Open Data Pipeline Mockups

v1.1.1

08/Aug/2014

Trento RISE

Open Data Team

David Leoni

Juan Pane

Table of Contents

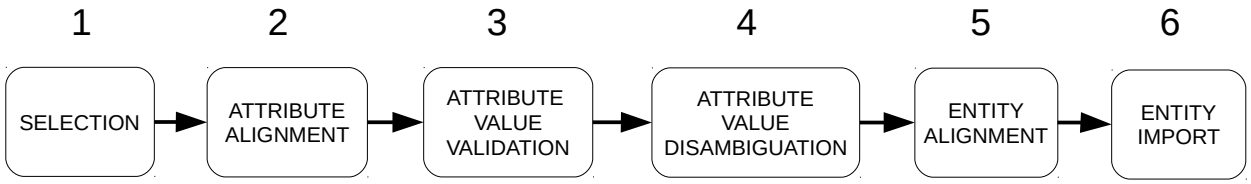
v0.5	19/04/2013		David Leoni
v0.6	26/04/2013	Added schema matching, data validation mockups	David Leoni
v0.8	09/04/2013	• Rewrote examples to snowfields • Added enrichment • Added pipeline steps visual guide	David Leoni
v0.8.6	13/05/2013	• fixed some formatting issue • added single resource CKAN export • reworked xml import problems example	David Leoni
v0.8.7	27/05/2013	• reviewed until Sec 2 Schema matching excluded • added Monitor processes tab • moved Issues to sec 9	David Leoni
v0.8.8	28/05/2013	• Separated rows and records cases in SOTA for XML import	David Leoni
v0.8.9	31/05/2013	• Created Further details section • Moved Issues under Further details • Moved Batch processing under Further details section • Changed and simplified schema import example • Added multiple ckan support to config panel • Added Terminology section • Reviewed Schema Matching section	David Leoni
v0.9	03/06/2013	• Moved state of the art for enrichment to Further details section • Fixed help text in schema matching • Removed mapping from anagrafica_neve/località to URI in schema matching • Reviewed Data Validation section	David Leoni
v0.9.1	04/06/2013	• Reviewed Enrichment step	David Leoni
v0.9.2	05/06/2013	• Reviewed Reconciliation step	David Leoni
v0.9.3	05/06/2013	• Fixed most of Juan comments	David Leoni
v0.9.5	27/06/2013	• Changed XML example into CSV • Added panel to add attributes to EType in schema matching (step 2) • Added Facet for selecting rows with errors in Data Validation step (step 3) • Added Entity disambiguation panel in Enrichment (Step 4) • Added Export step (step 6) • Added Publication step (step 7) • Filled Visualization section with short description	David Leoni

v0.9.6	31/07/2013	• Added title page • Added alternating footers with copyright notice • Marked old changes as accepted • Revised help system, now all help lies on the left sidebar • Removed Open, Export and Help button from all drawings • Translated EntityType names into Italian Selection: • Introduced Date column, shown statistics percentages, added explanation numbers inside selection screen in Selection section 1.1 • Renamed 'step' into 'substep' in Selection mockups • redrew 'manually preprocess resource' 1.3 to show LocalitàTuristica example data Schema Matching: • Corrected example input table • Added *popolazione* to LocalitàTuristica EType • Added Entitytype management panel, Improved Entitytype search in Schema matching section 2.2 • introduced isMandatory attribute and language dropdown menu in attribute definition Drawing 12 Data validation: • Added Refine functionality examples table 3.1 • Fixed Facet for rows with errors • Now user has to manually create target columns for split and merge Enrichment: • fixed initial drawing which displayed wrong pic from previous step Drawing 27 • Better specified meaning of disambiguation links, added description popup wen hovering on links • In entity disambiguation panel Drawing 29: Added Okkam, removed Sindice, added original fields, added mapping feature Reconciliation: • Added initial service choice screen • Changed URI to ID in first column • Added menu to add other id columns Drawing 47 • Transposed table for entity matching, added crowdsourcing button in horizontal reconciliation Drawing 44 • Added unmapped column *popolazione* • Added menus for crowdsourcing “Further details” section 7: • Fixed reference errors	David Leoni

		• updated images to use substep Terminology: • added more definitions	
1.0.0	01/08/2013	• Justified all paragraphs • Fixed white spaces between images and text • Removed page number from first page • Fixed images margins • Turned 'entity matching' into 'identity disambiguation' • Added explanation of attribute weights in unique indexes panel Drawing 13 • Removed references to 'free attributes'	David Leoni
1.1.0	19/June/2014	• changed old odr logo with the cool one • renamed steps • merged step 7 into step 6 • removed annotate.js example • fixed too light gray in dcat selection diagrams • todo update steps 1,2,3,4,5	David Leoni
1.1.1	8 Aug 2014	• Fixed naming of steps • Fixed steps images not being shown in pdf	David Leoni

Introduction

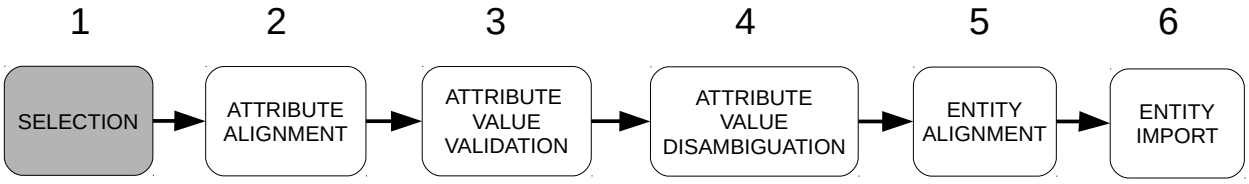
This document contains the user interface mockups for OpenDataRise, a customization of OpenRefine that we will develop to support the open data enrichment pipeline. To produce the required mockups we used JavaFX Scene Builder 1.1[1], a free tool for rapid GUI design which can handle complex layouts. The original files of the mockups will be uploaded to Google Drive and later to the GitHub repository of the project.
There are eight steps in the pipeline, as shown in Drawing 1:



Drawing 1: Pipeline steps

For a detailed description of each step we refer the reader to a separate document[2]. For each step we now describe the proposed mockups.

1 Source Selection

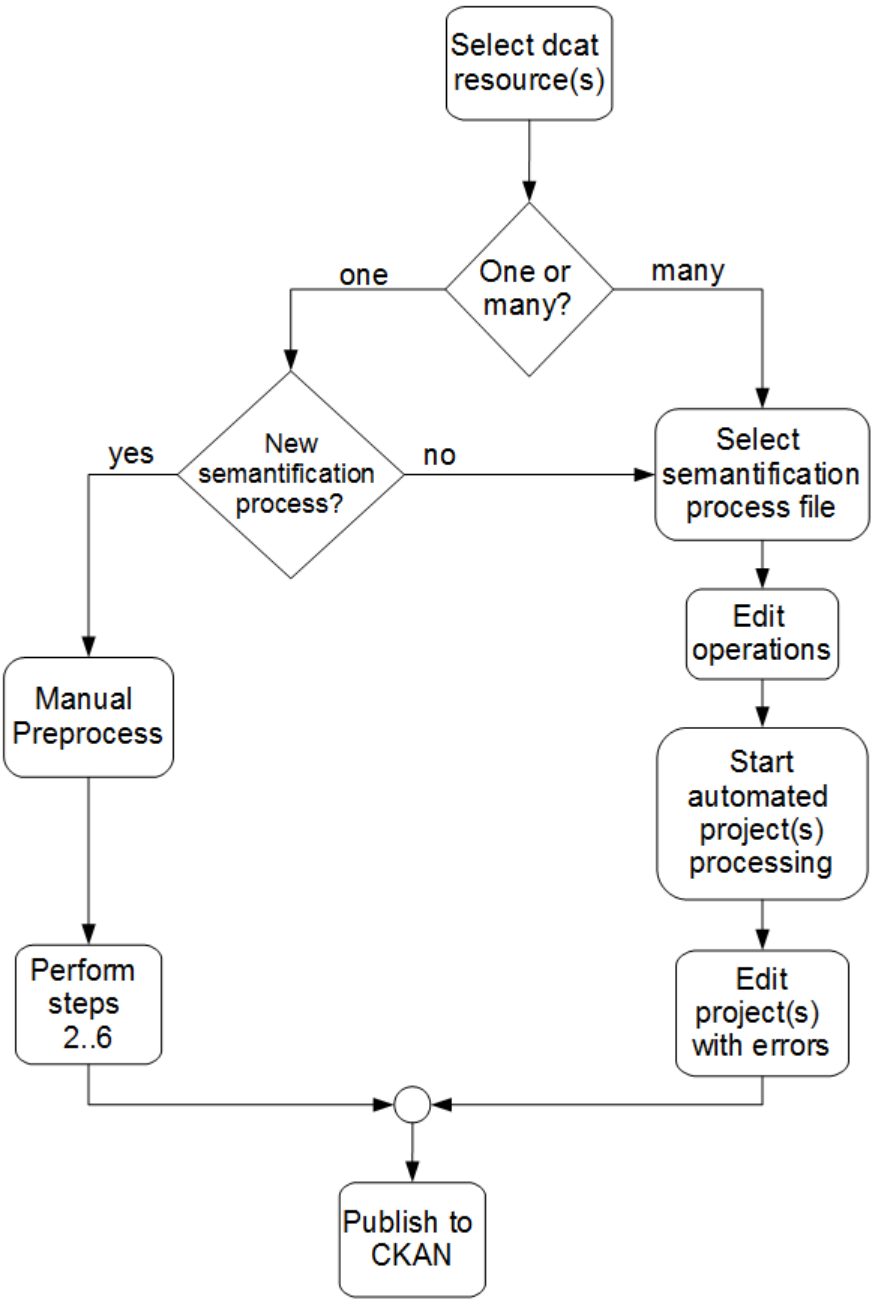


Drawing 2: Step 1 of the pipeline

The first step in the pipeline is Source Selection, as shown in Drawing 2. In order to import dataset resources into OpenRefine, the user first selects a DCat catalogue, and later selects the dataset resources available in the catalogue. For this step we need to clarify a bit the terminology. In OpenRefine, the step of fetching data and preprocessing it is called *Create project*, while the *Import project* function imports a project which was previously exported as an OpenRefine project. Also, a *dataset* is not just a single file. It is a group of possibly many *dataset resources,* which are actually files often in formats like CSV or XML. *F*or example, a *dataset* of "Bollettino Meteo" can have many *resource* files representing the actual data, like "Bollettino Meteo della Val di Non in formato XML", and "Bollettino Meteo della Val d'Adige in formato XML". For these reasons, we chose to allow the user to create a project in OpenRefine out of each dataset resource s/he may select in a DCat catalogue. Optionally, several resources can be processed at once by

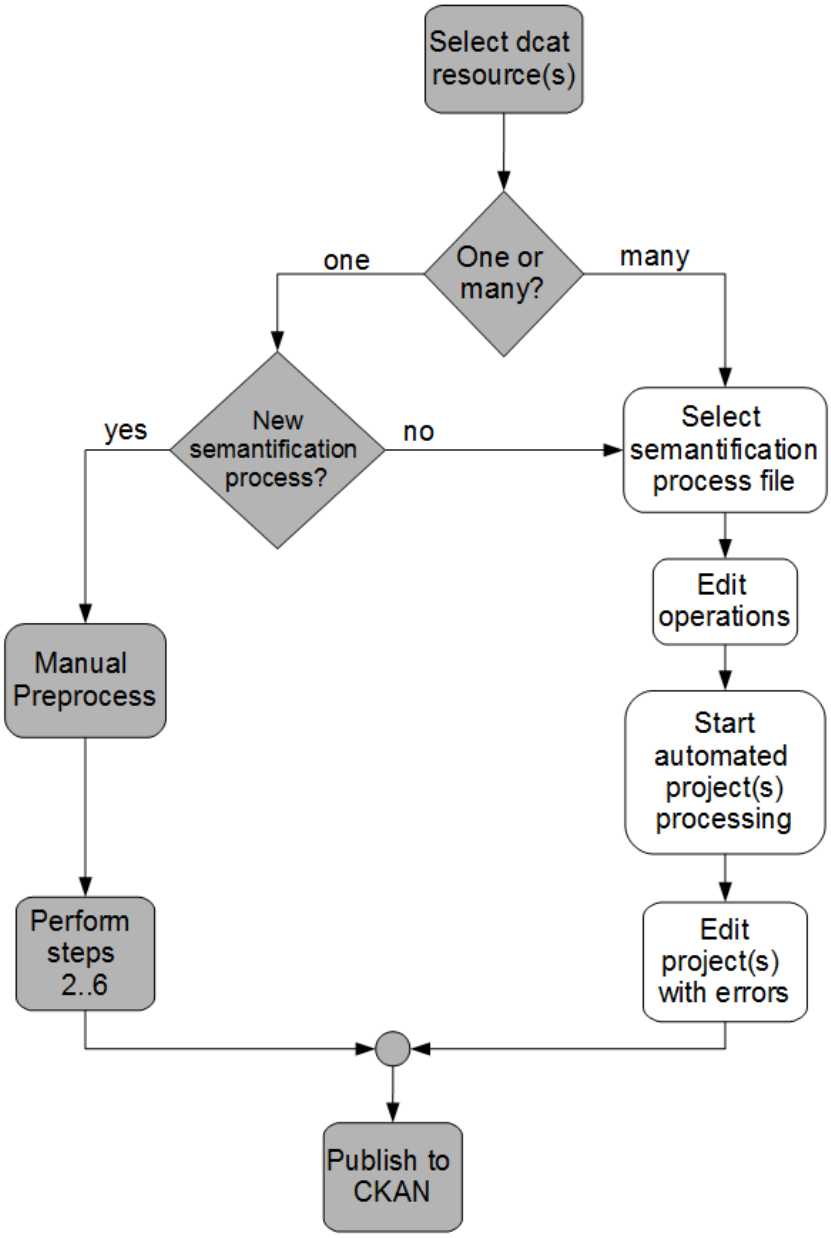
[1] http://www.oracle.com/technetwork/java/javafx/tools/index.html
[2] An Entity Centric Enrichment Pipeline For Open Data https://docs.google.com/document/d/19I3tJtijb5FxNFFNPMhAd8ruUdk-28hTEbLL4MUcLJc

indicating an existing *semantification process file* containing operations to perform on each resource automatically. The workflow is depicted in the following Drawing 3:



Drawing 3: Workflow of dataset resources selection and editing

The simplest flow is given by the left branch, where single dataset resource selection and editing is represented. We detail this flow in the following paragraphs, explaining the more complex batch processing only later in Section 7.1. Screenshots reported hereafter will result with small text as we set required screen size for using Open Refine to 14 inches. It seems hard to proficiently use a spreadsheet with a smaller screen.



Drawing 4: Selection of a single dataset resource from Dcat

During the single dataset resource selection, depicted in Drawing 4, the user must do three substeps:

1.1) Select the resource from the DCat catalogue (as shown in Drawing 6)

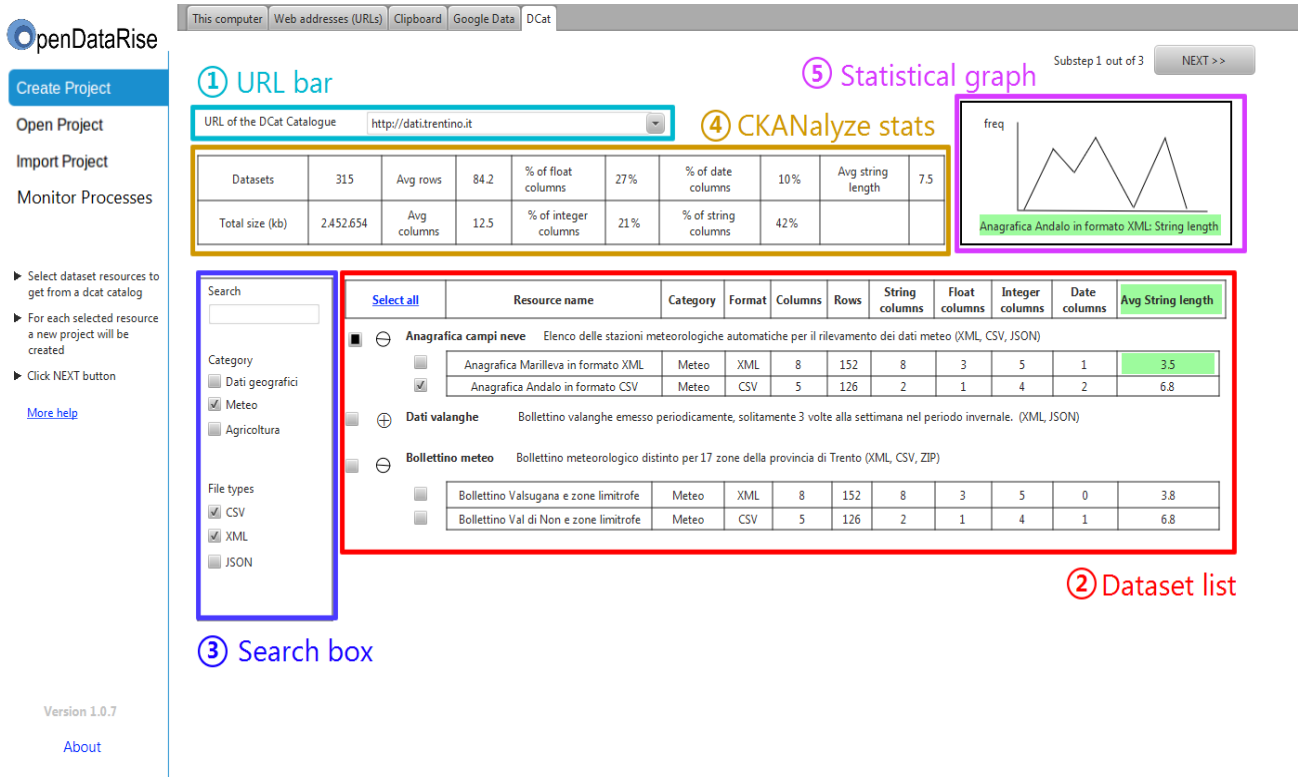
1.2) Choose to start a new semantification process (Drawing 7)

1.3) Manually preprocess the resource (Drawing 8)

We now detail these three substeps.

