OpenRefine for Libraries

Maike Kittelmann

DINI KIM
Workshop 2016

Inhalt

1 Messy Data
2 OpenRefine
3 Vorwäsche
4 30° Wäsche
5 Handwäsche
6 40° Wäsche
7 Handwäsche
8 60° Wäsche
9 Handwäsche
10 95° Wäsche
11 Handwäsche
12 Nicht bügeln
13 Handwäsche
14 Bleichen
15 Handwäsche
16 Trocknen
17 Nicht waschbar

Themen

Vorwäsche:	Installation
30° Wäsche:	Datenimport, Ansicht
40° Wäsche:	Filter, Facets, Cluster
60° Wäsche:	GREL
95° Wäsche:	Project History, Backup
Nicht bügeln:	Inhalt exportieren
Bleichen:	Reconciliation, RDF, NER
Trocknergeeignet:	REST API
Nicht waschbar:	Info

Messy Data

- Tippfehler
- ungültige Werte
- fehlende Werte
- unklare Werte
- Werte im falschen Feld
- mehrere Werte pro Feld
- verletzte Abhängigkeiten
- widersprüchliche Werte
- verschiedene Aggregatlevel
- verschiedene Bezugseinheiten
- Dubletten
- …

Messy Data

Most of us know that the term 'metadata architect' rareley matches the reality. 'Digital landfill manager' sounds less glamorous but reflects the job content more adequately.

[Source: Linked Data for Libraries, Archives, and Museums]

Messy Data

Messy Data

What it should be like . . .

OpenRefine

Daten sichten, aufräumen, anreichern

- Reguläre Ausdrücke
- Undo / Redo
- Importformate
- Exportformatierung
- Reconciliation
- Extensions (RDF, NER)

Hauptzielgruppen

- Open Data
- Journalisten

Datenvolumen

Allocate more memory 

https://github.com/OpenRefine/OpenRefine/wiki/FAQ:-Allocate-More-Memory

Grenzen der Datenkapazität

4 GB RAM -> max. 512 MB Daten [P. Larsson, Juni 2013]
(entsprechend höher bei mehr Arbeitsspeicher)

Installation

- un-zip
- .exe
- Kommandofenster geöffnet lassen
- http://127.0.0.1:3333/
- öffnet sich im Standardbrowser
- Chrome

Allocate memory

- Java Virtual Machine
- google-refine.l4j.ini
- initial memory heap size
- -Xms512M
- -Xms2048M

Allocate memory

Chrome Timeout

Chrome Timeout $\neq$ OpenRefine Timeout
$\Rightarrow$ *Warten*

Workspace

- Open > Browse workspace directory
- Backup

Extensions installieren

- nicht im Programmverzeichnis
- Browse workspace directory > extensions
- Ordner extensions neu anlegen, wenn er nicht existiert


30

Datenimport

- Formate
- Zeichenkodierung
- URLs
- mehrere Dateien

Customizing der Ansicht

View

- Collapse column(s)
- Doppelclick zum Wiederanzeigen

Customizing der Ansicht

Edit column

- Rename
- Move left / right / beginning / end
- Remove = löschen (nicht zu verwechseln mit collapse)

Customizing der Ansicht

All

- Re-order / remove columns

Customizing der Ansicht

- Projekt umbenennen

Sort

- Sort options
- Reorder rows permanently

- Importieren Sie die Tabelle von
  http://tinyurl.com/mvfkbrg
  über Clipboard
- Importieren Sie die Daten
  http://book.freeyourmetadata.org/chapters/4/british-library.csv
- Importieren Sie die Daten von
  https://gist.github.com/acka47/9bdc24359fe811e90026
  über Web Adresses (URLs) (Character Encoding!)


40

Funktionen

- Text filter (RegEx!)

