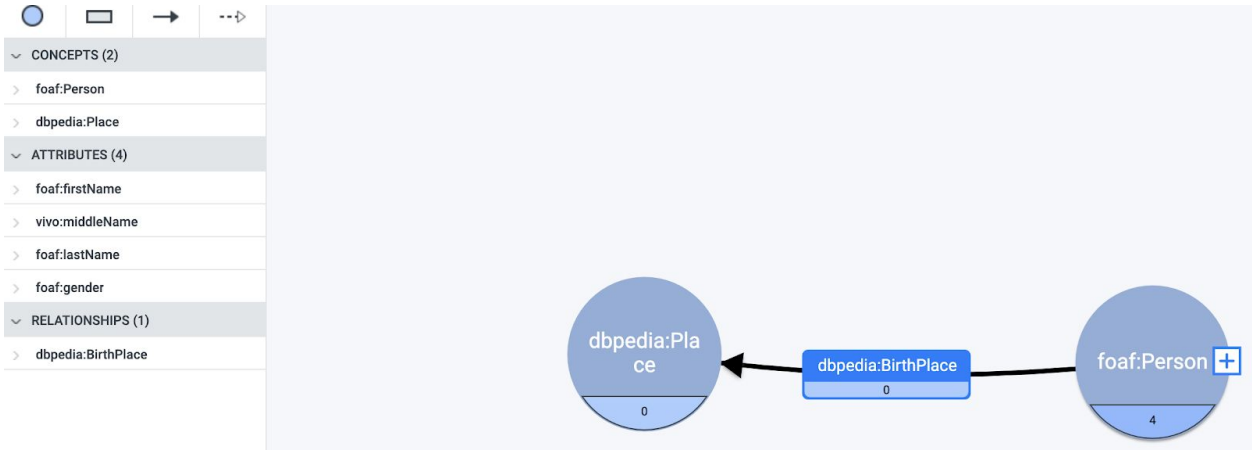
FAIRification tutorial

Adapted from the tutorial: "OpenRefine / FAIRifier walkthrough" by Mark Thompson and the "Group_hands-on_3_FAIRifier_tutorial" by Annika Jacobsen. Further inspired by the "FOAF FAIRification example" by Rajaram Kaliaperumal and "The Big Book of FAIRification" by Erik Schultes et al (https://osf.io/sjzc8/).

Table of contents:

In this tutorial you will use the OpenRefine tool with RDF and WikiData extension to make a dataset about the history of bookkeepers in Leiden more FAIR. You will work on the same data elements and variables that you just constructed a data model of, and take some initial steps to make this dataset more FAIR. For a description of each FAIR principle have a look at https://www.go-fair.org/fair-principles/.

The data model:



1. Load your data

1. Open OpenRefine with RDF and WikiData extension
2. Download the data file to your computer
3. To load your data select “Create Project”, “This Computer” and “Choose Files”.
4. Change the parameter: “Use character “ to enclose cells containing column separators” to ‘ instead; leave other parameters unticked. *NB other input formats, such as Excel, are also possible. See menu left side down “Parse data as”.*
5. Append the ‘Project name’ with your name, so that is is easy to retrieve your project later.
6. Select “Create the project”.
   NB There might be some empty rows when loading the file, but this is no issue.

2. Example: view the different values in a column

You are now in the main UI of OpenRefine where you can access all OpenRefine functions. If at any point the OK button at the bottom is out of the viewing pane, drag the window up or zoom out on your browser window. Let’s first look at some of the basic options.

7. Select a column that contains many different values! How many different values are in that column? An easy way to answer this question is to use the built in facet browsing.
8. Select the pull-down menu for the column you chose.
9. Select “facet -> text facet”. A view will appear on the left giving a summary of occurrence. You can use this view to select only certain values. You can also use this to spot errors in your data and fix them before taking the next step.

3. The first step of your semantic model

In your data each row describes a person. In your model this person is illustrated by a blue circle. This means that the person is an instance and that we need to create a globally unique identifier for each instance of persons in your data. The FAIRifier automatically appends different numbers, for each instance, to the Base URI. However, you can make this more descriptive by appending “Person/” followed by a unique description for each person. In your data this can be found in the “**id**” column.