1.1 Select single DCat resource

For this substep in the first mockup in Drawing 5 we show how the user can select one resource (i.e. a CSV file) from a dataset of a DCAT catalogue. In the upper part of the screen the user can enter the URL of the DCat catalogue (1), in the central part dataset resources can be selected from a dataset list (2), which can be filtered via a search box on the left (3).



Drawing 5: Selection from Dcat screen areas

Statistics on both dataset resources and the whole DCAT catalogue are displayed if the repository has been previously analyzed by means of CKANalyze[1]. Statistics about the single datasets resources are shown in the dataset list (2) while statistics for the whole catalogue (4) are displayed under the URL bar. A statistical graph on the right (5) offers further information about aggregated values. In order to reduce dependencies, the statistics will be displayed only if the CKANalyze plugin is enabled.

[1] https://github.com/opendatatrentino/CKANalyze

As shown in Drawing 6, for the selection the user must do the following:

a) Select Create Project on the left, and DCat tab on the right

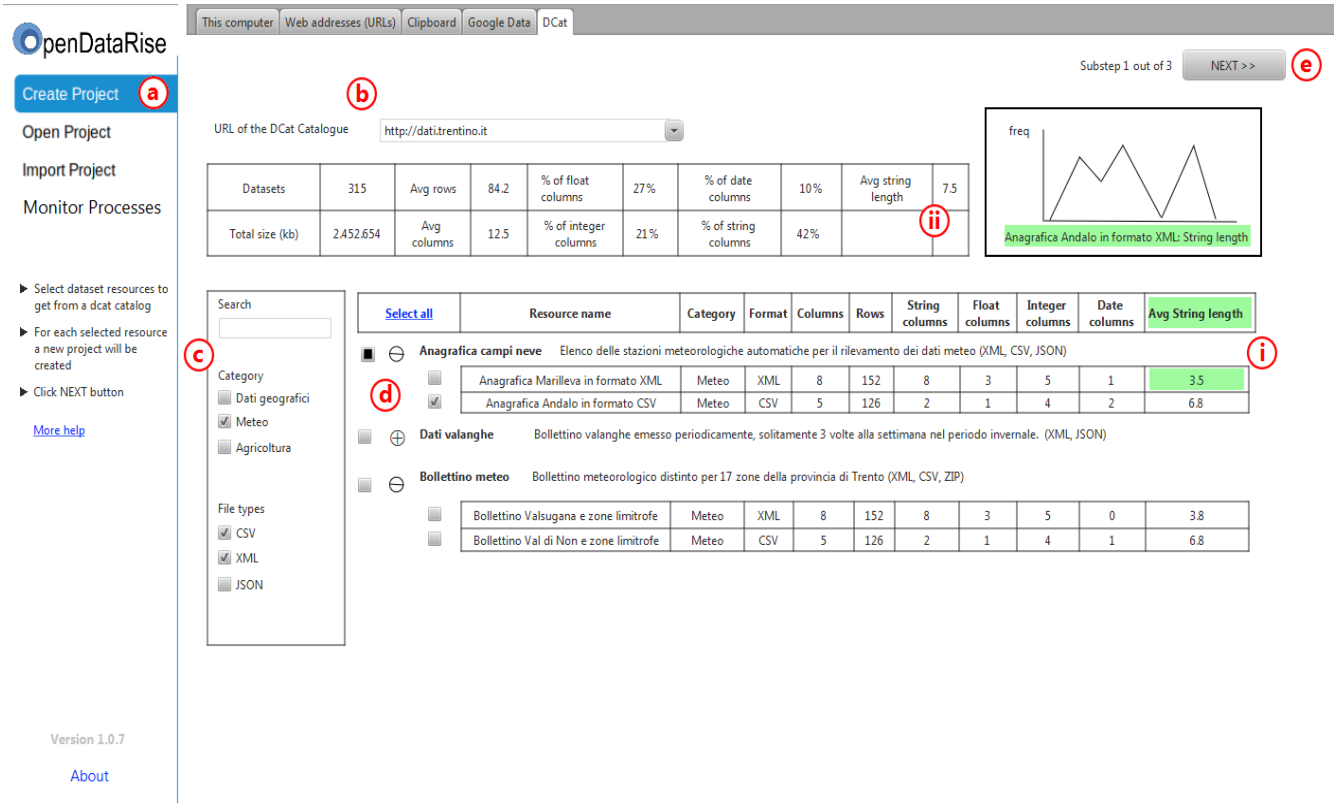
b) Type address of a DCat catalogue.

- By default the system will show the address of the last used DCat catalogue, if available. Addresses which have been used in the past will be available via a dropdown menu.
- If the catalogue has already been analyzed, statistics about it are shown below the catalogue address bar

c) Search among the available datasets and their resources by using the faceted search in the left panel

d) Select the desired resource to open by clicking on the checkboxes to the left of the big central table

e) Click NEXT button in the upper right corner



Drawing 6: Substep 1.1: Single dataset resource selection

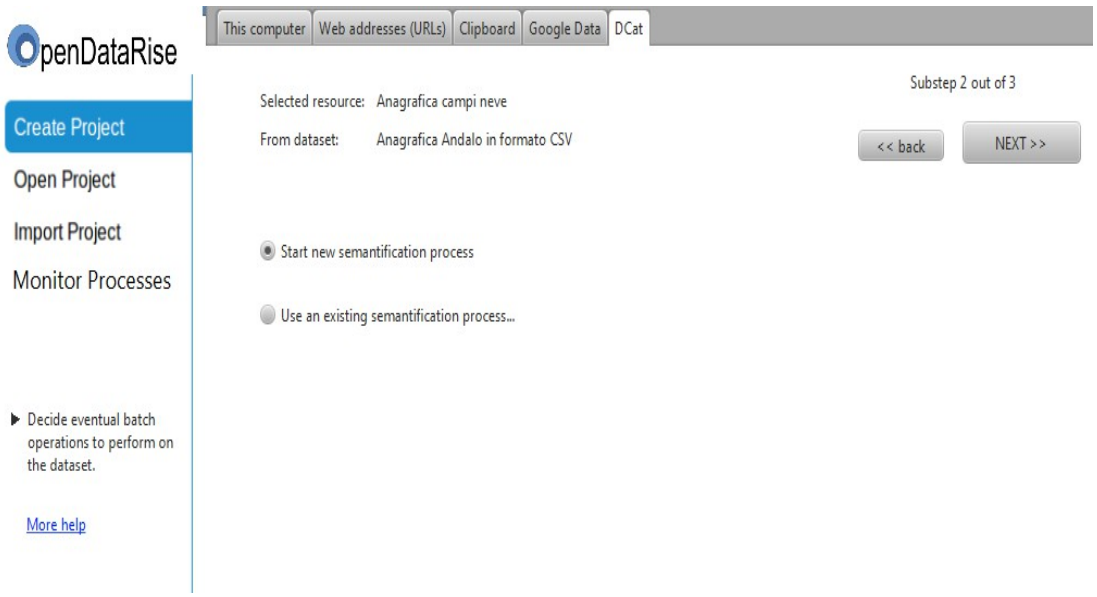
We decided to reserve the area occupied by the big graph in the upper right corner for displaying statistical information according to mouse position in the page. Content will be displayed as described in the following Table 1:

Mouse position	Displayed content in the statistical graph section of Drawing 5
i) a cell of “Avg string length” column	Graph of string distribution of a particular dataset
ii) a cell holding an aggregated value in CKAN repo stats	Graph of distribution of the aggregated value in the whole catalogue

Table 1: Statistical information displayed when hovering on cells in substep 1.1

1.2 Start new semantification process

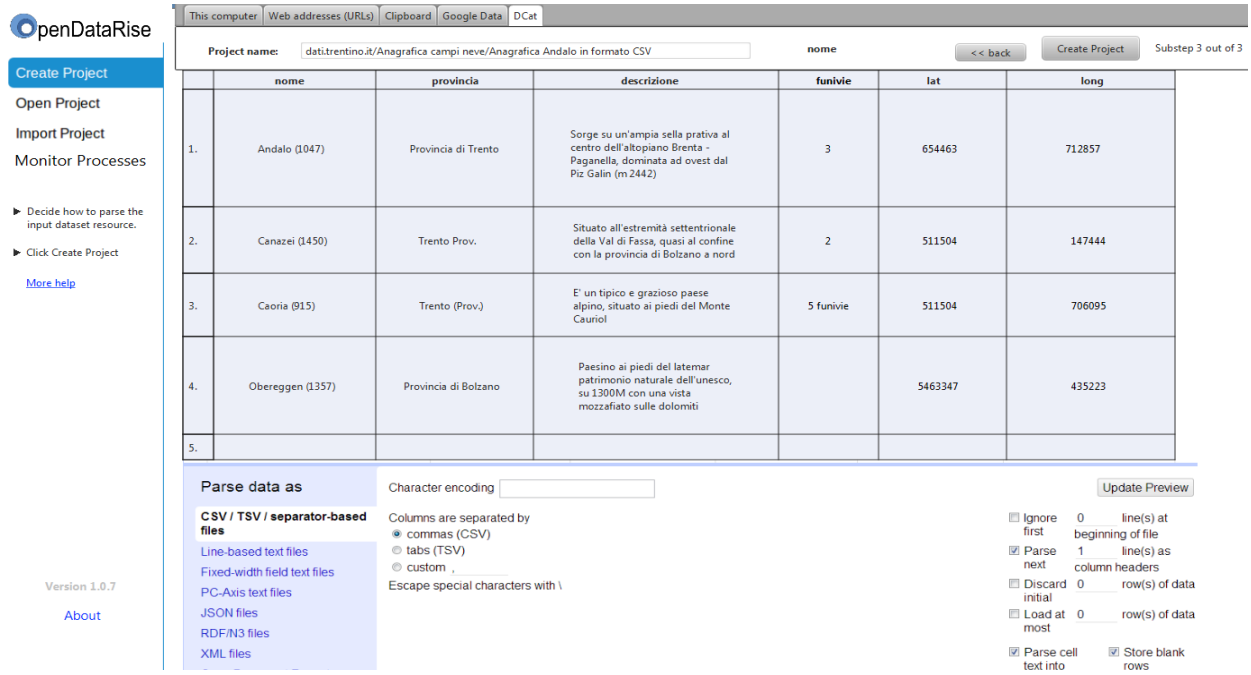
In the next substep shown in Drawing 7, the user chooses to start a new semantification process:



Drawing 7: Substep 1.2: Choice of new semantification process

1.3 Manually preprocess resource

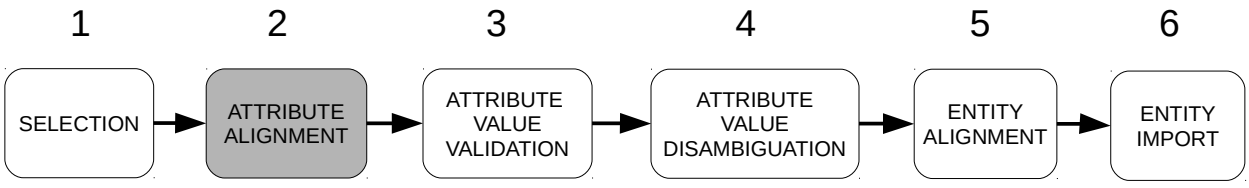
In this substep, depicted in Drawing 8, the user decides how to parse the dataset resource, using standard OpenRefine parse panel (the case for a CSV file is shown):



Drawing 8: Substep 1.3: Manual resource preprocessing

Finally, by clicking on Create Project, the user will be presented with the screen in Drawing 10 to perform the Attribute Alignment step, which we now detail in the following section.

2 Attribute Alignment



Drawing 9: Step 2 of the pipeline

In the Attribute Alignment step the user assigns an entity type to the dataset he selected from Dcat, thus performing a mapping from source fields in the dataset to fields in the target entity type.
In the following Section 2.1 we show an example CSV file to import, while in Section 2.2 we propose a user interface to perform schema matching in OpenDataRise.

2.1 Schema matching example

We provide in Table 2 an example of a CSV file that could be imported in OpenDataRise:

nome	provincia	descrizione	funivie	lat	long
Andalo (1047)	Provincia di Trento	Sorge su un'ampia sella prativa al centro...	3	654463	712857
Canazei (1450)	Trento Prov.	Situato all'estremità settentrionale della...	2	511504	147444

Table 2: Proposed XML to tabular conversion example

The system will have full multilingual support, so hereafter we will use Italian names for attributes. Let's suppose the desired target entity type of name *TourismResort* (LocalitàTuristica, in Italian) is given by the schema depicted in Table 3:

Attribute Name	Attribute Type
name (nome)	String (Stringa)
province (provincia)	City (Città)
height (quota)	Integer (Intero)
coordinate (coordinate)	Coordinates (Coordinate)
description (descrizione)	SemanticString (StringaSemantica)
population (popolazione)	Integer (Intero)

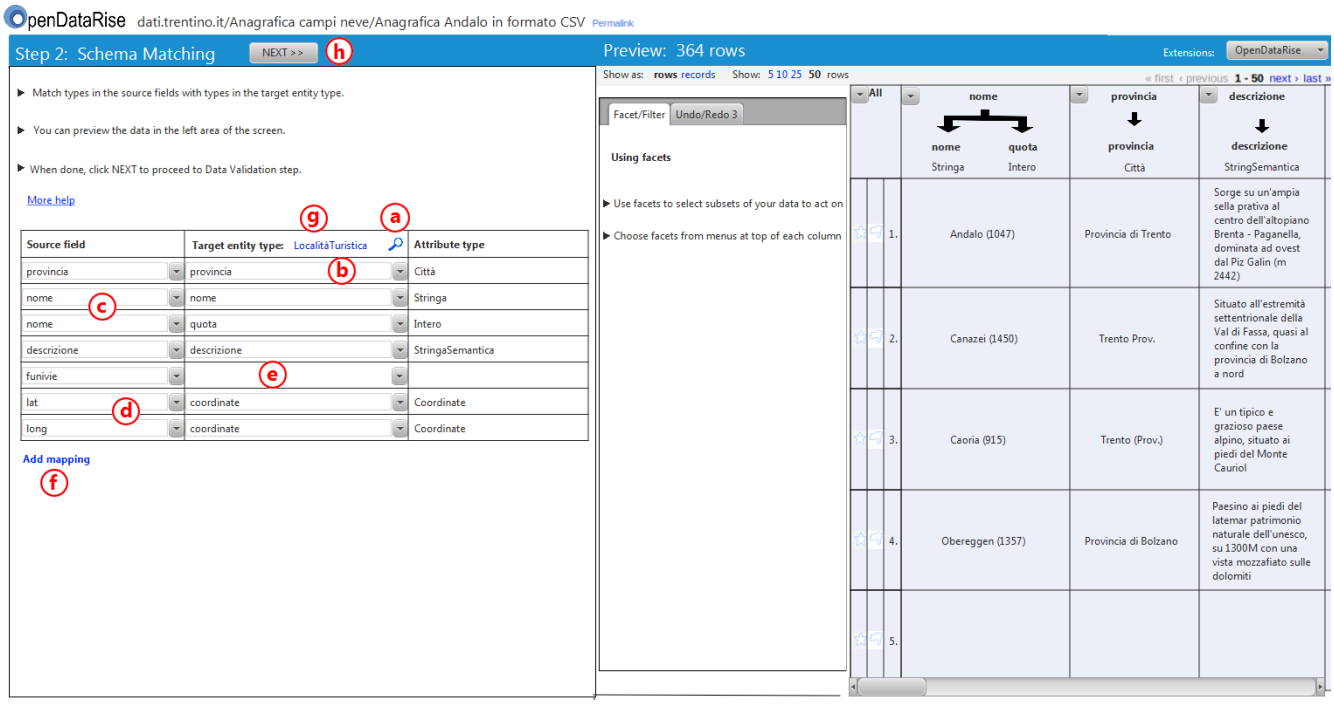
Table 3: Example of target type of entity TourismResort (LocalitàTuristica in Italian). Translation in Italian is given in round brackets

The attributes *nome*, *quota*, *coordinate*, *descrizione* and *popolazione* are associated to datatypes, while *provincia* is a relational attribute. In the final semantified table (after step 4), cells in the *provincia* column will be URIs pointing to entities of type *City*.

2.2 Schema matching interface

In the screen depicted in Drawing 10, the left panel allows the user to perform the schema matching, while in the right panel a preview of the dataset under the new schema is shown. The preview displays the data like in Step 3 (Attribute Value Validation), allowing data navigation via facets like it normally happen in Refine. We defer to the next Section 3 a more detailed description of the preview content.

We have chosen to subdivide the screen this way because for schema matching field names stacked in vertical allow the greatest number of fields to be displayed, while for navigating the data the user can still use the familiar Refine interface on the right.