Funktionen

- Facets
- Cluster

Facets

Overview

- Text facet
- Customized facet
- Custom text facet (click on number of choices)

Facets

Funktionen

- Filter by facet
- Edit facet
- Refresh
- Reset All
- Sort by count
- Set choice count limit
- Facet Choices as Tab Separated Values (click on number of choices)

Facets

by star / by flag

- Facet by star
- Facet by flag

Records

Multivalued cells

- Edit cells > Split multivalued cells
- Edit cells > Join multivalued cells

Records

Multivalued cells

- Rows vs. Records

Clustering

- Cluster Size
- Row Count
- Use value
- Merge?
- Select / Unselect All
- Browse this cluster
- Merge Selected & Re-Cluster
- Merge Selected & Close
- Close

Clustering

Method

- Key Collision Methods
  - + schnell
  - - wenig Feintuning
- Nearest Neighbor Methods
  - - langsam
  - + genau skalierbar

Clustering

Method

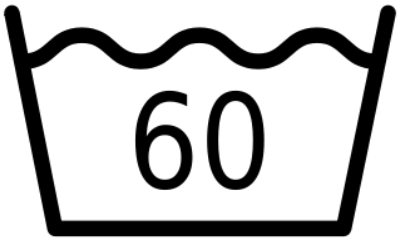
https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth

Edit column

Split into several columns

- Edit column > Split into several columns
- British Library example data, Spalte: extent
- Facet > Customized Facets > Text Length Facet *oder*
  Facet > Custom text facet mit length(value)

- Bilden Sie eine Texfacette über Species Group und sortieren Sie nach Count! Aus welcher Gruppe stammen die meisten Tiere in der Liste? Aus welche die wenigsten?
- Bilder Sie eine Texfacette über Common Name. Bilden Sie Cluster mit den Methoden Fingerprint und ngram-fingerprint mit den Werten 1 und 2. Was fällt Ihnen auf?
- Finden Sie die Funktion Browse this cluster!
- Mergen Sie die unterschiedlichen Schreibweisen der Ringelrobbe (Ringed Seal) in Common Name mit der Cluster-Funktion
- Bilden Sie eine Dublettenfacette über Scientific Name.
- Sortieren Sie permanent nach Common Name.
- Nutzen Sie den Textfilter. In welcher Zeile ist das Schwarzfuß-Frettchen (Black-footed ferret)?


60

GREL

General Refine Expression language

- Variables
- GREL Controls
- GREL Functions overview
- GREL Boolean Functions
- GREL String functions, including parsing, splitting, encoding and hashing
- GREL Array functions
- GREL Math functions
- GREL Date functions
- GREL Other functions including JSON and Jsoup

GREL

General Refine Expression language

https://github.com/OpenRefine/OpenRefine/wiki/General-Refine-Expression-Language

GREL

General Refine Expression language

- Edit cells > Common Transforms
- Edit cells > Transform
- Edit column > Add column based on this column

GREL

Edit cells > Transform *bzw.* Edit column > Add column based on this column

- Preview
- History
- Starred
- Help

GREL

Edit cells > Transform *bzw.* Edit column > Add column based on this column

- Ausdruck GREL / Jython / Clojure

GREL

Variables

value	the value of the cell in the base column of the current row; can be null
row	the current row; an object with more fields, with details below
cells	the cells of the current row, with fields that correspond to the column names; more details below
cell	the cell in the base column of the current row; an object with more fields, with details below
recon	the recon object of a cell returned from a reconciliation service or provider; an object with more fields, with details below
record	one or more rows grouped together to form a record; an object with more fields, with details below

GREL

replace()

```
value.replace(/exp-or-str/, 'exp-or-str')

replace( value, /exp-or-str/, 'exp-or-str')
```

GREL

replace()

```
value.replace('Google', 'Open')

value.replace(/Google/, 'Open')

value.replace(/Google(Refine)/, '$1')
```

GREL

escape() / unescape()

```
escape(value, 'html')
```

- html, xml, csv, url, javascript
- Edit cells > Common Transforms > Unescape HTML entities

GREL

trim()

```
value.trim()
trim( value )
```

- Edit cells > Common transforms > Trim leading and trailing whitespace

GREL

length()

```
value.length()
length( value )
```

- Facet > Customized Facets > Text Length Facet oder Facet > Custom text facet mit length(value)

GREL

HTML Parsing mit JSoup

- https://commons.wikimedia.org/wiki/Library
- value.parseHtml().select('img').toString()
- forEach(value.parseHtml().select('img'), v, v.htmlAttr('src')).join('|')
- forEach(value.parseHtml().select('a[href]'), v, 'http://ecos.fws.gov' + v.htmlAttr('href')).join('|')
- https://github.com/OpenRefine/OpenRefine/wiki/StrippingHTML
- http://jsoup.org/cookbook/extracting-data/selector-syntax