10. Open the ‘RDF’ plugin options (top right) and select “**Edit RDF skeleton**”. (It should look similar to this screenshot.) Remove the FAIRifier property suggestions by clicking on all the blue X appearing before each property.
11. Select “**(row index) URI**”. Under “Use content ...” select “**id**”. Under “Content used …” select “**URI**”. Under “Use custom expression...” select “Preview edit”. (For reference, see this screenshot.)
12. At the top of the window, from the drop-down box select -> **General Refine Expression Language (GREL)**” and use this expression to build the URL:
    `"Person/" + value`
    Inspect the preview window (it should look like this screenshot).

The URI that you see in the preview uses values from the source data ‘as is’. This can lead to

problems: the source values may not have been intended for machines and may contain special characters ('!', '@', etc.) that can create malformed URIs. The function 'md5(value)' creates a so-called 'hash' from the value. This 'hash' is unique for the value. It is not human readable, but it guarantees that the URI is correctly formed for machines. NB How machines can 'understand' what the URI is for follows in step 16.

13. Now apply md5() function to step 12:
    `"Person/" + md5(value)`
14. Click OK twice to return to the RDF Schema alignment (it should look like this screenshot).
15. We will not actually do this now but in order to make sure your 'do-it-yourself' URI is a globally unique identifier (principle F1), use could use the "edit" option to change the 'Base URI' to a custom domain, like one use by PURL: You then need to make sure that the Base URI ends with a '/'.
    NB Ideally, as a data provider you provide good URIs for your data. For more information on choosing globally unique persistent identifiers, see Identifiers for the 21st century: How to design, provision, and reuse persistent identifiers to maximize utility and impact of life science data.

Next, we will add information so that a machine can interpret the URI correctly. A person (denoted by the URI above) is a type of 'Homo sapiens'. We can use this class URI as a globally unique, machine readable identifier for Homo sapiens: http://purl.obolibrary.org/obo/NCBITaxon_9606. We will next use OpenRefine to apply this to our data to indicate that each person in our data represents an instance of a Homo sapiens.

16. Make sure you are in the 'RDF Schema alignment' window again and select "**Add type**". Then copy-paste the 'Homo sapiens' class URI above. Click 'Add it' (see this screenshot), and 'Apply'. The results looks like this screenshot. *NB Because we did not pre-load any ontology, 'start typing to get suggestions' does not work.*
17. Remove the OpenRefine property suggestions by clicking on all the blue X appearing before each property. To add a human readable label to the instance first select 'add property', select 'property?', type 'rdfs:label', and select the top suggestion.
18. Then select 'Configure?', and under 'Use content...' select 'Constant value' and type 'Person'.
19. Next, select the "**RDF Preview**" tab to see which machine readable relations (subject-predicate-object triples) we just defined (it looks like this screenshot). Note the automatically generated subject URIs for the persons that we defined.

Now machines also know that the persons that you have collected information about are human beings! This may seem trivial, but imagine a world where you could not tell a table apart from a human. Computers cannot do anything fast and correctly across multiple data sources if they do not know what is what. Identifiers alone are not sufficient.
NB knowing that tables are not humans is important, but this is only one type of relationship.

20. Click “**ok**” to save the skeleton and return to the main screen.
21. Note: If at any point you are unhappy about any operations, you can **undo** them by selecting the Undo/Redo tab in the left panel.

4. Extend semantic model with ‘First name’, and ‘Last name’

You will now extend your semantic model by adding name properties describing our person entity.

22. In the ‘RDF Schema alignment’ window click “**property?**”. In the field ‘**Search for property:**’ search for “first”. Select “foaf:firstName”.
23. Select “**Configure?**”. Under “Use content ...” select “**firstName**”. Under “Content used …” select “**Text**”. Under “Use custom expression...” select “Preview edit”. Inspect your choice and click “**ok**” to save the skeleton.
24. In the ‘RDF Schema alignment’ window click “**property?**”. In the field ‘**Search for property:**’ search for “last”. Select “foaf:lastName”.
25. Select “**Configure?**”. Under “Use content ...” select “**lastName**”. Under “Content used …” select “**Text**”. Under “Use custom expression...” select “Preview edit”. Inspect your choice and click “**ok**” to save the skeleton.