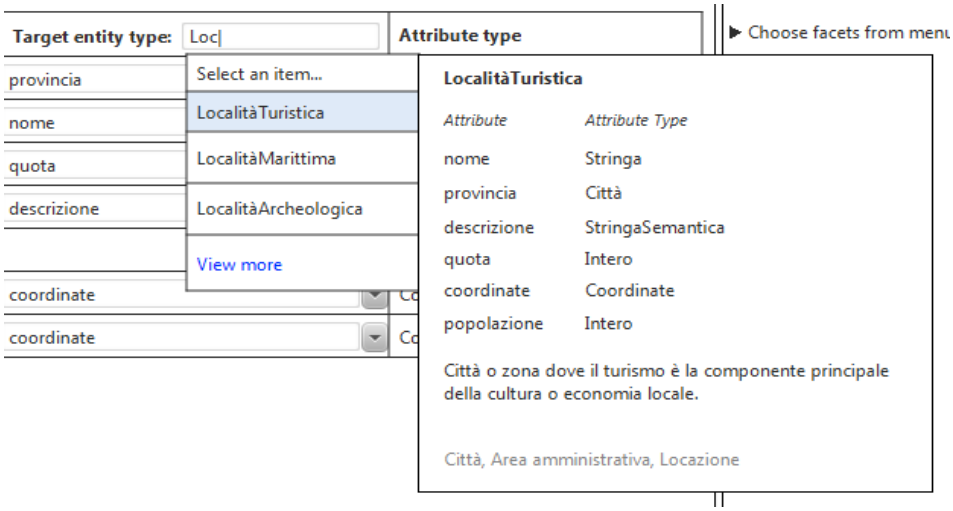


Drawing 10: Attribute Alignment panel

When the interface is presented to the user, in the schema matching panel to the left a mapping to a guessed target entity type is already automatically provided by the system. Source fields that have no counterpart in the target entity type (like *funivie)* aren't associated to any target field.

The user must then do the following:

a) Check the target entity type, and choose another one by clicking on the lens icon in case it is not correct. In this case, a textbox to allow searching entity types will substitute the current entity type name as shown in Drawing 11:



Drawing 11: Entity type selection during Attribute Alignment step

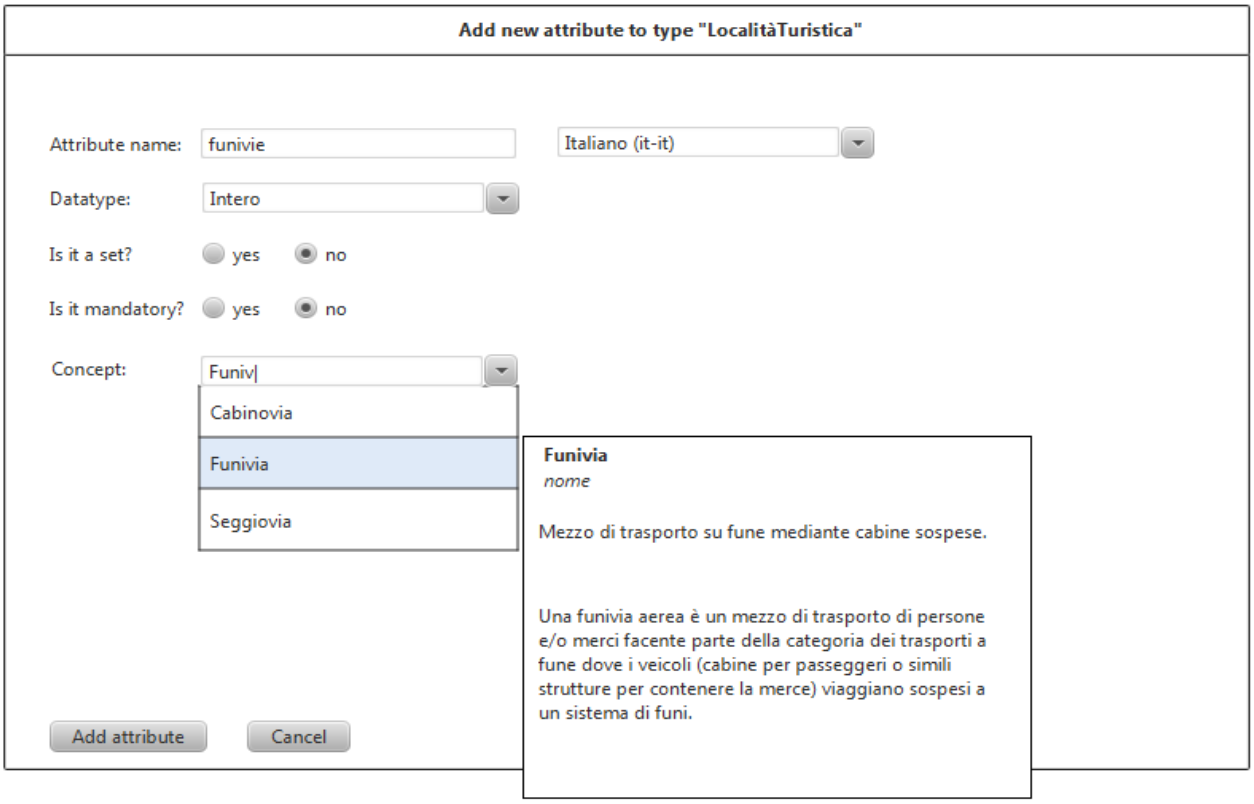
b) Upon selection of a new entity type the system will automatically guess a new match between the source and target fields and the lens icon will reappear to allow new searches.

c) For each target field, check if it is correct, and eventually choose another target field.

d) Eventually map a source field to two or more target fields, like for example happens for the *nome* field which is mapped to *nome* and also to *quota.* Later during the Attribute Value Validation step the user will have to manipulate the data to properly split the input fields to conform to the target attributes.

e) Eventually map two or more source fields to one target field, like for example happens for the *lat* and *long* fields which are mapped to *coordinate.* During the data validation step, the input fields will be merged if needed to conform with the target attribute formatting.

f) Eventually choose to map a source field into a field not present in the target entity type. In this case a new attribute must be added to the target entity type, by clicking a field named <new attribute> in the drop down menu. This operation will open the panel shown in Drawing 12:



Drawing 12: Adding a new attribute to an entity type

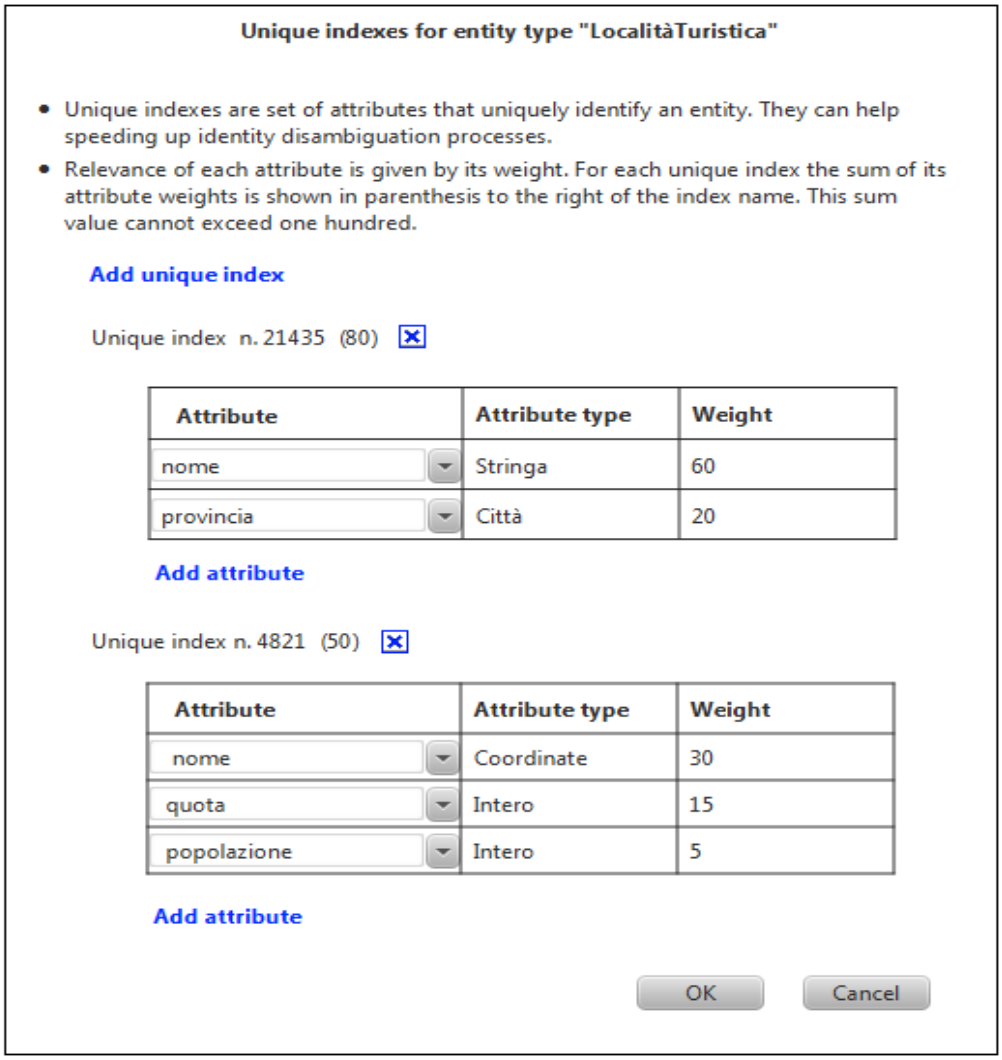
In the panel it is possible to set the name and language for the attribute, the datatype, whether or not it is a set of values and the concept associated to the attribute. All of these values are automatically set by the system when the panel is shown to the user. The user will be then able to modify them if they are not correct. The concept can be either searched by typing its name or selected from a list of possible choices in the dropdown menu. If a concept cannot be found it shall not be possible to add the attribute to the entity type[1].

g) Add mappings by clicking on the *Add mapping* link. Adding mappings will be necessary in case there are one-to-many or many-to-one mappings. It will be possible to remove a mapping by choosing the special label *<delete mapping>* from the dropdown menu in the source field cell

h) Eventually manage the selected entity type by clicking on its name, to display the panel depicted in Drawing 13. For this iteration of the software in the entity type management panel it will only be possible to set unique indexes, which are sets of attributes that uniquely identify an entity. They are useful to speed up identity disambiguation activities. To this end, each attribute is given a weight,

[1] If this is the case, then in the first version of the system the knowledge base will have to be updated by the experts that maintain it.

and for each unique index the sum of its attribute weights is shown in parenthesis to the right of the index name. This sum value cannot exceed one hundred.



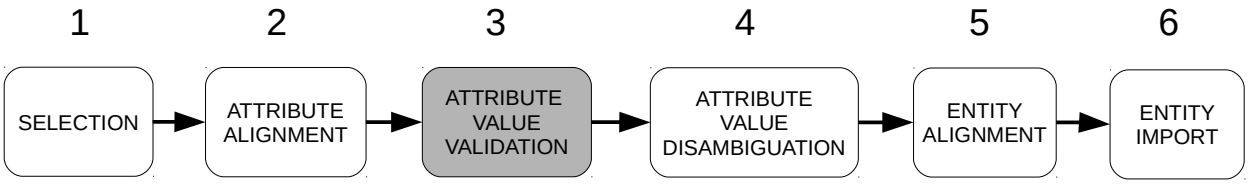
Drawing 13: Panel to manage entity types

i) Click NEXT button to proceed to Attribute Value Validation step.

Now that we have the correspondences between the input columns and the target attributes of the selected entity type, we can proceed to validate the values inside the dataset columns to make sure they conform to the type they were assigned during this step. Original columns which where not mapped will not be

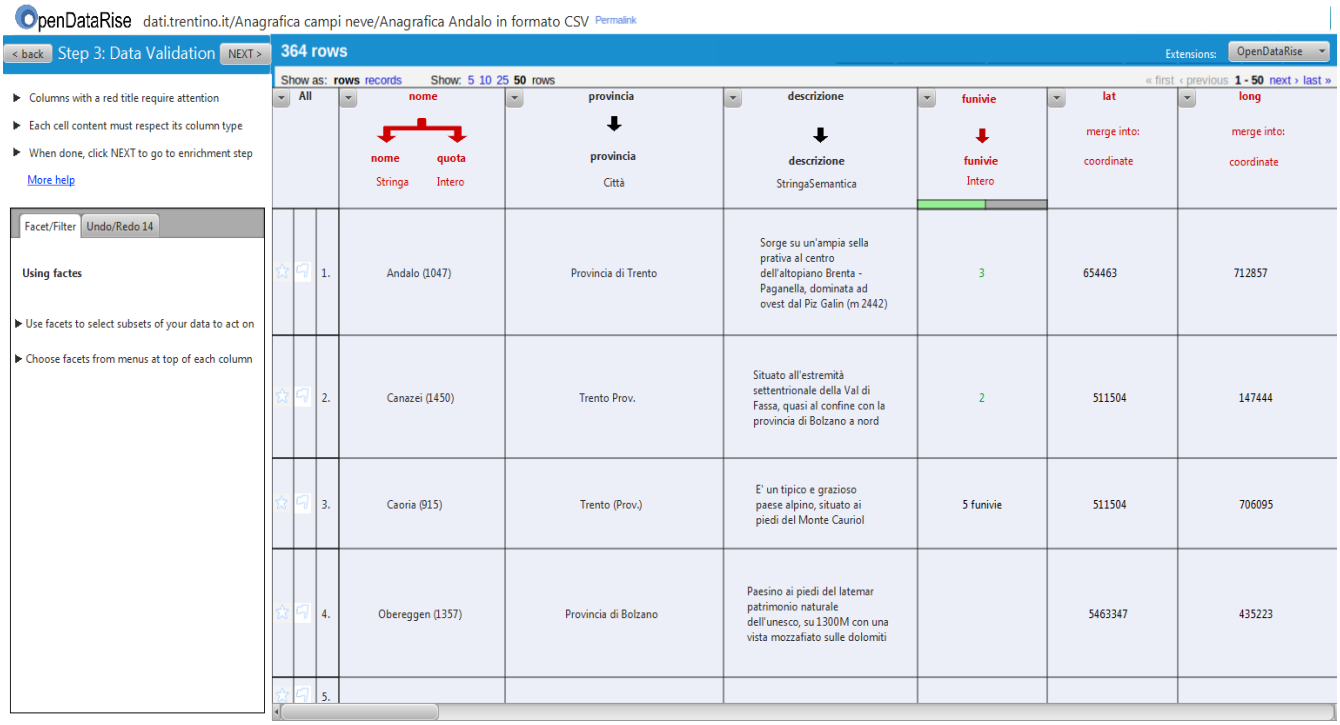
exported during the export phase nor will be considered in the unique indexes, even if they are shown in the user interface.

3 Attribute Value Validation



Drawing 14: Step 3 of the pipeline

In this third step in the pipeline (Drawing 14) the user validates the data in the cells using the standard Refine editor, an example of which can be seen in Drawing 15[1]:



Drawing 15: Mockup for attribute value validation

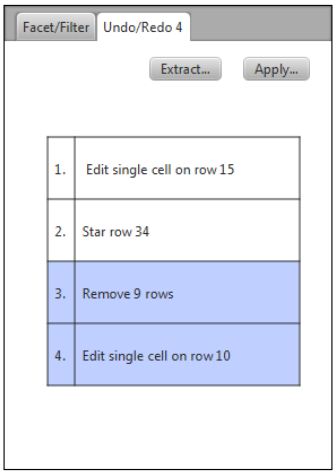
[1] For more in depth examples of the data validation capabilities of open refine please see the videos:
- OpenRefine Introduction: http://www.youtube.com/watch?v=B70J_H_zAWM
- OpenRefine Data Transformation: http://www.youtube.com/watch?v=cO8NVCs_Ba0

3.1 OpenRefine highlights

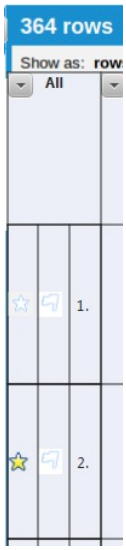
OpenRefine makes easy to operate on data with faceted browsing and infinite undo/redo:

In the left sidebar there is a panel to show facets that filter desired rows. Modification to data are only made to selected rows.

Infinite Undo/Redo support is also accessible in the left sidebar.



Rows can be marked by clicking on the stars and flags in the first column named *All.*



3.2 Changes to OpenRefine

Since OpenRefine interface can be confusing at times, and modifying it too much would probably cause further confusion into existing Refine users, as a general rule we tried to preserve Refine conventions. Still, since a schema match has been established in the previous step, now a number of changes are introduced:

a)

Columns headers show both the original dataset column name and the target attribute name. The target datatype or attribute type of each column is shown under the attribute name.

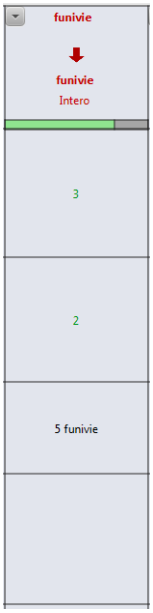
b)

Columns containing values that should be edited in this step show a progress bar under the column type.

The progress bar in column headers indicates how many cells satisfy the target datatype format in a column. The same bar is present in standard Refine when reconciling, to display amount of linked cells. We extend this behavior to datatypes.

The column header is displayed in red if the column contains cells with wrong values. In the example, the *"5 funivie"* value is not *an Integer.*

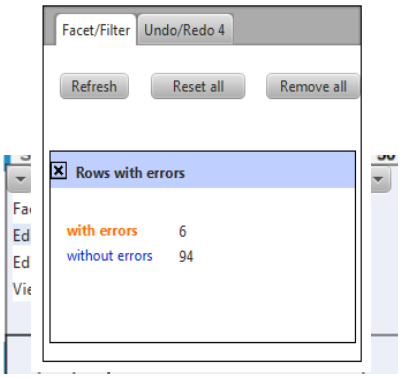
When cells respect the format specified by their datatype, values contained in the cells are colored in green, like *2* and *3*, otherwise they are considered of Refine native text datatype and displayed in black, like *"5 funivie"*. This follows a Refine convention.



c)

Refine allows to select rows on which to operate by using facets, which are displayed on the left sidebar. We will implement a facet to allow the user to select/exclude rows which have cells containing erroneous values.