- Erstellen Sie eine neue Spalte mit generierten Links, indem Sie Common Name html-escapen und an https://www.google.com/?q= anhängen (Auf diesem Weg können Sie auch URIs für Identifier erstellen.)
- Splitten Sie die multivalued cells in der Where Listed, um bei der Facettierung die korrekten Zahlen zu bekommen
- Importieren Sie die Webseite http://tinyurl.com/mvfkbrg (hinterlegten Link nutzen!) über Edit column > Add column by fetching URLs
- Extrahieren Sie die Links mittels forEach(value.parseHtml().select('a[href]'), v, 'http://ecos.fws.gov' + v.htmlAttr('href')).join('|')


95

Project History

- Undo / Redo
- Extract
- Apply
- Schritte in Extract auswählbar → Arbeitsschritte aufräumen

Project History

- Speichern aller Arbeitsschritte als json-Parameter
- Teilautomatisierung mit Apply
- Dokumentation

Backup

- Export project
- Import project
- .tar.gz

Backup

Nicht vergessen den Workspace-Ordner zu sichern!
Bei Export über Export Project Vorsicht bzgl. der Project History

Backup

Project-Dateien im Workspace

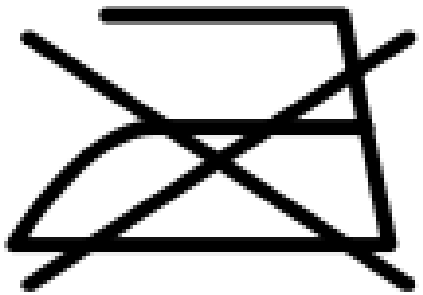
- metadata.json
- Projektordner in Workspace legen und OpenRefine neu starten
- http://127.0.0.1:3333/project?project=[project_id]
- [project_id] steht für die Nummer des Projekt-Ordners

Backup

Tools

http://www.7-zip.de/
https://www.microsoft.com/en-us/download/details.aspx?id=15155

- Exportieren Sie die History aller bisherigen Schritte und speichern Sie sie als operations.json
- Erstellen Sie über Open > Create Project ein neues Projekt mit denselben Daten und wenden Sie alle bisherigen Schritte darauf an
- Erstellen Sie ein Projekt-Backup und Re-importieren Sie es



Datenexport

- diverse voreingestellte Formate
- Custom Tabular Exporter
- Templating → konvertieren via Export

Datenexport

Custom Tabular Exporter

- Option Code > Apply
- als json speichern und wiederverwenden

Datenexport

Templating

- Default: json
- mit {{ . . . }} auf GREL zugreifen
- jsonize()-Funktion setzt Werte in Anführungszeichen
- alle GREL-Objekte und Funktionen anwendbar

