](#)Namespace shortcuts: Bigdata ▼

```
1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 SELECT ?person (COUNT(DISTINCT ?influenced) AS ?count) WHERE {
3   ?person dbo:influenced+ ?influenced .
4 } GROUP BY ?person ORDER BY DESC(?count) LIMIT 1
5
```

[Advanced features](#)

Execute

Clear

person	count
<a href="http://dbpedia.org/resource/Thales">http://dbpedia.org/resource/Thales</a>	2233



## Week 5 – Hands On

This quiz is a selftest. You can repeat it as often as you like.

### *Practical Exercise*

Create your own FOAF profile and publish it on the web.

Create your FOAF profile

You can start creating your profile using the [FOAF-a-Matic online tool](#). After filling the forms it generates an RDF document containing the description. Save this description to a file called `foaf.rdf`.

The document is in RDF/XML which is not handy for further manual editing, so we transform it to Turtle syntax using the rapper tool, that we know from [Week 2](#) already. We set the base URI to the future web location of the file. In this case we use `http://www.example.com/people/alice`.

```
rapper -i rdfxml -o turtle ./foaf.rdf http://www.example.com/people/alice > ./foaf.ttl
```

Have a look on the Turtle document and add some more information. You can find a number of additional properties in the [FOAF vocabulary](#). Moreover, there are vocabularies for biographical information (e.g. [BIO](#)), more detailed relationship descriptions (e.g. [RELATIONSHIP](#)), contact information (e.g. [VCARD](#)), project descriptions (e.g. [DOAP](#)), and so on.

If you are in the lucky position to have some webspace, you can publish the file on the web. Remember to use (or to adapt) the base URI from the transformation step, which is defined in the Turtle document. Good practice would be to set up content negotiation, but for the beginning it may suffice to create a symbolic link on the web server:

```
ln -s <path-to-folder>/foaf.ttl <path-to-folder>/alice
```

Once the file is published, you should **validate it** and then we encourage you to share the link in the forum, so that other people can refer to your profile. Also discuss your insights and the challenges in the forum.

### Question

When generating a FOAF profile using the FOAF-a-Matic, it gives you two resources one with the base URI (e.g. <http://www.example.com/people/alice>) and one with a fragment identifier **#me**. What is the difference between the two?

**The FOAF-a-Matic provides you with two resources when you generate a FOAF profile: one with the fragment identification #me and one with the base URI (<http://www.example.com/people/alice>, for example). The fragment identification identifies a particular portion of your profile, namely you, whereas the base URI identifies your FOAF profile as a whole. A hash (#) symbol is used to append the fragment identification to the base URI, creating a URI that leads to a sub-resource inside the main resource. To give an example, the URL <http://www.example.com/people/alice#me> points to the self-descriptive section of your profile. You can link to particular sections of your profile from other documents or sites by using the fragment identifier. The URI <http://www.example.com/people/alice#me/foaf:name>, for instance, points to the property foaf:name of the sub-resource #me of your profile and can be used to connect to your name from another document. Different individuals with the same base URI for their profiles can be distinguished from one another using the fragment identifier. For instance, two individuals with the same name Alice and the base URI <http://www.example.com/people/alice> can distinguish one another using distinct fragment identities, such as #me and #you. By doing this, they can link to their profiles without uncertainty or confusion.**