To the right of this text in Drawing 16 and Drawing 17 we show the process of opening the facet and indicating that only rows with errors must be shown. We stress that Refine only operates on rows selected by facets,

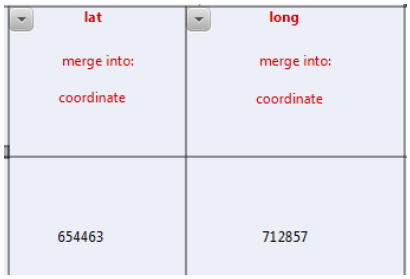


Drawing 17: Selection of rows with errors

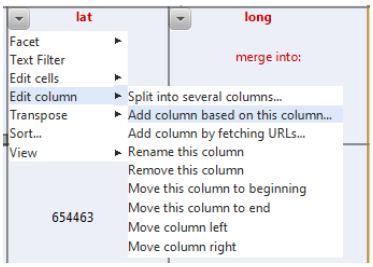
so to eliminate all rows with errors it will be then necessary to use Refine menu to delete filtered rows, which is shown in Drawing 18.

d)

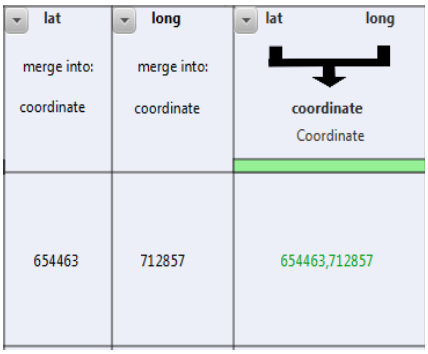
When two ore more columns like *lat* and *long* shown in *Drawing 19* are to be merged in another one they are drawn in red until the user creates the target column, in this case *coordinate*. To do so s/he can exploit standard Refine 'Add column based on this column' functionality, accessible with the menu shown in Drawing 20. The merge can be then performed with Refine regular expressions capabilities, to obtain the column displayed in Drawing 21.



Drawing 19: Columns to be merged



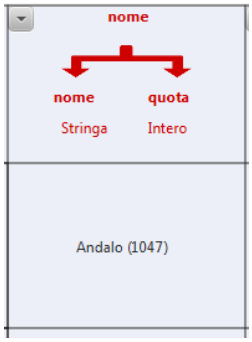
Drawing 20: Menu to add a column



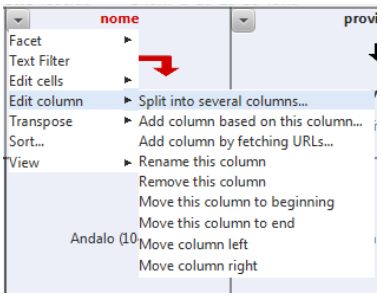
Drawing 21: The result of the merge

e)

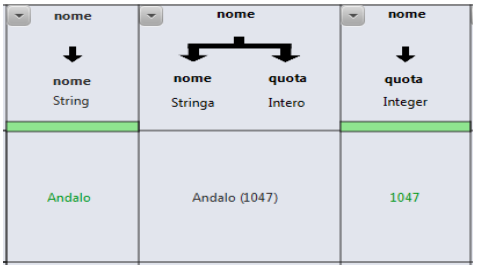
Columns which are to be split are displayed with a red header like in Drawing 22 until the user creates the target columns by using standard Refine facilities. One way to achieve this task is to use the 'Split into several columns...' function like done in Drawing 23, which will allow to perform the split by using Refine regular expressions. capabilities. The result is shown in Drawing 47



Drawing 22: A column to split



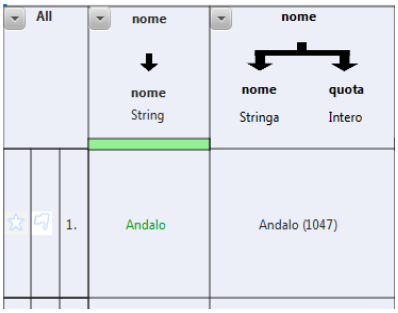
Drawing 23: Refine function to split a column



Drawing 24: Resulting columns after a split

f)

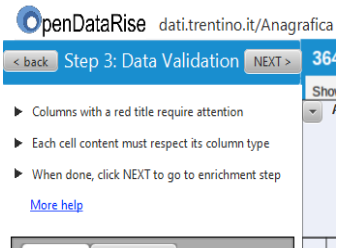
The first column shows the name of the entity at row level, not counting the *All* column native of Refine. In this case the column was created out of a split. Only later in step five (Entity Alignment) an additional column will be added as first column where the user will be able to set the URI of the entity at row level.

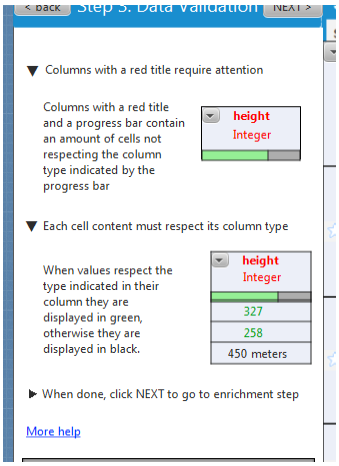


g)

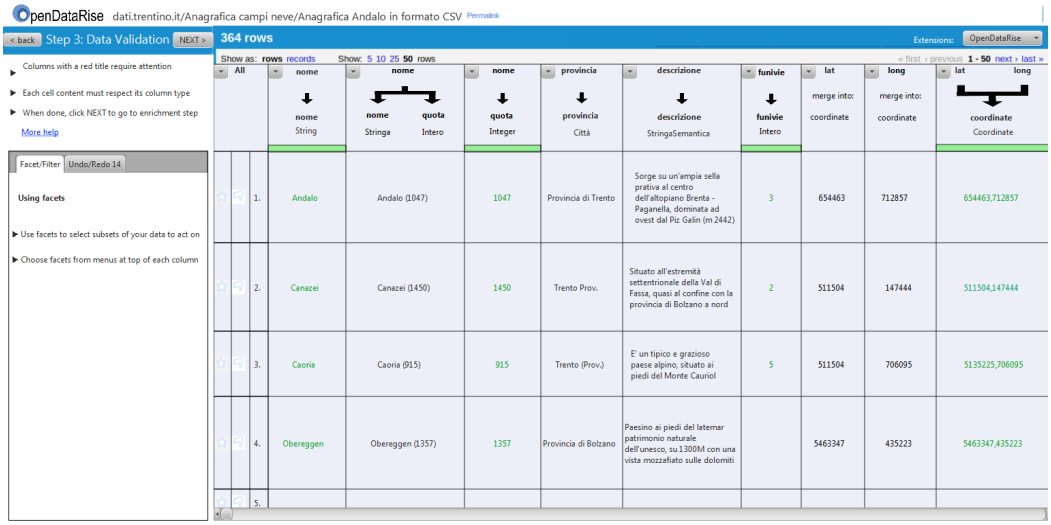
To guide the user a help box is added at the top of the left sidebar.

Clicking on the items will expand them. Clicking on *More help* link will open the online manual in another page.





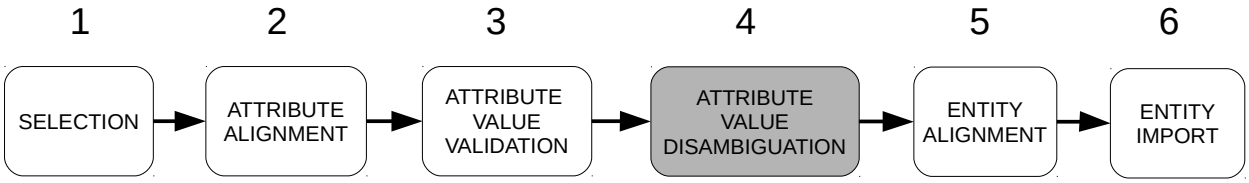
When the user completes all the necessary validation so that no column header is marked in red, the screen will look like in Drawing 25:



Drawing 25: Attribute value validation step after all the columns have been validated

By pressing the NEXT button the user will be lead to the screen depicted in Drawing 27 to perform the next step of semantic enrichment, described in the following Section 4.

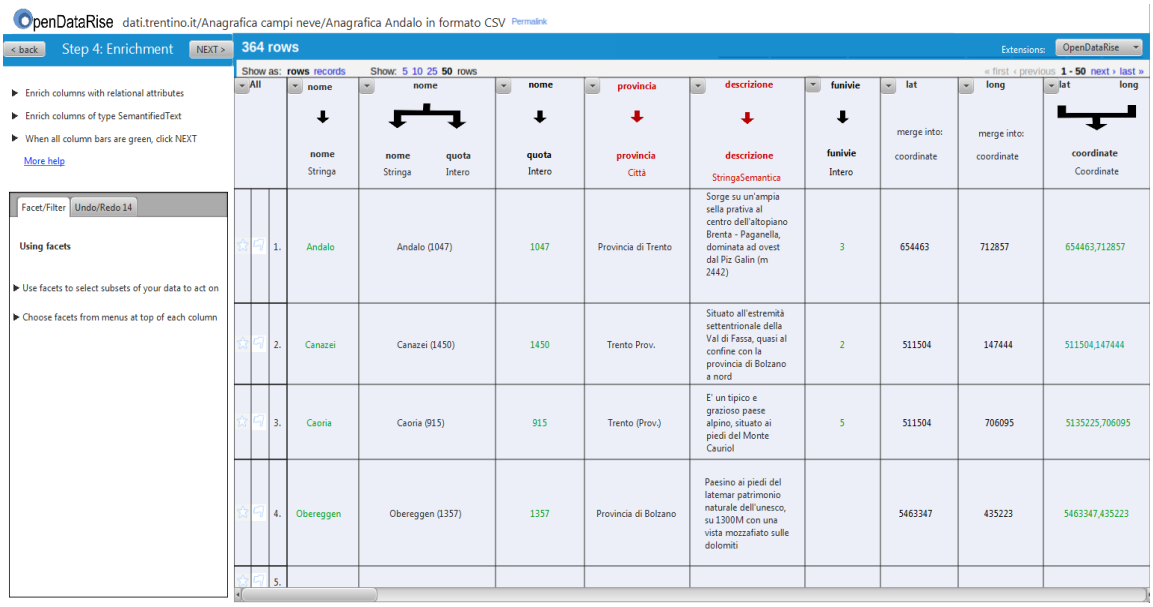
4 Attribute Value Disambiguation



Drawing 26: Step 4 of the pipeline

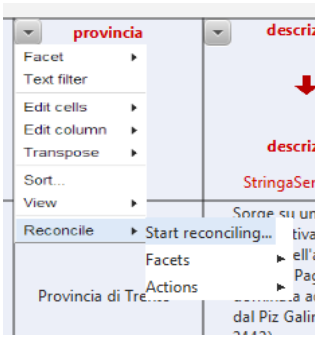
We now propose our solution for semantic enrichment, the fourth step of the pipeline (Drawing 26). Semantic enrichment is composed of *entity disambiguation* and *natural language processing. Entity disambiguation* in our terminology is the task of taking a column holding a relational attribute and linking the names contained in the cells to actual entities being referred by those names. We stress that this task is done at column level (*vertical*) and, even if Refine calls it generically *Reconciliation,* it is different from reconciliation at *row* level (also called horizontal reconciliation), which we will detail in Section 5. *Natural Language Processing* (*NLP*) is the task of assigning a meaning to words occurring inside cells that hold free text. Refine currently cannot handle NLP, although this functionality can be added via plugins described later in Section 7.2, where the state of the art for enrichment is described. When the user enters Step 4, s/he is presented with the screen in Drawing 27, where s/he is invited to act on columns with headers marked in red This can be done thanks to step 2 (attribute alignment), where the system was informed

about which columns need to be processed with NLP and which hold relational attributes. We now describe the tasks for Entity disambiguation in Sec. 4.1 and Natural Language Processing in Sec. 4.2.



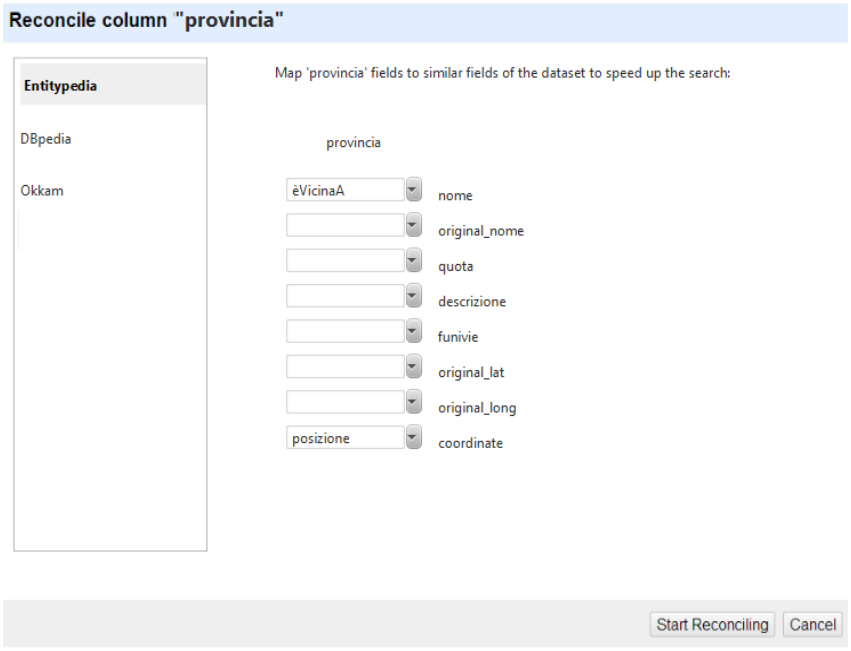
Drawing 27: Semantic enrichment, initial screen

4.1 Entity disambiguation



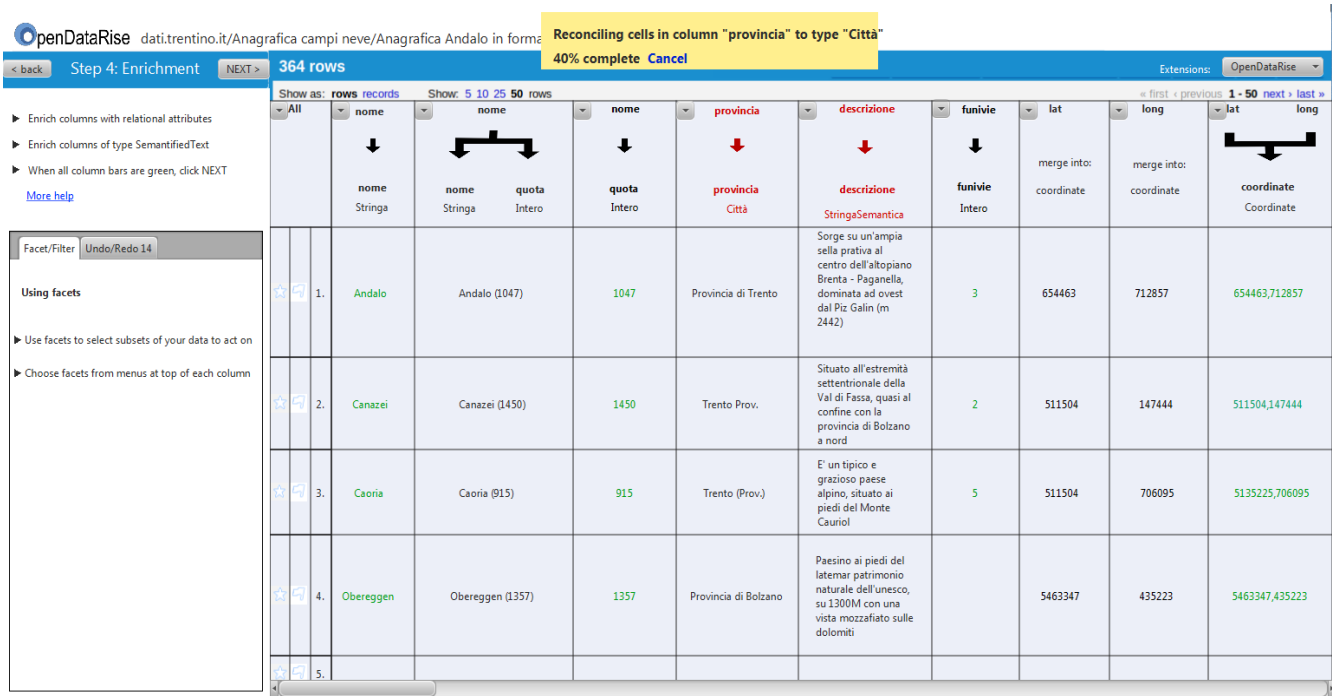
Drawing 28: Menu for executing entity disambiguation on a column

To perform entity disambiguation in a column holding a relational attribute, for example *provincia*, the user must click on the column header arrow button and select *Reconcile->Start reconciling...*, which is the standard menu item in Refine for vertical reconciliation This menu is shown in Drawing 28:
After clicking it, the panel depicted in Drawing 29 will appear. The interface follows the standard one of Refine with the difference that the service for Entitypedia is displayed by default and some buttons for service management are removed to not confuse the user. Since the type of each column has been determined during Attribute alignment step (Section 2), the type of *provincia* column is known and attributes of provincia can be eventually mapped to similar columns in the spreadsheet to speed up the search. In the example of Drawing 29 we map the attribute *èVicinaA* (isNearTo) to *nome* of *LocalitàTuristica* and *posizione* (position) to *coordinate.* Note the search will only use the mapping as a heuristic, so similarity measures (like closeness of coordinates) will be used instead of exact matches.