Datenexport

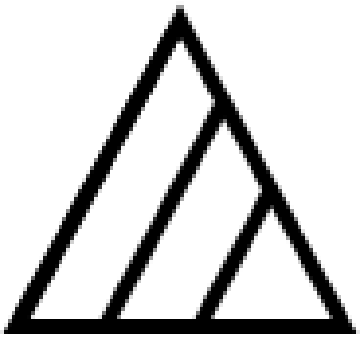
Templating - Beispiele

- z.B. nur eine Spalte exportieren
- z.B. MODS

Datenexport MODSXML

```
<!-- For Prefix -->

<?xml version="1.0" encoding="UTF-8"?>
<modsCollection xmlns="http://www.loc.gov/mods/v3"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.loc.gov/mods/v3 http://www.loc.gov/standards/mods/v3/mods-3-4.xsd">

<!-- For Row Template -->

<mods xmlns="http://www.loc.gov/mods/v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.loc.gov/mods/v3 http://www.loc.gov/standards/mods/v3/mods-3-4.xsd">

      <titleInfo>
         <title>{{jsonize(cells["Title"].value)}}</title>
      </titleInfo>
      <name>
         <namePart>{{jsonize(cells["Name"].value)}}</namePart>
         <role>
            <roleTerm type="text">{{jsonize(cells["Role"].value)}}</roleTerm>
         </role>
      </name>
      <genre>{{jsonize(cells["Genre"].value)}}</genre>
      <subject>
         <topic>{{jsonize(cells["Subject"].value)}}</topic>
      </subject>
      <note>{{jsonize(cells["Note"].value)}}</note>
      <originInfo>
         <publisher>{{jsonize(cells["Publisher"].value)}}</publisher>
      </originInfo>
      <originInfo>
         <dateCreated>{{jsonize(cells["Date"].value)}}</dateCreated>
      </originInfo>
      <physicalDescription>
         <extent>{{jsonize(cells["Size"].value)}}</extent>
      </physicalDescription>
      <identifier type="local">{{jsonize(cells["Identifier"].value)}}</identifier>
      <language>
         <languageTerm type="text">{{jsonize(cells["Language"].value)}}</languageTerm>
      </language>
      <accessCondition>{{jsonize(cells["Rights"].value)}}</accessCondition>
   </mods>

<!-- Don't put anything in Row Separator -->

<!-- For Suffix -->

</modsCollection>
```

- Speichern Sie die Parameter des Custom Tabular Exports und wenden Sie beim nächsten Aufruf wieder an.
- Exportieren Sie über Templating zwei Spalten Ihrer Wahl als csv (csv gibt es auch als voreingestellte Funktion) oder xml.



Reconciliation

- Linking and enriching data
- Datenanreicherung über Matchen von Einzeltermen

Reconciliation

- Freebase → deprecated
- Sindice - the Semantic Web Search Engine → service currently not available

Reconciliation

Standard Service: LOBID

- Reconcile > Start Reconciling > Add Standard Service
- http://beta.lobid.org/organisations/reconcile
- Beispieldaten: https://gist.github.com/acka47/9bdc24359fe811e90026

Credits Beispiel: Adrian Pohl, HBZ

Reconciliation

Explore results

Recon

A `recon` object has a few fields

field name	meaning	deeper fields
`recon.judgment`	a string that is one of: "matched", "new", "none"	
`recon.matched`	a boolean, true if judgment is "matched"	
`recon.match`	null, or the recon candidate that has been matched against this cell	`.id` `.name` `.type`
`recon.best`	null, or the best recon candidate	`.id` `.name` `.type` `.score`
`recon.features`	an object encapsulating reconciliation features	`.typeMatch` `.nameMatch` `.nameLevenshtein` `.nameWordDistance`
`recon.candidates`	an object encapsulating the default 3 candidates	`.id` `.name` `.type` `.score`

Reconciliation

Explore results

Zugriff auf das recon-Object in der benutzten Spalte:

- cell.recon.best.score
- length(cell.recon.candidates)
- cell.recon.best.name
- cell.recon.best.id (→ was wir haben wollen)

Zugriff auf das recon-Object aus anderen Spalten:

- cells['bibliothek'].recon.best.id

Reconciliation

Standard Service: VIAF

- Reconcile > Start Reconciling > Add Standard Service
- http://refine.codefork.com/reconcile/viaf

Reconciliation

Übersicht über weitere Reconciliation Services

- https://github.com/OpenRefine/OpenRefine/wiki/Reconcilable-Data-Sources

Reconciliation

How to program your own reconciliation service

- https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API

Reconciliation

Custom Reconciliation to csv

- http://okfnlabs.org/reconcile-csv/
- comma separated value
- java -Xmx2g -jar reconcile-csv-0.1.2.jar <CSV-File> <Search Column> <ID Column>
- Search Column = Spalte, über die gematcht werden soll
- ID Column = Unique identifier
- java -Xmx2g -jar reconcile-csv-0.1.2.jar bib_ohne_sigel.csv bliothek dbs-id
- Reconcile > Start reconciling > Add standard service
- http://localhost:8000/reconcile

Bei genauen Matches können Sie auch die GREL cross Funktion verwenden. https://github.com/OpenRefine/OpenRefine/wiki/GREL-Other-Functions#crosscell-c-string-projectname-string-columnname

Reconciliation

rdf-extension: LSCH

- RDF > Add Reconciliation Service > Based on SPARQL endpoint
- Name: LCSH
- Endpoint URL: http://sparql.freeyourmetadata.org/
- Type: Virtuoso
- Label properties: skos:prefLabel

