PACKED
Expertisecentrum Digitaal Erfgoed

Handout Workshop Open Refine

Adlibgebruikersgroep

30.04.2015

Alina Saenko | PACKED vzw
Inge van Stokkom | Rijksmuseum

Inhoud:

Deel 1. Introductie

1.1 Installeren

http://openrefine.org/

1.2 Dataset importeren in Open Refine

Create Project -> Choose Files -> Next -> Character encoding: 'UTF-8' -> Parse data as (kies het juiste bestandsformaat) -> Voor csv-import: Columns are separated by > commas (CSV) -> Vul in een Project name -> Create Project

1.3 Look & Feel

Verschillende manieren om je tabel te manipuleren:

- Kolommen / cellen
- Show as: rows/records
- Show: 5 10 25 50 records
- « first ‹ previous 1 - 10 next › last »

Eerste manipulaties

- Facet&Filter
- Undo/Redo
- Sort

1.4 Onnodige kolommen verwijderen

All > Edit columns > Re-order/remove columns -> Drag & drop columns > OK

1.5 Onnodige rijen verwijderen

Stel in: Show as: rows
Maak gebruik van facets en filters om een keuze te maken

Voorbeeld 1:

- Zet een ster tegenover rijen die je wilt verwijderen
- Kies All -> Facet by star -> Select True
- All > edit rows > remove all matching rows

Voorbeeld 2:

- Kies kolom Gemeente -> Text Facet -> Kies 'Brussel'
- All > edit rows > remove all matching rows

Deel 2. Schonen

2.1 Waardes aanpassen (Grel expressions)

Voorbeeld: Vervang ongewenste schrijfwijze in kolom ‘Materiaal’
Kies kolom -> Edit cells -> Transform -> Vul in de GREL expression: value.replace('potlood en inkt','potlood, inkt')

Let op! cursieve tekens in de functie worden als een fout aangegeven
Overzicht GREL functies: https://github.com/OpenRefine/OpenRefine/wiki/GREL-Functions

2.2 Concepten samenvoegen (Cluster and edit)

Voorbeeld: Cluster and edit - in kolommen ‘Ontwerpers’ en ‘Type Gebouw’
Kies kolom met termen-> Edit cells -> Cluster and edit -> Kies de nodige waardes -> Klik op Merge Selected

Deel 3. Normaliseren en verrijken

Er zijn in ieder geval vier methodes voor ‘reconciliation’ - geautomatiseerd de eigen termen met die van een autoriteit matchen. Ten eerste kan de autoriteit als lokaal bestand geüpload en vervolgens gekoppeld aan de eigen thesaurus worden (3.1). Er kan gebruik gemaakt worden van een speciaal opgezette reconciliation service, zoals Packed heeft gedaan voor de ODIS/RKD (3.2). Ten derde kunnen via de optie “add column by fetching URLs” gegevens automatisch opgehaald worden bij webservices zoals Openstreetmap.org (geografische coördinaten), Geonames.org (plaatsnamen), Wikidata, Viaf, en de Getty thesauri (AAT, TGN). Daarnaast is er een ‘extension’ van OpenRefine ontwikkeld (DERI RDF Extension), waarmee de connectie met een autoriteit gemaakt kan worden. Dit staat verder uitgelegd op de bijzonder informatieve website Freeyourmetadata.org (een aanrader!) en wij gaan er hier niet verder op in.

3.1 Via een export - twee tabellen linken

- Upload *AAT* bestand in OpenRefine

- Werk verder in het project ‘Adlibgebruikersgroep_workshop’
Maak een nieuwe kolom voor AAT-ids: Kies kolom ‘Meubel’ > ‘Edit column’ > ‘Add column based on this column’ -> Enter a new column name ‘’AAT-id’ -> Laat GREL expression op ‘value’ -> OK

Kies kolom ‘AAT-id’ > ‘Edit cells’ >Transform > vul in de GREL expression: cell.cross("AAT", "record - uf").cells["record - recordId"].value[0] -> OK

3.2 Reconciliation service

Op voorbeeld van door PACKED opgestelde ODIS - reconciliation service

-*Kies kolom met de oorspronkelijke waardes -> Edit column -> Add column based on this column -> New column name: vb. 'ODISrecon'*

-Reconciliation service toevogen: *Kies 'Reconcile' kolom -> Edit cells -> Reconcile -> Start reconciling -> Add standard service -> Enter the service's URL:*

•Voor ODIS: http://projects.packed.be/reconciliation/reconciliation-generic/reconciliate_odis.php

-*'Auto-match candidates with high confidence' uitvinken -> Start reconciling*

-Kies van de voorgestelde resultaten de juiste

-Haal de id en de term van de beste kandidaat op uit de gereconciled data:

Kies kolom met persoonsnamen -> Edit column -> Add column based on this column -> New column name: 'ODISid' -> vul in een GREL expression: cell.recon.match.id

Kies kolom met persoonsnamen -> Edit column -> Add column based on this column -> New column name: 'ODISname' -> vul in een GREL expression: cell.recon.match.name

>> screenshot van RKD

3.3 Gegevens ophalen uit VIAF

Identificeren met VIAF

- Kies kolom met ontwerpers -> Edit column -> Add column by fetching URLs -> New column name: 'VIAFjson' -> vul in de GREL expression: 'http://viaf.org/viaf/AutoSuggest?query=' + escape(value, 'url') -> *optioneel wachttijd verminderen:* zet Throttle delay op 1000 milliseconds -> OK