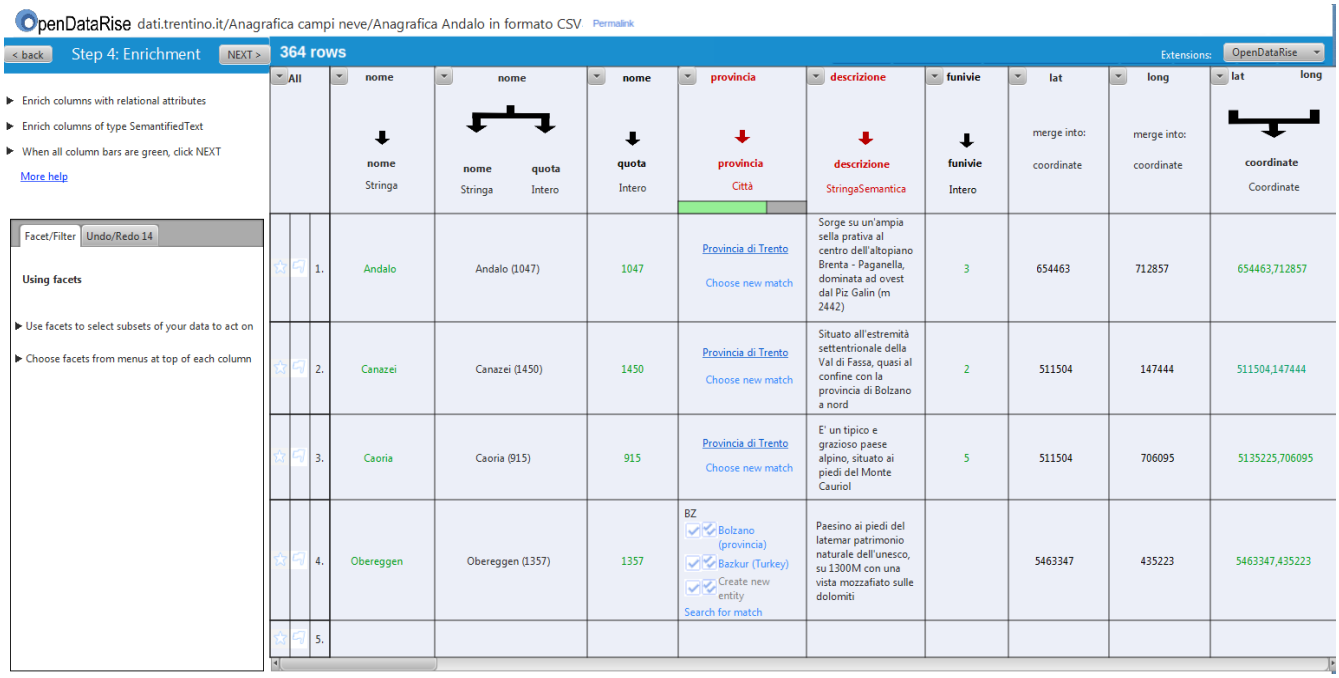
Drawing 29: Mockup for Entity disambiguation panel (also called vertical reconciliation)

The service will allow to choose other columns which might help the disambiguation service to identify the correct entities. By pressing Start Reconciling button, the user will be brought back to the editing screen, with a label in the upper part of the screen indicating the progress of the reconciliation as shown in Drawing 30. This is the standard Refine way to show long running operations.



Drawing 30: Progress of entity disambiguation for column provincia

At the end of the process of enrichment for *provincia* attribute, the column will look very much like after a Refine original reconciliation where the text is underlined, as shown in Drawing 31. Links to entities for provincia will be now displayed in the column.

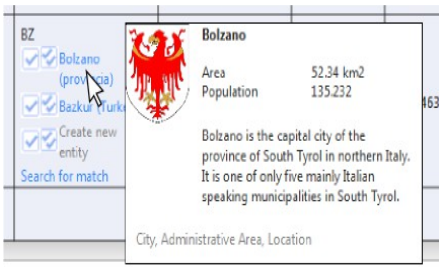


Drawing 31: provincia column after automatic entity disambiguation step

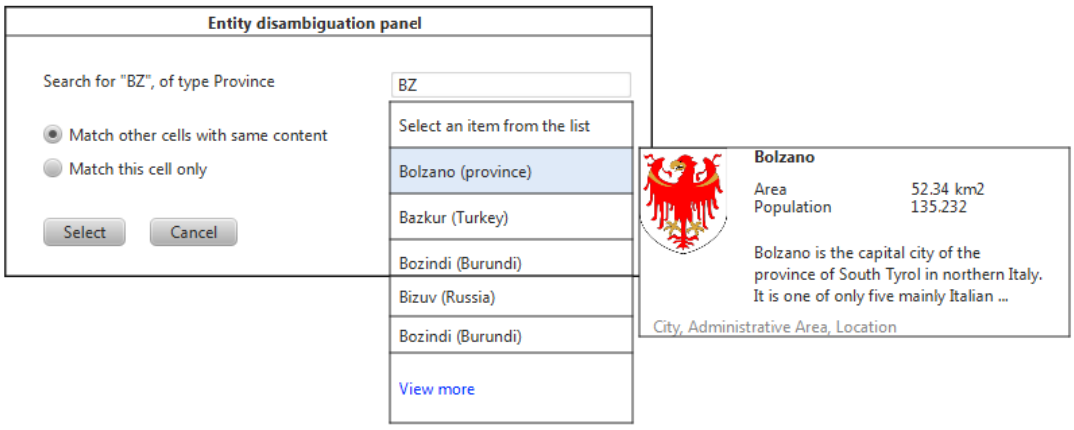
4.1.1 Entities yet to be linked

If the confidence is below a certain threshold many possible links ordered decreasingly by confidence will be shown, and the user will have to select one, as shown in cell with value *BZ*. The user will then click

- the single *v* icon, to link the value for the single cell to the corresponding entity
- the double *vv* icon, to link all the cells in the column with the same name.
- the *v* or the double *vv* corresponding to “Create new entity”, to create a new entity with the name given by the cell value
- “Search for match”, to open the entity disambiguation panel depicted in Drawing 32



4.1.2 Entity disambiguation panel for enrichment step



Drawing 32: Entity disambiguation panel during enrichment step

The panel in Drawing 32 will be presented with a search box in the upper right corner filled with the text in the cell and a list below holding the possible entities to choose. The user will then have to do the following:

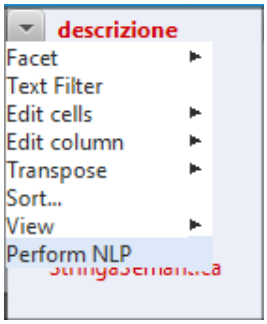
1. Find the correct entity in the list and click on it. Hovering on an item in the list will produce a pop up on the right with information about the entity inside
2. In case the correct entity is not present in the list, the user will either
   - click “View more”
   - Enter new text in the search box and be presented with a new list of possible entities corresponding to the inserted text
3. Indicate whether to disambiguate all the cells with the same content or only the current cell
4. Click Select button

4.1.3 Linked entities

When an entity name is linked to an id, either by user choice or because the confidence is above a certain threshold in the automatic enrichment, only one link per cell is shown, like for *Provincia di Trento* link. If the user is not satisfied with the link, clicking on “Choose new match” displays in the cell a list of possible best matches as the one shown for cells with confidence level below the threshold.

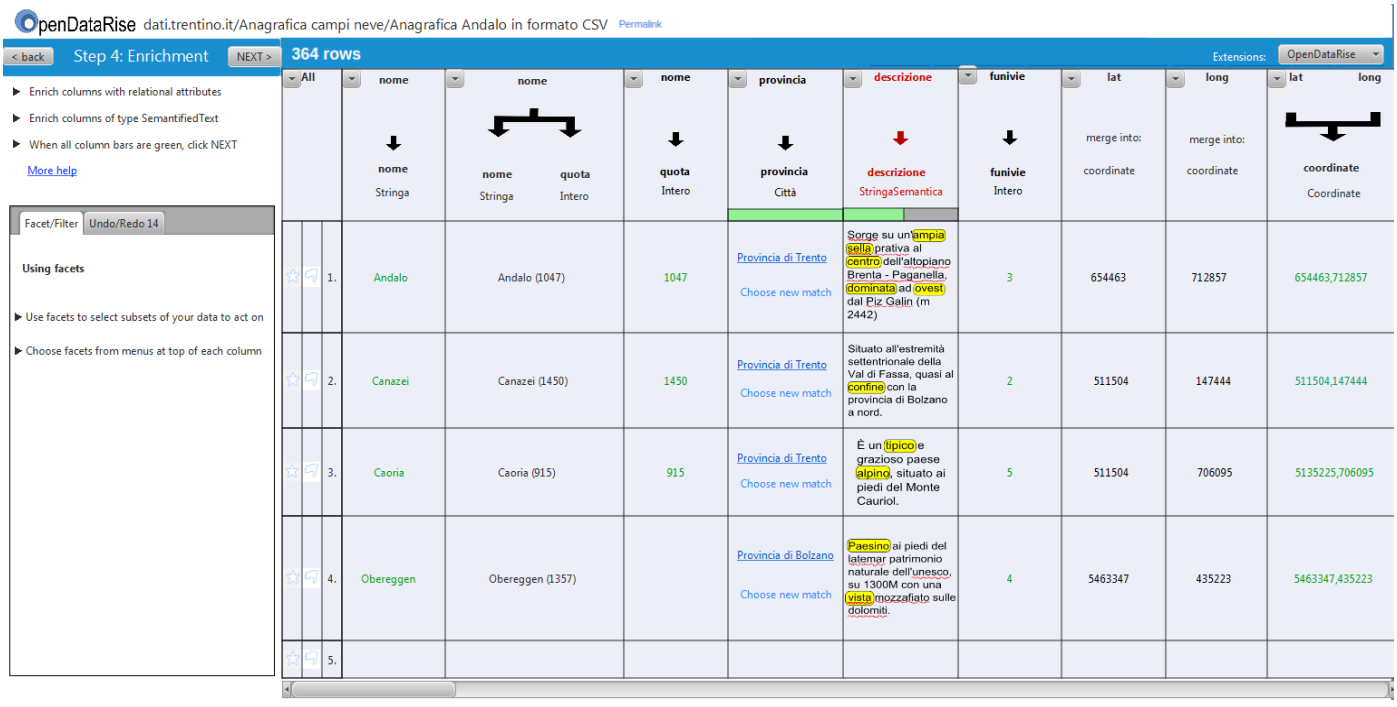
4.2 Natural language processing

To perform natural language processing in a column holding an attribute of type S*emantifiedText*, like for example *descrizione*, the user must click on the column header arrow button and select *Perform NLP,* as shown in *Drawing 33:*



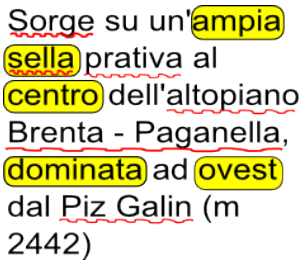
Drawing 33: Menu to start natural language processing

The software will then find the meaning of words contained inside the free text inside each cell. The progress will be shown as during Entity Disambiguation in Drawing 30. At the end of the process the user will see a screen like in the following Drawing 34:



Drawing 34: "descrizione" column after automatic Natural Language Processing

A progress bar will appear under the column name to indicate the number of cells that have been completely semantified in the column. A cell will be considered completely semantified if all words will have no red underlinings. Let's suppose the user will see the following description of Andalo town in Drawing 35 after the automatic enrichment:



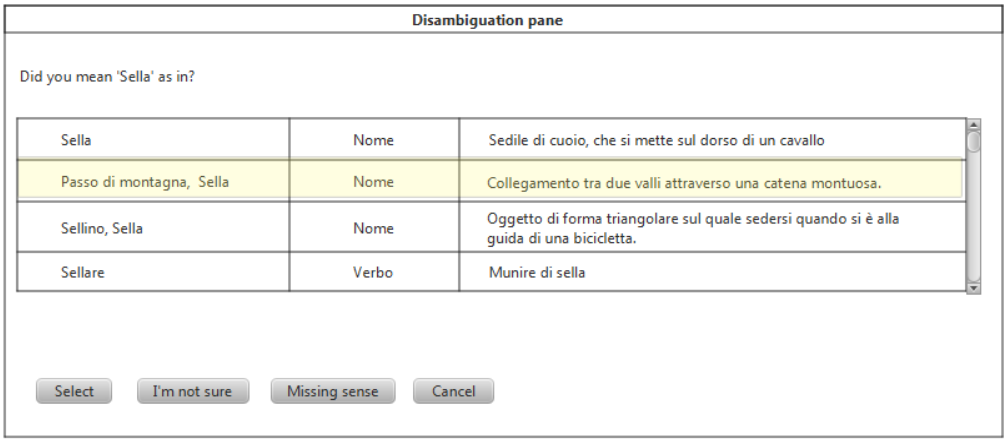
Drawing 35: Example of a semantified text to correct

For each word, the automatic enrichment process will have either:

a) determined the word sense with a degree of confidence above a preset threshold. In this case the word will be marked in plain yellow, like the word 'ampia'.
b) determined a word sense, but with a confidence below the threshold. In this case the word will be marked in yellow and underlined in red, like the word 'sella'
c) failed to find the word in the vocabulary. In this case the word will be just underlined in red, such as 'Sorge'

Disambiguation pane

By clicking on *sella,* the following dialog shown in Drawing 36 will appear:



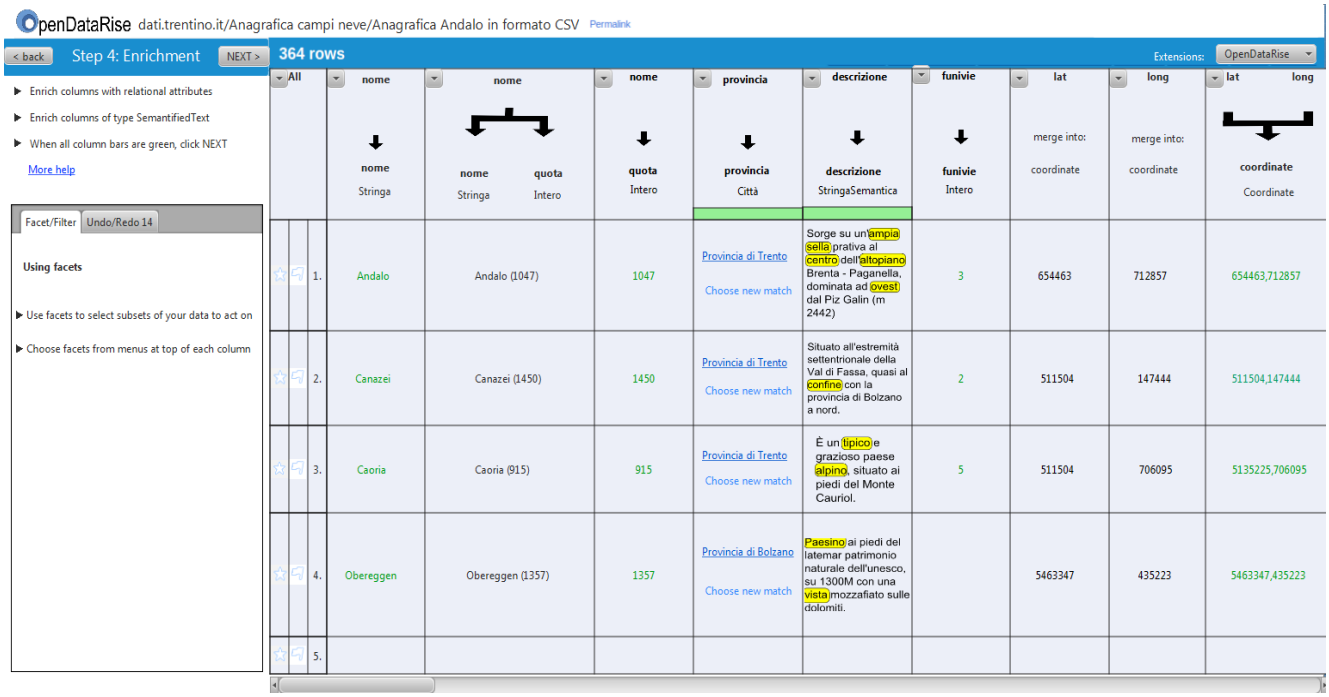
Drawing 36: Mockup for the Disambiguation pane, where the user searches for 'sella'

Clicking on 'I'm not sure' or 'Missing sense' button' will remove the eventual red underline from the word. All the indications expressed by the user will be saved in a log. In the end the corrected text will look like

Sorge su un'ampia sella prativa al centro dell'altopiano Brenta - Paganella, dominata ad ovest dal Piz Galin (m 2442)

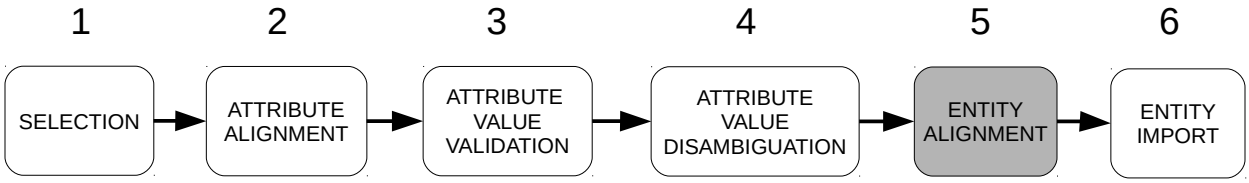
Drawing 37: Example of a semantified text with no errors

After having corrected all the senses the screen will look like in Drawing 38 and the user will then be allowed to proceed to Step 5 for Entity Alignment at row level by pressing the *NEXT* button.