5. Extend semantic model with ‘Middle name'

26. In the ‘RDF Schema alignment’ window click “**property?**”. In the field ‘**Search for property:**’ paste the link for middle name from the vivo ontology: “http://vivoweb.org/ontology/core#middleName”. Select “**Add it**” and “**Apply**”.
27. Select “**Configure?**”. Under “Use content ...” select “**secondName**”. Under “Content used …” select “**Text**”. Under “Use custom expression...” select “Preview edit”. Inspect your choice and click “**ok**” to save the skeleton.

7. Extend semantic model with ‘Gender’

Following principle F1, we will now add globally unique identifiers for the genders, Female and Male, in column “Gender”.

42. To provide these globally unique identifiers to the genders go to the “**Gender**” column and in the drop-down menu select “**Facet -> Text facet**”. For **‘Female’** select **‘edit’** and replace the text **‘Female’** with: http://purl.bioontology.org/ontology/SNOMEDCT/703118005.
43. Likewise for **‘Male’** select ‘edit’ and replace the text **‘Male’** with: http://purl.bioontology.org/ontology/SNOMEDCT/703117000

You will now extend your semantic model by creating a triple describing the relation between the person and the gender.

44. To create the triple: ‘person’ ‘has_phenotypic_sex’ ‘gender’ first specify the property by clicking “**add property**” and click on “**property?**” link and in the field ‘**searching for**

property:' you add URI http://purl.obolibrary.org/obo/ERO_0001966. Select "**Add it**" and "**Apply**".

45. Second, specify the object by clicking "**Configure?**" and under "Use content from cell..." select "**Gender**" radio button. Under "The cell's content is used ..." select "**as URI**". Click "**ok**" to close the rdf node dialogue box.
46. Next select the "**Preview**" tab to see which triples we just defined. Note the automatically generated subject URLs for the persons that we defined.
47. Click "**ok**" to save the skeleton and return to the main screen.

8. Extend semantic model with 'Birthplace'

Similar to how we have added a URL for 'Gender' in point 7, we can also add URLs to 'Birthplace'. If we do not want to search for the URLs for places of birth, we can use the inbuilt reconciliation service "Wikidata". To provide these globally unique identifiers to the geographic locations go to the "**PlaceOfBirthName**" column and in the drop-down menu select "**Reconcile -> Start reconciling**". Select the Wikidata service. The service will now start searching for appropriate entities (the results you will get depends on the entity type you have chosen, I recommend the last one: geographic location).

You will now extend your semantic model by describing the relation between the person and birth place. First, you need to create a new instance representing the birth place. Just as with the instance representing the person, this instance is illustrated by a blue circle in the model. Meaning that we need to create a globally unique identifier for the birth place. Further we need to define the relationship between the person and the birth place.

32. To create the triple: 'person' 'has birth place' 'Place', first specify the property by clicking "**add property**" and in the field '**searching for property:**' you add URI for "BirthPlace":http://dbpedia.org/ontology/birthPlace . Select "**Add it**" and "**Apply**".
33. Second, specify the object by clicking "**Configure?**" and under "Use content from cell..." select "**id**" radio button. Under "The cell's content is used ..." select "**as URI**". Under "Use custom expression..." select preview/edit.
34. At the top of the window, from the drop-down box select -> **General Refine Expression Language (GREL)**" and use this expression to build the URL:
"Birthplace/**"** + value.
35. In your model "Place" is specified with the class URL: http://dbpedia.org/ontology/Place Select "**add type**" and paste the 'Place' class URL.
36. To add a human readable label to the instance first select 'add property', select 'property?', type 'rdfs:label', and select the top suggestion.
37. Then select 'Configure?', and under 'Use content from cell...' select 'Constant value' and type 'placeOfBirthName'.

Create the triple describing the relation between the birth place instance and the place (URL).

38. To create the triple: 'birth place' 'has_value' 'URL' first specify the property by clicking "**add property**" and click on "**property?**" link and in the field '**searching for property:**' you add URI http://semanticscience.org/resource/SIO_000300. Select "**Add it**" and "**Apply**".

39. Second, specify the object by clicking “**Configure?**” and under “Use content from cell...” select “**placeOfBirthName**” radio button. Under “The cell’s content is used ...” select “**as text**”. Click “**ok**” to close the rdf node dialogue box.
40. Next select the “**Preview**” tab to see which triples we just defined. Note the automatically generated subject URLs for the persons that we defined.
41. Click “**ok**” to save the skeleton and return to the main screen.