- Haal VIAF id: Kies kolom 'VIAFjson' -> Edit column -> Add column based on this column -> New column name: 'VIAFid' -> vul in de GREL expression: value.parseJson().result[0].viafid

- Haal VIAF term: Kies kolom 'VIAFjson' -> Edit column -> Add column based on this column -> New column name: 'VIAFname' -> vul in de GREL expression: value.parseJson().result[0].term

- Maak persistente URI's ahv VIAF id's: Kies kolom 'VIAFid' -> Edit column -> Add column based on this column -> New column name: 'VIAFuri' -> vul in de GREL expression: 'http://viaf.org/viaf/' + value

Verrijking met geboorte- en sterfdatum uit VIAF

- Kies kolom ‘VIAFid’ -> Edit column -> Add column by fetching URLs -> New column name: ‘VIAFxml’ -> vul in de GREL expression: 'http://viaf.org/viaf/' + escape(value,'url') + '/viaf.xml' -> zet Throttle delay op 1000 milliseconds -> OK

- Kies kolom ‘VIAFxml’ -> Edit column -> Add column based on this column -> New column name: ‘VIAFbirth’ -> vul in de GREL expression: value.parseHtml().select('ns2|birthDate')[0].ownText()

- Kies kolom ‘VIAFxml’ -> Edit column -> Add column based on this column -> New column name: ‘VIAFdeath’ -> vul in de GREL expression: value.parseHtml().select('ns2|deathDate')[0].ownText()

3.4 Gegevens ophalen uit AAT

- Het volgende stappenplan haalt de AAT-identifier aan de hand van Nederlandse termen, om vervolgens daarmee de Engelse termen op te halen. Voor uitgebreider stappenplan zie de (verder ook zeer nuttige) website Semantic Web
- Haal json op bij het Getty: Create column by fetching URLs -> 'http://vocab.getty.edu/sparql.json?query=select%20*%20{?x%20skos:inScheme%20aat:;%20%28xl:prefLabel|xl:altLabel%29/gvp:term%20%22' + escape(value, 'url') + '%22@nl}'
- Parse de url uit de json met -> Add column based on this column -> value.parseJson().results.bindings[0].x.value
- Parse de identifier uit de url -> value[27,37]
- Haal de Engelse term op met -> Add column by fetching URLs op de identifier-kolom -> 'http://vocab.getty.edu/sparql.json?query=select+*+where%0D%0A{%0D%0A++%3Fx+gvp%3AprefLabelGVP+[skosxl%3AliteralForm+%3Flabel]%3B%0D%0A++dc%3Aidentifier+%22' + escape(value, 'url') + '%22%0D%0A+++++}&_implicit=false&implicit=true&_equivalent=false&_form=%2Fsparql'
- Parse de json om de Engelse term eruit te halen met: -> value.parseJson().results.bindings[0].label.value

3.5 Gegevens ophalen uit Wikidata

- Kies de kolom met namen -> Edit column -> Add column by fetching URLs -> name ‘wikidata’ -> vul in de GREL expression: "http://www.wikidata.org/w/api.php?action=query&list=search&format=json&srwhat=text&srinfo=totalhits&srprop=titlesnippet&srlimit=1&srbackend=CirrusSearch&srsearch=" + value

- Haal wikidata ID: Kies kolom ‘wikidata’ -> Edit column -> Add column based on this column -> New column name: vb. ‘wikidataID’ -> vul in de GREL expression: forEach(value.parseJson().query.search,v,v.title).join(";")”

- Maak persistente URI's: Kies kolom 'wikidataID' -> Edit column -> Add column based on this column -> New column name: 'wikidataURI' -> vul in de GREL expression: value.replace("Q",http://www.wikidata.org/wiki/Q)

3.6 Gegevens ophalen uit Geonames

- Kies de kolom met namen -> Edit column -> Add column by fetching URLs -> name 'Geonames' -> vul in de GREL expression: 'http://api.geonames.org/search?q=' + value + '&maxRows=10&username=demo'

*Tip: 'Demo' is voor een beperkt gebruik. Voor meer mogelijkheden creer een user account voor jezelf: http://www.geonames.org/login
Op het einde van de GREL expression gebruik dan je username ipv 'demo'

- Haal Geonames ID: Kies kolom 'Geonames' -> Edit column -> Add column based on this column -> New column name: vb. 'GeonamesID' -> vul in de GREL expression: value.parseHtml().select('geonameId')[0].ownText().toNumber()

- Maak persistente URI's: Kies kolom 'GeonamesID' -> Edit column -> Add column based on this column -> New column name: 'GeonamesURI' -> vul in de GREL expression: 'http://www.geonames.org/' + value

Deel 4. Export

4.1 Export voor adlib

Data exporteren uit OpenRefine

Dataset in Adlib-tagged-bestand veranderen.

- Splits de kolommen waarin meerdere occurences voorkomen, met Edit column: split column into several columns.
- Voeg de tag van de kolom toe aan elke occurence, met Transform 'toString("tag " + value)' (bijv. "TK" + value)
- Voer bij elke kolom de acties Edit cells -> common transforms -> trim leading and trailing whitespace én -> collapse consecutive whitespace
- Voeg een extra kolom toe met in elke cel "**", via Edit column, add column based this column "**".
- Maak van de kolommen rijen, door op de eerste kolom te kiezen: Transpose cells across columns into rows, to 2 columns, Key column = "naam1", Value column = "naam2"
- Verwijder kolom "naam1" (de oude kolomnamen)
- Export: csv/tsv, opslaan als .dat bestand via een teksteditor zoals Notepad++
- Haal in Notepad de kolomnaam boven de enig overgebleven kolom weg