Drawing 38: End of enrichment step

5 Entity Alignment

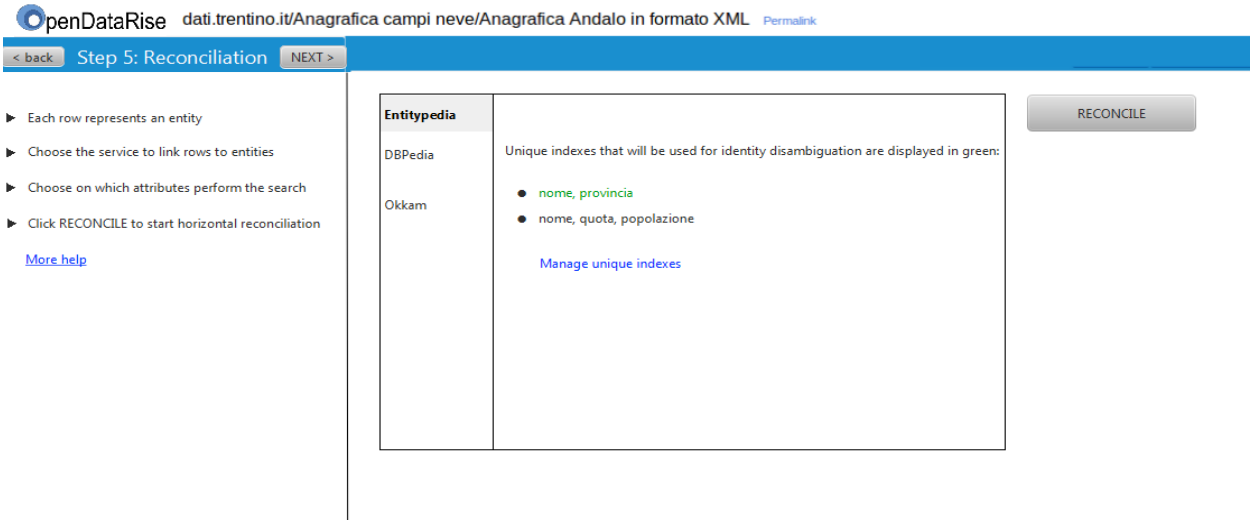


Drawing 39: Step 5 of the pipeline

In this step, the fifth in the pipeline as depicted in Drawing 39, each row is considered to represent an entity and a unique identifier in the format of a URI must be assigned to it. The row can be either assigned to a new URI or to an existing one. In case existing entities differ in their attribute values from the values in the row, the user will be able to modify an existing entity if needed.

5.1 Automatic reconciliation

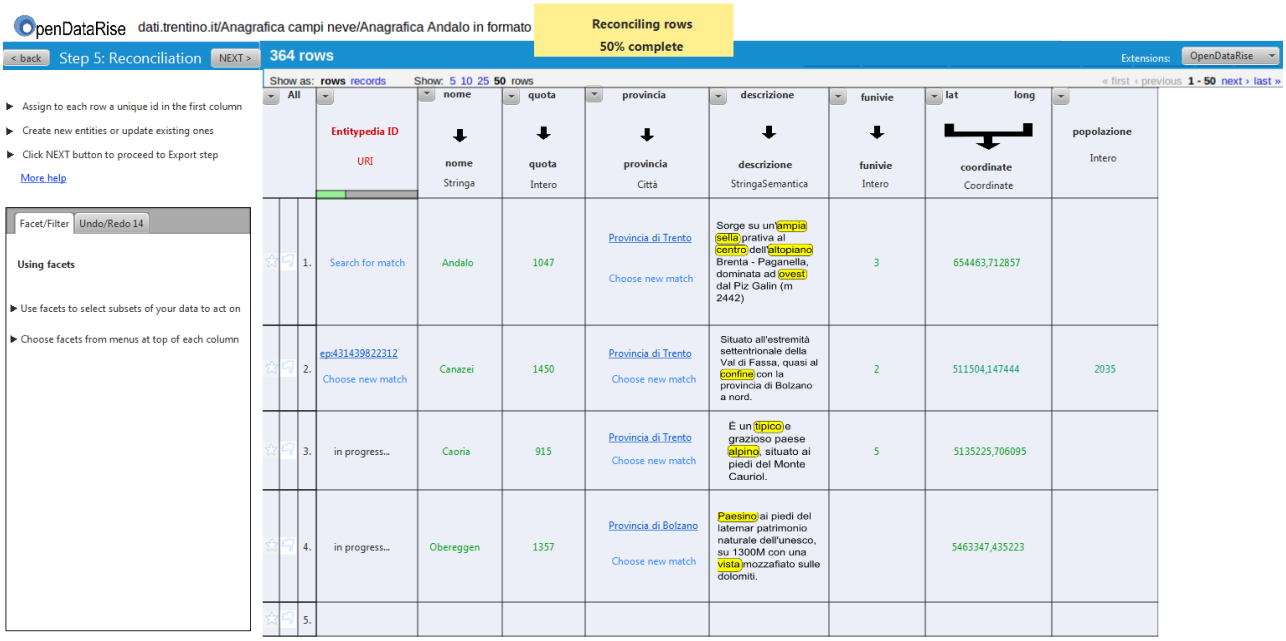
Upon entrance in Entity Alignment step 5 from Step 4, the user will be prompted to choose which service to use for linking rows to entities, as depicted in Drawing 40:



Drawing 40: Entity Alignment step, service selection screen

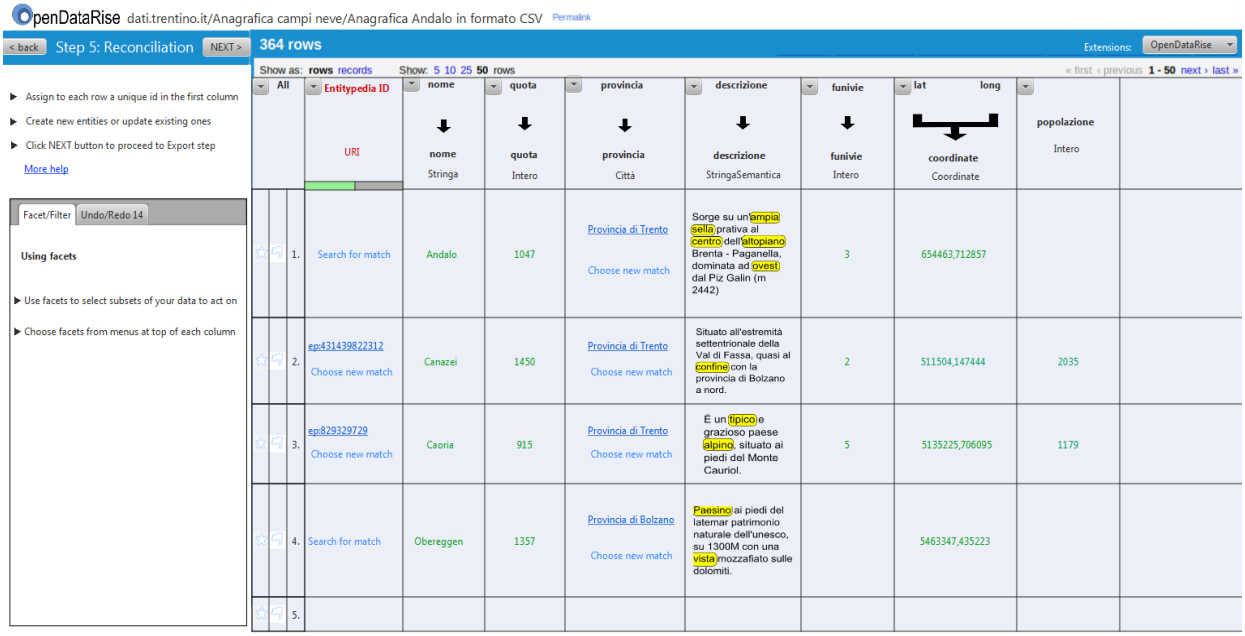
For Entitypedia service the panel shows in green the unique indexes that will be used to perform identity disambiguation during reconciliation. Sometimes unique indexes cannot be used because one of the attributes in their set was not imported from the dataset (like *popolazione,* which was absent in the original data). Pressing *Manage unique indexes* will open the panel already shown in Drawing 13 during Attribute

alignment step. After pressing *RECONCILE* button, automatic reconciliation will start and the spreadsheet along with the progress depicted in Drawing 41 will be shown to the user.



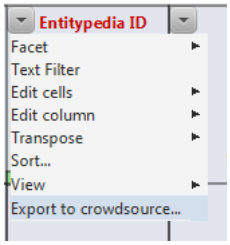
Drawing 41: Automatic reconciliation progress panel

Supposing Entitypedia was the service chosen in the previous screen (Drawing 40), a new column named Entitypedia ID is added to the spreadsheet to show linking of rows to entities. In this step columns that previously required to be merged or split, like original *nome*, *lat* and *long are* not shown. New columns are added for attributes of the row entity type *LocalitàTuristica* that were not imported from the original dataset, like *popolazione*. Cells in ID column for which the automatic reconciliation is pending show the text “in progress...”. In the meanwhile the user can edit cells for which automatic reconciliation has already been performed. After automatic reconciliation, the whole ID column will be filled like in Drawing 42:



Drawing 42: Dataset after automatic identity disambiguation

Automatic reconciliation can either succeed in finding an entity perfectly matching the values in a row or fail. If it succeeds, like in the case for *Canazei,* then the clickable link to the found entity is displayed in the respective cell in the Entitypedia ID column. Clicking on “Choose new match” will open the identity disambiguation dialog shown in Drawing 44, which we will describe later. If the automatic reconciliation fails, the user can export the task of linking all the rows that still don't have an id as a crowdsourcing job (Drawing 43). In the first iteration of the software we will not implement crowdsourcing functionality.

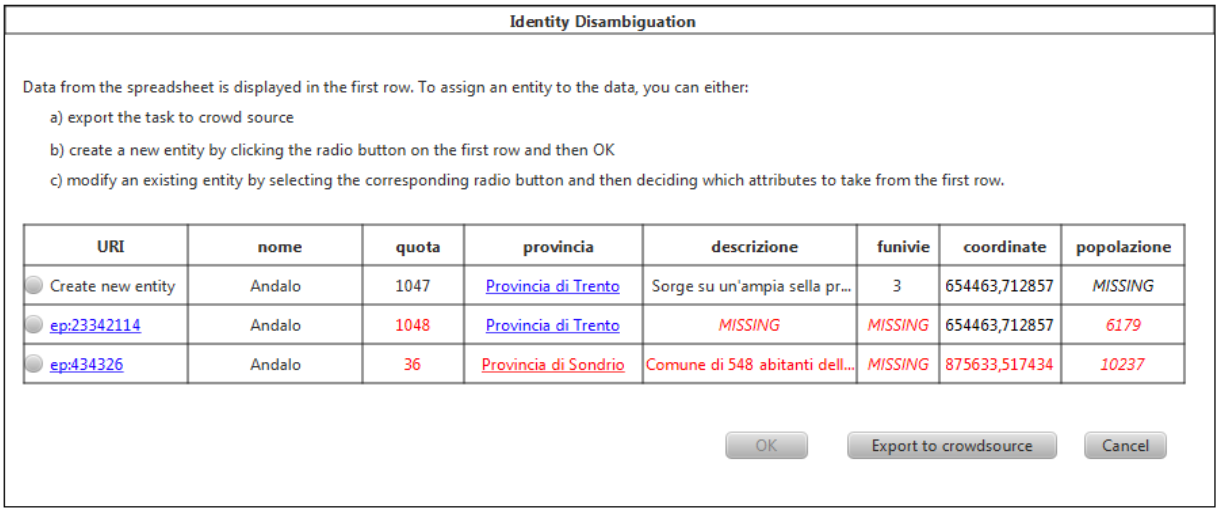


Drawing 43: export to crowdsourcing menu during Entity Alignment step

5.2 Manual linking

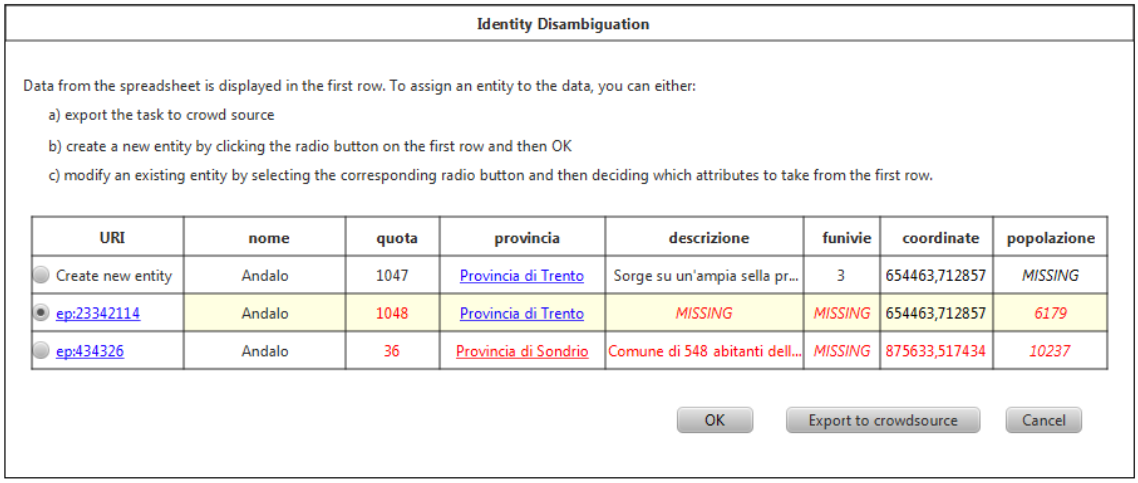
As an alternative to crowdsourcing, the user can manually link the rows to entities. For example, in the case of *Andalo*, the user can click “Search for match” on the URI cell to open the Identity Disambiguation dialog shown in Drawing 44. It allows to create a new entity based on the values in the dataset, or select/modify an existing one in Entitypedia, if present. The user must then do the following:

1. Observe the difference between the entity in the dataset, shown in the first column, and the other entities found in Entitypedia. Values different from the ones in the dataset will be marked in red. Missing values will be written with the label *MISSING* in red



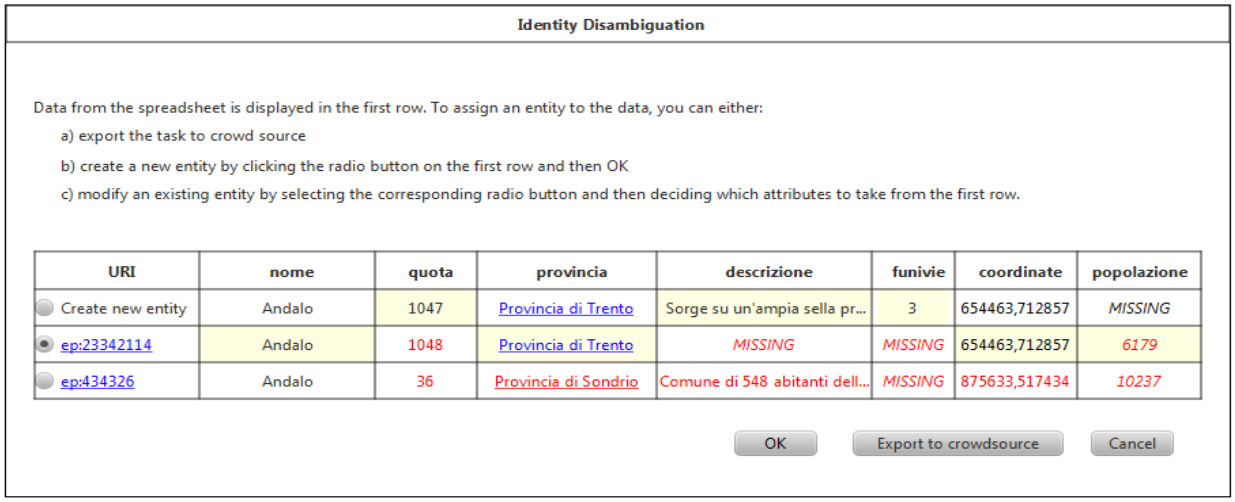
Drawing 44: identity disambiguation panel

2. Decide to either crowd source the task of identity disambiguation, create a new entity, or select/modify an existing one.
   a) to crowd source the task it will be enough to press the 'Export to crowdsource' button. For the first iteration of the software we will not implement this functionality
   b) to create a new entity the user will select the radio button over the first column and press OK
   c) to modify an existing entity, s/he will select one radio button for the other entities, like it is done for entity *ep:23342114* in Drawing 45. Upon selection of an entity, all the entity fields in it will be shown with a yellow background.



