Using GREL (regular expressions) in OpenRefine

For a general introduction to Regular Expressions, see Understanding Regular Expressions | Programming Historian, and lecture 5 of the course. For an explanation of the difference between greedy and lazy quantifiers, see: Greedy and lazy quantifiers (javascript.info). Remember to use https://regex101.com/ to do some try-outs.

click on the icon of the column in which you want to modify some text, then "Edit cells", then transform.

1. Replace a value with another

    replace(value, /a{2}/, "ab").

Takes the "value" found in the column, and replaces every sequence of two "a"s with the sequence "ab". Example "aaron" -> "abron"

    replace(value, /(\d\d)\d/, "$1")

Takes the value found in the column, and replaces every sequence of three digits with just the two first. Example "543" -> "54"

2. Condition on the replacement

if(value=='good', "super", value)

If a certain condition is met (value=='good', first argument of the function), replace it with "super" (second argument of the function); if not, keep the value as it is (third argument of the function)

3. Capture all the characters between two other characters:

replace(value, /9903(.*) be/, "$1")

replaces the sequence "9903"+other characters+"be" with just the "other characters" between the twos.

Example: "9903**25s1559** be |||| | 000|0 lat c" -> "25s1559 |||| | 000|0 lat c"

Other Openrefine specific resources for regular expressions:

- https://docs.oracle.com/javase/tutorial/essential/regex/ (this is the language that OpenRefine supports)
- https://docs.openrefine.org/manual/expressions#regular-expressions
- https://docs.openrefine.org/manual/grelfunctions
- https://gist.github.com/pmgreen/6e133c5dcde65762d29c
- http://www.rexegg.com/regex-quickstart.html