Reconciliation

rdf-extension: Beispielreconciliation LCSH

- Example: Listed Animals, Common Name
- Reconcile > Start Reconciling > LCSH

Reconciliation

rdf-extension: rdf-Datei

- Reconciliation auch aus lokaler rdf-Datei
- RDF > Add Reconciliation Service > Based on RDF file
- Upload file / Load file from URL

Named Entity Recognition

ner-extension

- Named Entity Recognition
- = Extraktion von Named Entities aus Fließtext

Named Entity Recognition

ner-extension

- DBpedia Spotlight, frei zugänglich
- Erfordert Registrierung, für wissenschaftliche Zwecke in gewissen Grenzen kostenlos:

→ Alchemy http://www.alchemyapi.com/api/register.html

→ Dandelion https://dandelion.eu/accounts/register/?next=/semantic-text/entity-extraction-demo/

→ Zemanta

- Comparative evaluation of services (2013): http://freeyourmetadata.org/publications/named-entity-recognition.pdf

Named Entity Recognition

ner-extension

- Extract named entities > Start extraction

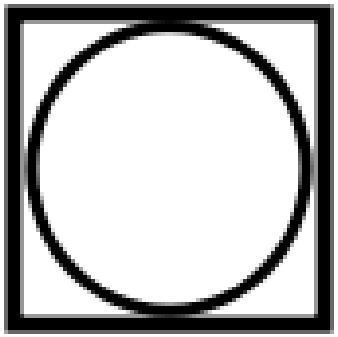
Anleitung

Scaffold für Ihre eigene Extension

- Write your own extensions
  https://github.com/OpenRefine/OpenRefine/wiki/Write-An-Extension
- Sample Extension
  https://github.com/OpenRefine/OpenRefine/wiki/Sample-Extension

- Speichern Sie http://data.nytimes.com/organizations.rdf lokal auf dem Rechner
- Legen Sie unter RDF > Add Reconciliation Service > Based on RDF file > Upload file einen lokalen Reconciliation Service an
- Erstellen Sie ein Projekt aus den Daten unter https://docs.google.com/spreadsheets/d/1jTWHot6Kn2QR-AuW599_szNZoV58mp-Tcugqhqqv_ic/edit?usp=sharing
- Starten Sie eine Reconciliation

Credit Beispiel: http://refine.deri.ie/



REST API

Automatisierung

- → Extract / Apply
- REST API
- cURL
- Libraries: http://openrefine.org/download.html (ganz unten)
- API Dokumentation: https://github.com/OpenRefine/OpenRefine/wiki/OpenRefine-API

API 

nicht stable!

REST API

refine-ruby

- https://github.com/mkittelmann/refine-ruby (forked)
- Unit-Tests und Anwendungsbeispiele unter /test
- Ruby >= 1.9.3

REST API

refine-ruby

- git clone https://github.com/mkittelmann/refine-ruby.git
- oder Download als .zip

REST API

refine-ruby

examples_of_usage.rb

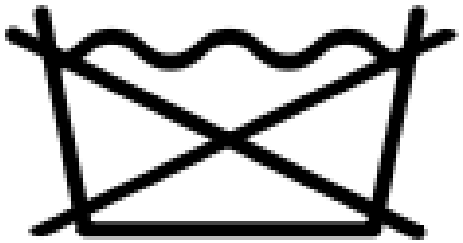
```ruby
### NOTE: The internal client-server protocol used by OpenRefine is not yet maintained as a stable external API, subject to change. ###
### Therefore, plase indicate changes you notice to kittelmann@sub.uni-goettingen.de ###
### Some examples require cURL http://curl.haxx.se ###
### It is assumed that examples are run from the 'test' directory. Otherwise paths need to be adjusted.
load '../lib/refine.rb'

##########################
### create initial project
##########################
prj = Refine.new({ 'project_name' => 'date_cleanup', 'file_name' => 'dates.csv' })


##########################
### create another project
##########################
prj.create_project( 'date_cleanup', 'dates.txt' )                    # return value = project id, example: 1484090391100


################
### do something
################
prj.apply_operations( 'operations.json' )                            # return value = status code, example: {'code'=>'ok'}


######################
### extract operations
######################
prj.get_operations                                                    # return value = operations as Hash

####################################
### save extracted operations to file:
####################################
extracted_operations = prj.get_operations
File.open('../test/extracted_operations.json', 'w') do |f|
  f.write extracted_operations
end


###############
### export data
###############
prj.export_rows                                                       # return value = exported data as tsv
prj.export_rows( {'format'=>'tsv'} )                                  # return value = exported data as tsv
prj.export_rows( {'format'=>'csv'} )                                  # return value = exported data as csv
```