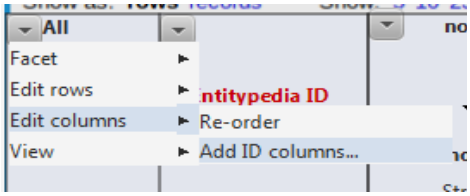
Drawing 45: Identity disambiguation, selection of existing entity

3. Eventually modify an existing entity, as depicted in Drawing 46. If the user selected an existing entity, like *ep:23342114* in the example, he will be able to select fields from the original dataset (in the first column) to substitute to the fields of the selected entity. Clicking on a field of either the original dataset in the first column or in the selected entity column will switch its background color to yellow and the background of the field in the other column to white. So for example clicking on the value *1047* for the *quota* field in the original column will turn its background to yellow, to mean the value of *1048* in the entity *ep:23342114* is going to be changed to *1047.*



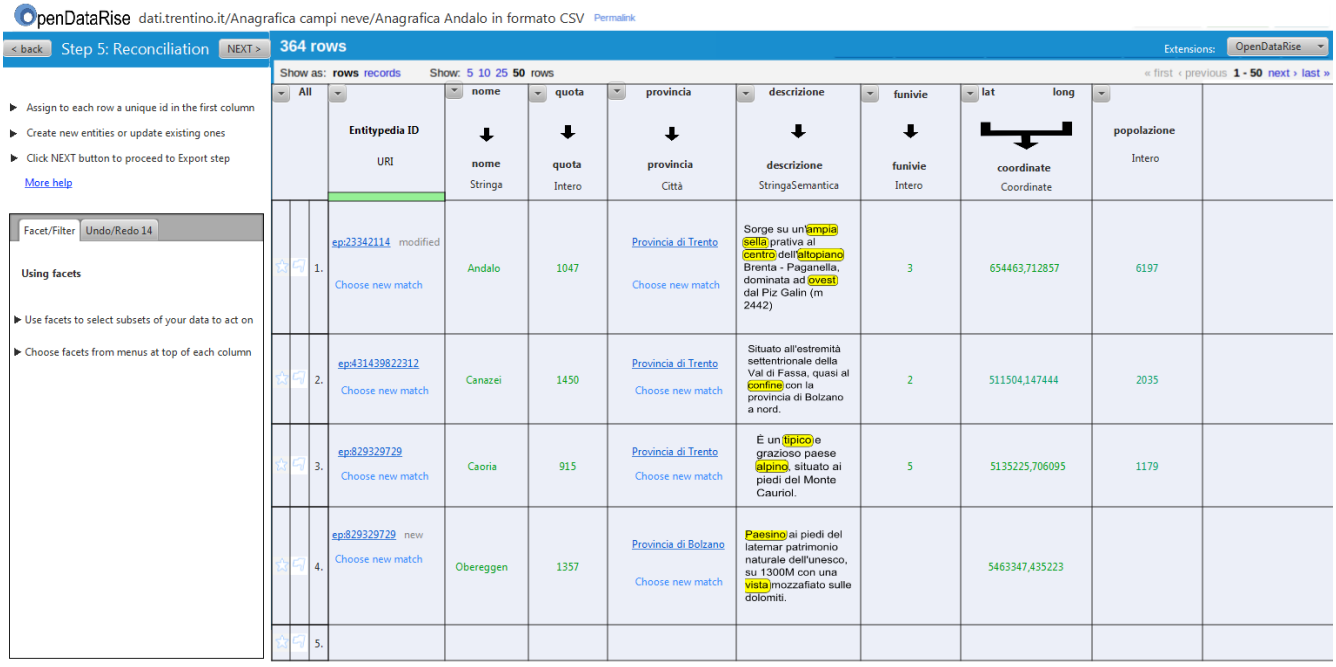
Drawing 46: Entity matching, existing entity modification by merge of values from the dataset

After pressing OK the user will be brought back to the spreadsheet. Additional ID columns may be added by clicking in the options for the 'All' column, as shown in Drawing 47. After clicking, the user will be lead to the screen depicted in Drawing 40, which will show services which haven't already been used for reconciliation.



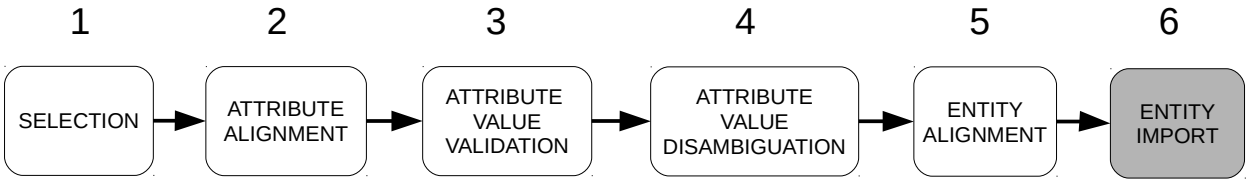
Drawing 47: Menu to add an ID column

After all the entities are linked, the dataset will look like in Drawing 48, where the data is completely semantified and ready to be exported. A grey label in an ID cell, if present, will indicate whether the URI represents a modified entity or a new one. The label color and position follows a Refine convention, and we just added the possibility to indicate a modification has occurred.



Drawing 48: Completely semantified dataset

6 Entity Import

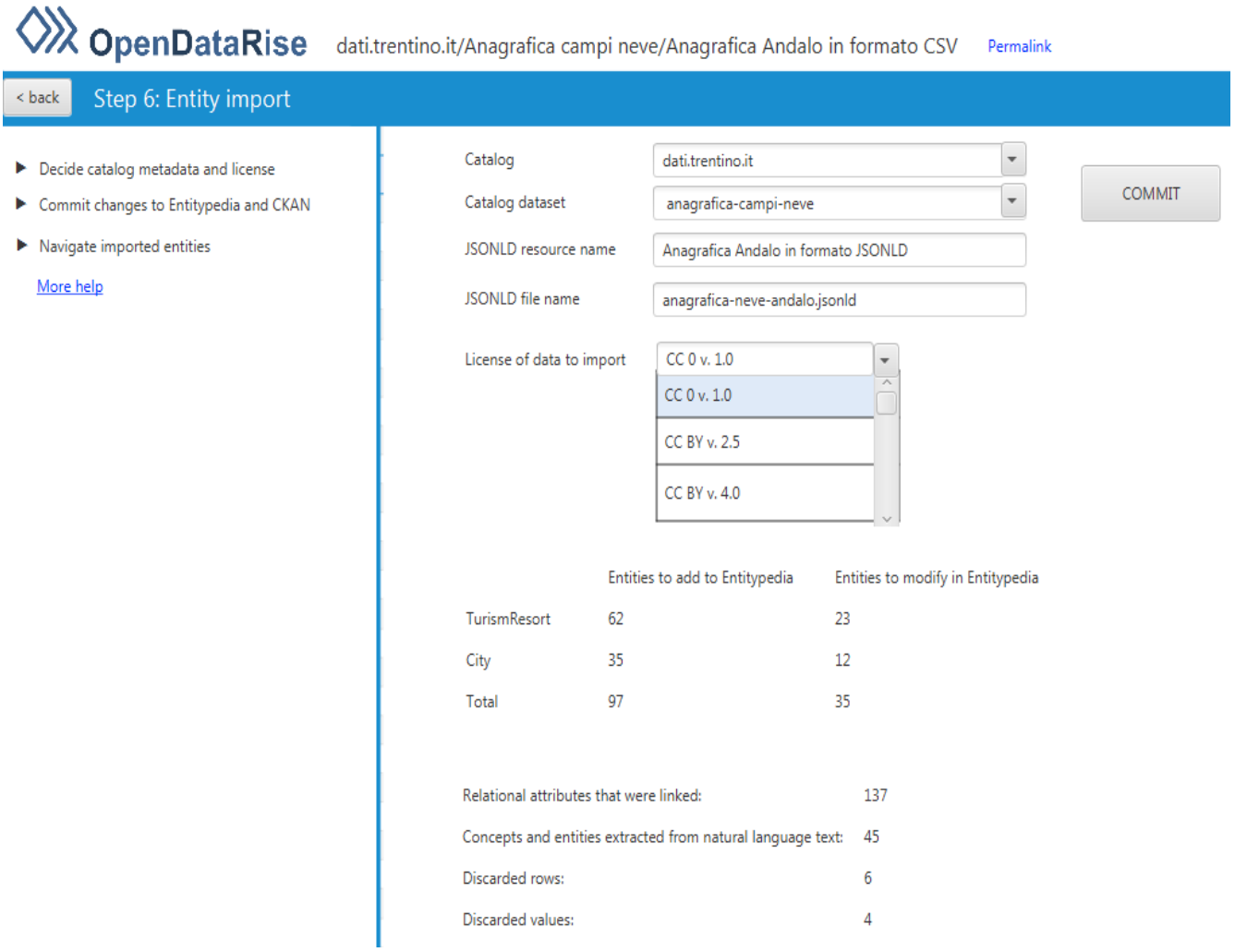


Drawing 49: Step 6 of the pipeline

In the sixth and last step of the pipeline depicted in Drawing 49 it will be possible to

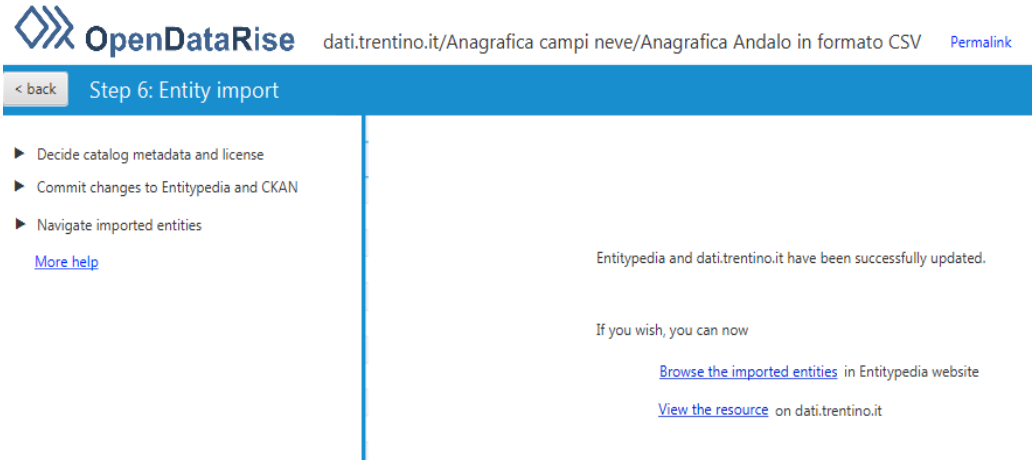
1. import entities in Entitypedia
2. create (and possibly download) a JSONLD file to be be stored on OpenDataRise server
3. create a resource in the target CKAN catalog pointing to the JSONLD stored in ODR server.

The user is presented with the screen depicted in Drawing 50, where CKAN metadata and a summary of modifications to do in Entitypedia is shown. By default the target catalog and dataset are the same as the source ones. Dropdown menus will allow to see previously used catalogs and datasets.



Drawing 50: Export step

Upon pressing *COMMIT* button, if everything went fine the user will see the success message shown in Drawing 51.



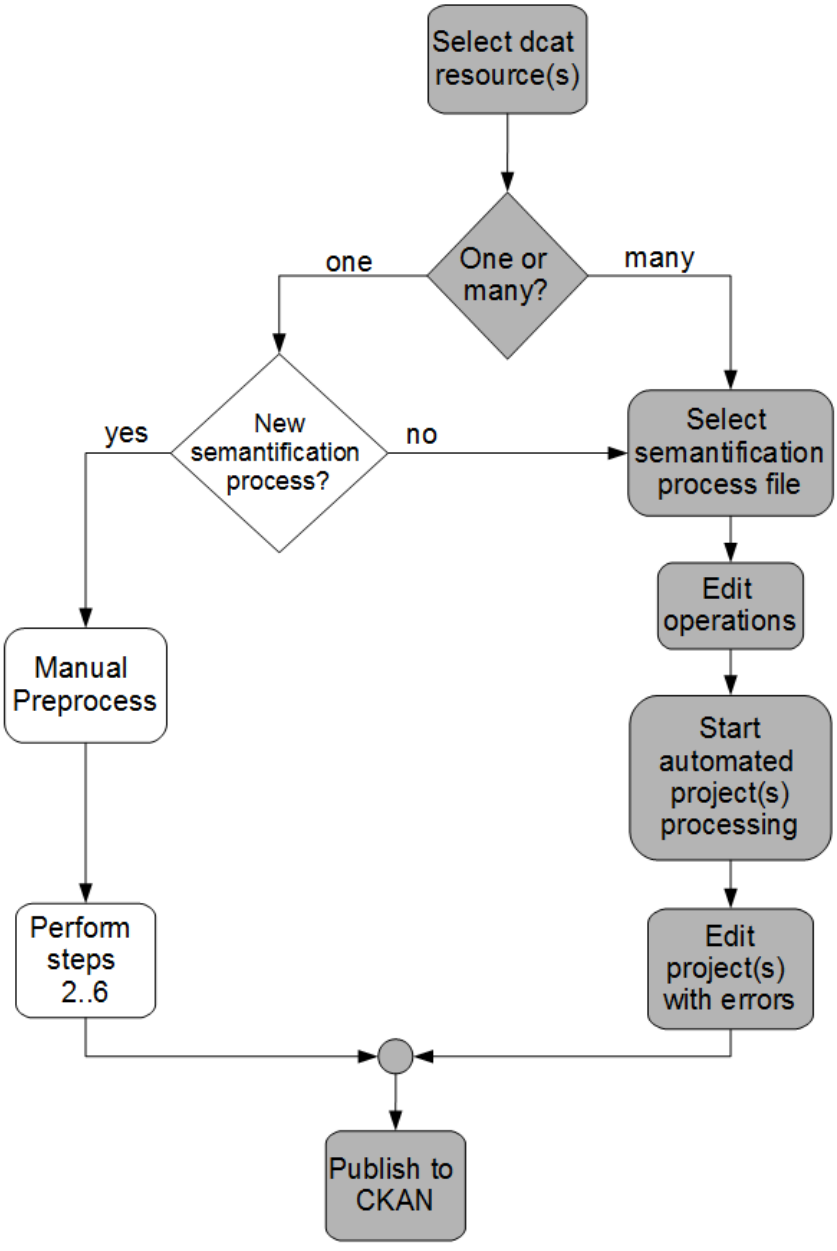
Drawing 51: Import step success

7 Further details

We now present the automated application of an existing semantification process to several datasets in Section 7.1, and the state of the art for text enrichment in Refine in Section 7.2.

7.1 Multiple resource selection

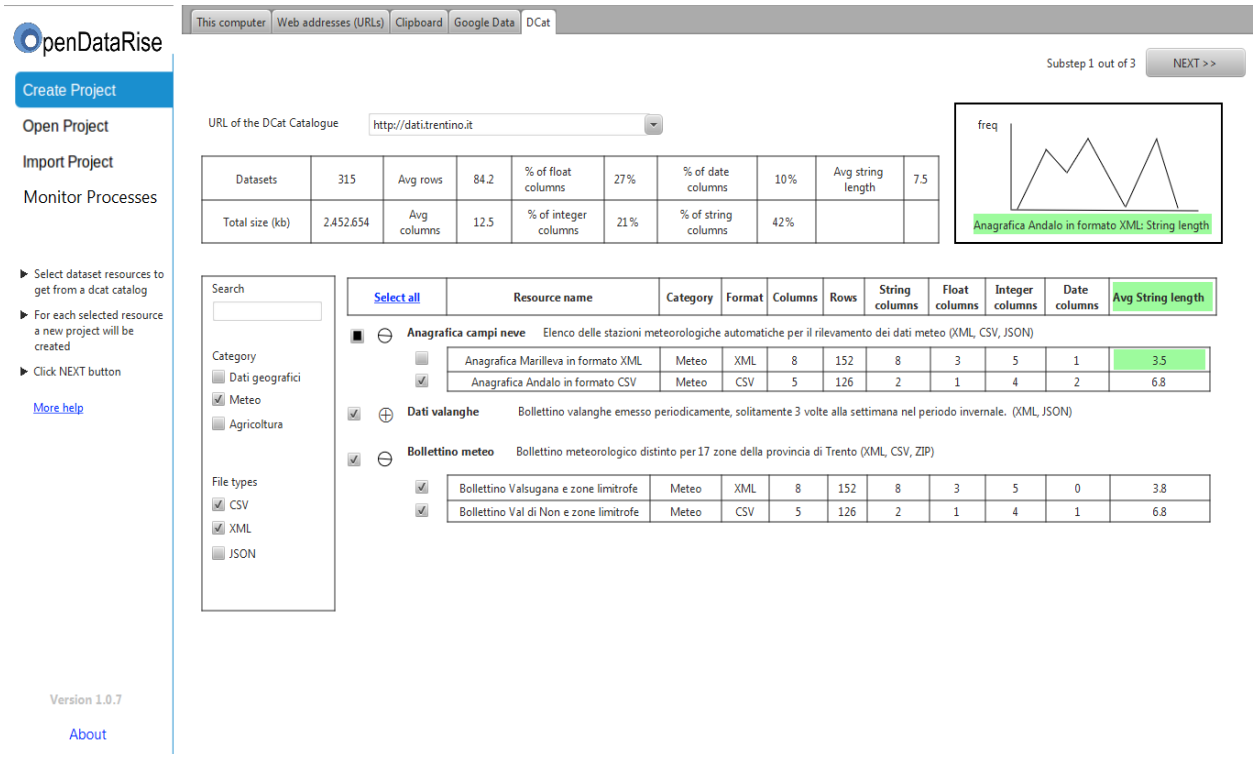
Many dataset resources can also be selected in order to automatically perform a pre-existing semantification process on each of them. This workflow is depicted in Drawing 52:



Drawing 52: Multiple dataset resources selection for batch processing

7.1.1 Select dcat resources

The first substep is similar to the single dataset selection shown in Section 1.1, but in this case the user selects many resources, as depicted in Drawing 53:

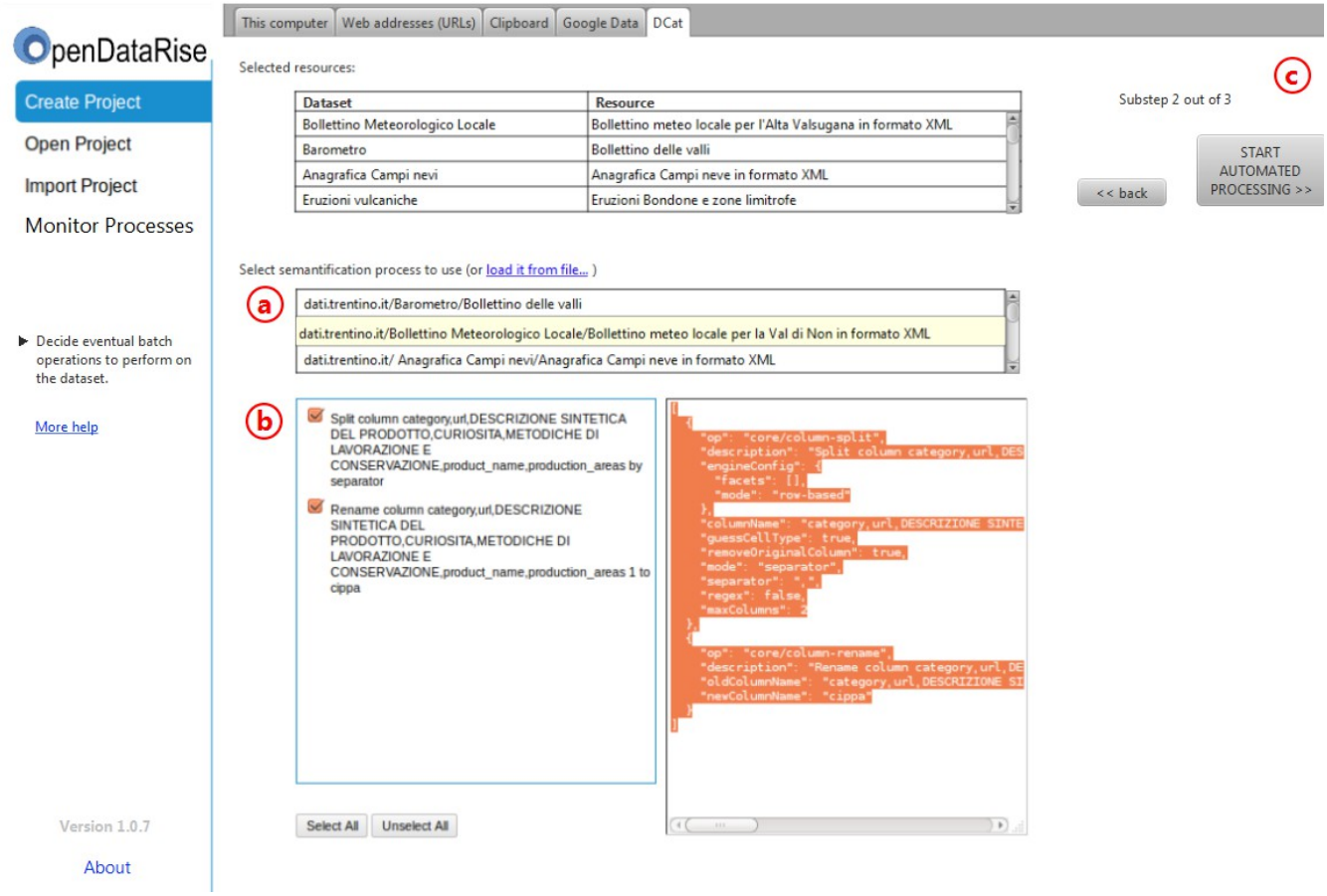


Drawing 53: Substep 1.1-multiple: Multiple dataset resources selection

7.1.2 Select semantification process

The system records in a semantification process file all the operations carried out during a dataset resource semantification. Each dataset resource will then have associated a semantification process file stored in the system. This file can be exploited to automatically perform the operations described inside to similar dataset resources. In this substep, depicted in Drawing 54, the user must do the following:

a) Select the semantification process file to use either from harddisk or from a dropdown menu containing a list of all the semantification processes previously carried out by the user.
   1. If all the resources selected in the previous substep 1.1-multiple belong to the same dataset, and in the past the user already carried out a semantification process on a resource of that dataset, this semantification process will be shown by default in the input bar of the dropdown



Drawing 54: Substep 1.2-multiple: Semantification process definition

menu. Otherwise, the input bar will be blank. In order to fill it, the user can select a value from the dropdown list or click on a link to load a semantification process file from hard disk. Once the file gets loaded, the input bar will show the file path on the harddisk.

2. Whenever the input bar gets filled with a valid semantification process file, the operations contained inside the semantification file are shown below the dropdown menu.

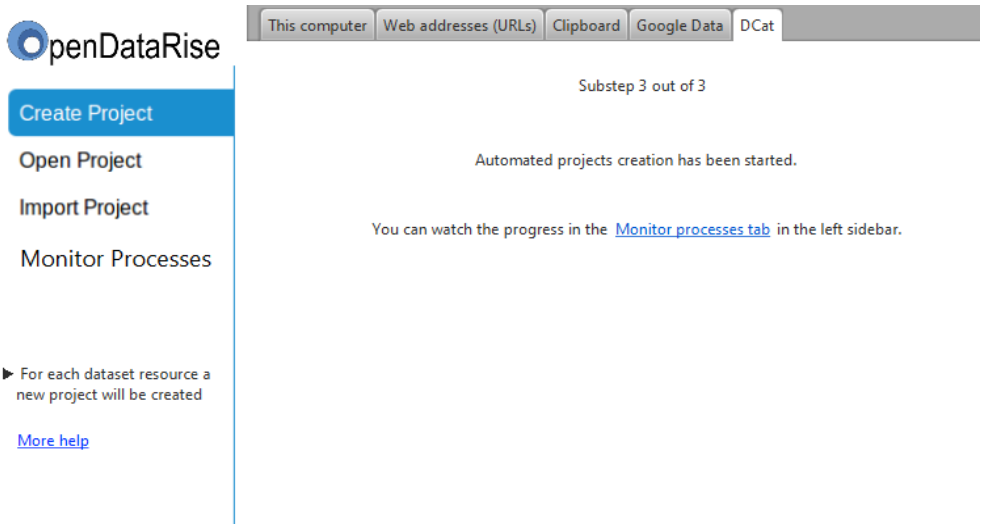
b) Edit the operations to perform on the dataset resources selected in the previous substep 1.1-multiple,

c) Click 'START AUTOMATED PROCESSING' button in the upper right corner

To begin with we can reuse the Operation Editor of Refine to select operations of interest. During next iterations we could improve it as certain operations can only be understood by looking at the JSON representation which is not suitable for inexperienced users.

7.1.3 Automated projects creation

In the third and last substep, depicted in Drawing 55, the user is informed the automated project creation has started, and that its progress can be checked in the *Monitor processes* tab in the left menu. The *Monitor processes* tab is depicted in Drawing 56. This way even if the user closes the browser he will be later able to easily locate the panel to check the importing progress. Also, in the future the *Manage processes* tab could be useful for displaying other possible pending processes (batch upload to CKAN, calculation of statistics, etc).



Drawing 55: Substep 1.3-multiple: Automated projects creation

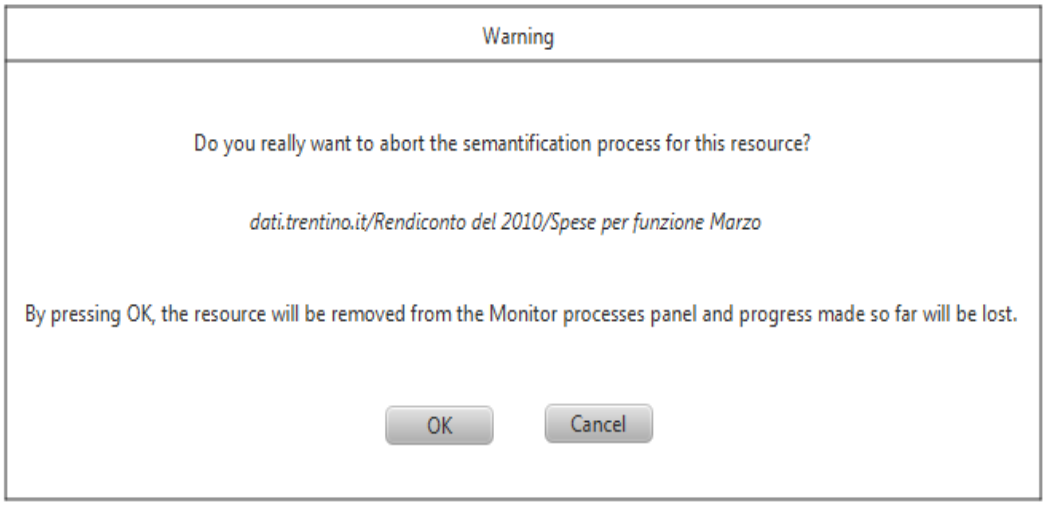
7.1.4 Monitor processes tab

In the *Monitor processes* tab, depicted in Drawing 56, all the dataset resources to be automatically semantified are shown, grouped by their semantification process name displayed in bold.

Drawing 56: Monitor processes tab

a) Each dataset resource has associated the time when it started, and its completion percentage shown in a progress bar. Initially the dataset resource is displayed in black

b) Two buttons are shown next to the progress bar. The right one starts/pauses/resumes the process. The left one labeled with an an 'X' aborts the process, before asking for a confirmation like the one depicted in Drawing 57:



Drawing 57: Abortion of a semantification process warning

c) When the semantification process for a resource completes:
   - a project for this resource is created in Refine
   - the progress bar turns into the current time
   - eventual messages about the process are shown in the *Process info* column, divided in the following types: errors, warnings, manual intervention required (i.e. for entities/concept disambiguation). Hovering on the writings with the mouse will display a pop up with a detailed description
   - the resource name becomes an underlined link. By clicking on the link the user will be lead to the newly created Refine project in another tab

d) The dataset resources are grouped by their semantification process name along with the time when the process was started
   - If in a group there are still resources to be semantified, a progress bar is shown in the 'Completed' column of the group, indicating the average completion percentage for all resources in the group. Otherwise the time when the last process in the group ended is shown.
   - Abort, pause, and start buttons are shown next to the progress bar. These buttons affect all the resources in the group
   - A group can be either expanded or contracted.

e) When a group is contracted:
   - no dataset resources are shown
   - in the Process info, the messages divided by type of every resource in the group is shown

f) When a group is expanded:
   - all the resources in the group are shown below the group name
   - the Process info cell for the group name is left empty

g) Buttons to abort, pause, start/resume all the dataset resources in all groups at once are displayed in the upper part of the screen, along with the progress status of all processes. Clicking 'ABORT ALL' will trigger a warning similar to the one to abort single resources displayed in Drawing 57
h) Since progress bars in different positions can have different meanings (number of executed steps, number of processed resources) to avoid confusion numbers displayed in the progress bars will be represented as percentages

7.2 State of the art for enrichment

For long text enrichment there are currently two solutions in Refine, both dealing only with Named Entity Recognition (NER). One is provided by Zemanta[1], and the other one by FreeYourMetaData group[2]. The first one, given a column of descriptions, allows to restrict the type of entities to be found along with a preview for the first description in the column. Entities are taken from Zemanta NER service. Here in Illustration 1, we can see an example for a column `Summary` of books:

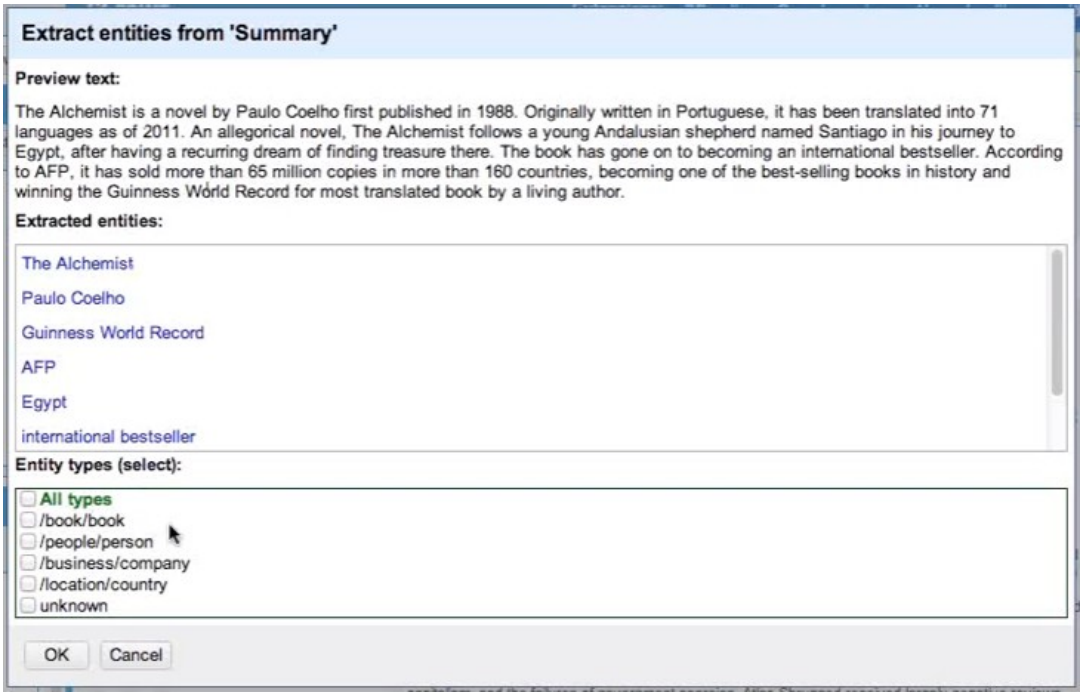


Illustration 1: Zemanta NER plugin example, entity type selection

After OK button is clicked we can see the result in Illustration 2, where in the column person [/people] the newly added entities, in these case two persons `Ayn Rand` and `John Galt` referred to in the `Summary` field.

[1] https://github.com/sparkica/Refine-NER-Extension
[2] http://freeyourmetadata.org/named-entity-extraction/

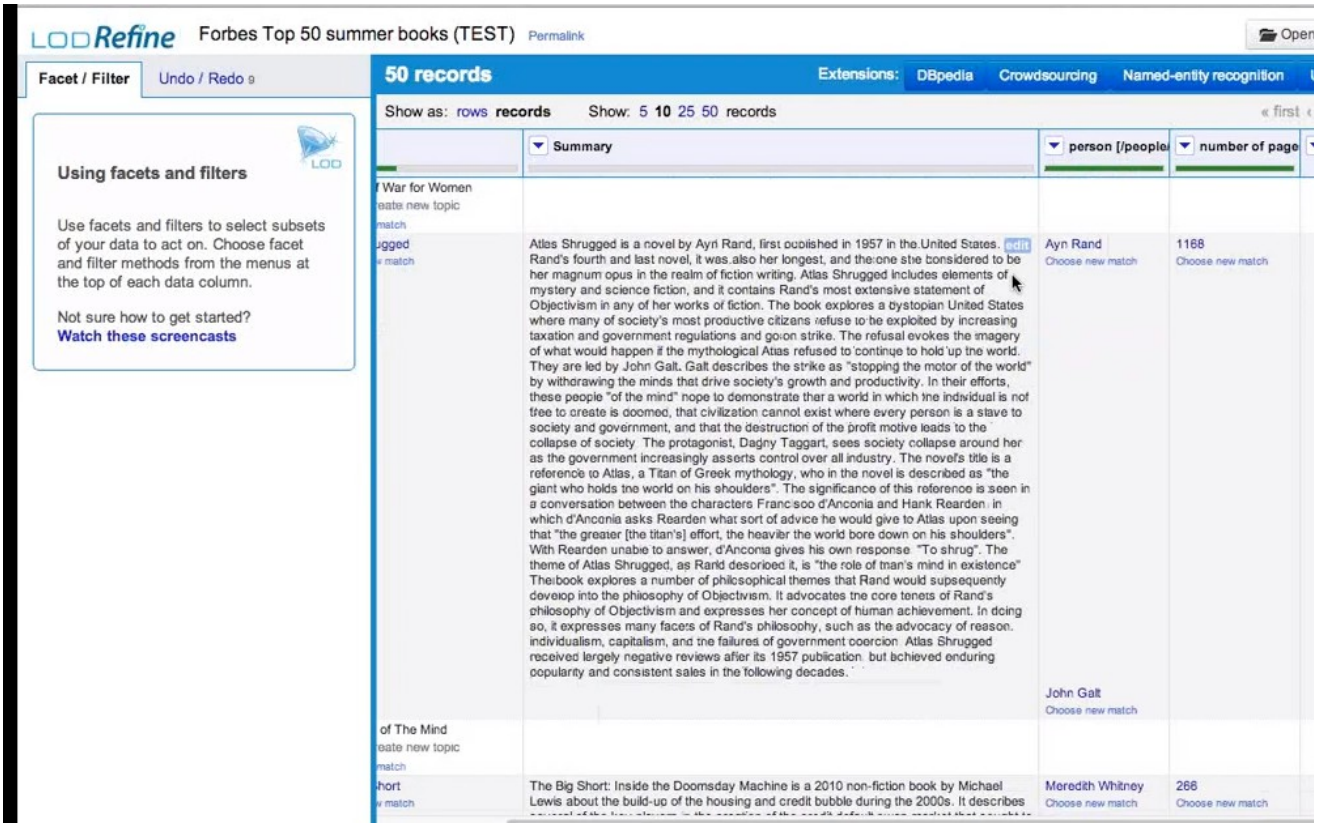


Illustration 2: Zemanta NER plugin example, enriched dataset

The important thing to notice in Illustration 2 is that the description text is not enriched with labels, so it is not possible to find at a glance where the two names are located. The plugin by FreeYourMetadata produces a similar result, while allowing to link entities against 4 services at the same time (AlchemyAPI, DBPedia Spotlight, Zemanta and dataTXT). However, it doesn't allow to restrict entity types as Zemanta extension.

We would like to improve the situation by allowing to perform not only NER but also WSD (actually the service of DBPedia Spotlight included in FreeYourMetaData plugin already spots a pool of 300 concepts). Also, we will provide labels directly on the identified words.

8 Terminology

Catalog	A data management system for publishing datasets
CKAN	CKAN is a catalog that provides tools to streamline publishing, sharing, finding and using data
Dataset	A group of *dataset resources*. Both DCat and CKAN use the concept of *dataset*
Dataset resource	A file in a *dataset*, in the terminology of CKAN
DCAT	Data Catalog Vocabulary - DCAT is an RDF vocabulary designed to facilitate interoperability between catalogs published on the Web
Distribution	A file in a *dataset*, in the terminology of Dcat
NED	Named Entity Disambiguation - a process which takes as input text enhanced by NER and links each entity name to a unique identifier
NER	Named Entity Recognition - an NLP process for identifying entity names in natural language text
NLP	Natural Language Processing - Automated processing of text in natural language to extract information out of it
RDF	Resource Description Framework - a general method for conceptual description or modeling of information suited for web resources
Semantification Process	A sequence of operations to clean and enrich raw data. It may be stored in Semantification process files
WSD	Word Sense Disambiguation - an NLP process for identifying which sense of a word (i.e. meaning) is used in a sentence when the word has multiple meanings