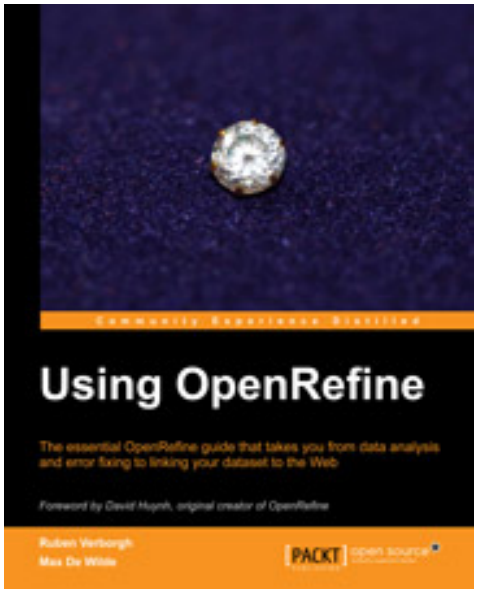
REST API

refine-ruby

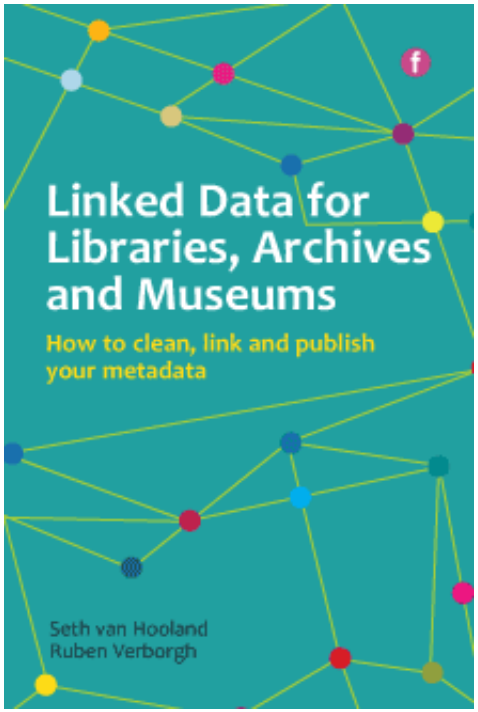
test_refine.rb

```ruby
gem 'minitest'
require 'minitest/autorun'
require_relative '../lib/refine.rb'

class TestRefine < MiniTest::Unit::TestCase

  def setup
    @refine_project = Refine.new({ "project_name" => 'date_cleanup', "file_name" => '../test/dates.txt' })
  end

  def test_refine_initializer_has_instance_variable_project_name
    assert_equal 'date_cleanup', @refine_project.project_name
  end

  def test_refine_initializer_has_instance_variable_project_id
    assert @refine_project.project_id.match(/^[0-9]+$/)
  end

  def test_get_all_project_metadata
    assert Refine.get_all_project_metadata.instance_of? Hash
  end

  def test_apply_operations
    assert @refine_project.apply_operations( '../test/operations.json' )
  end

  def test_call
     assert @refine_project.call( 'apply-operations', 'operations' => File.read( 'operations.json' ) )
  end

  def after_tests
    @refine_project.delete_project
  end

end
```





http://openrefine.org/



http://book.freeyourmetadata.org/

Anwendungsbeispiele

Metadaten

- A complete example of how to create linked data (OpenRefine in Teil 2: Creating RDF) http://wifo5-03.informatik.uni-mannheim.de/latc/toollibrary/screencast.html
- Erstellung wiederverwendbarer RDF-Geodaten mit Google Refine https://journals.ub.uni-heidelberg.de/index.php/ip/article/view/23784/18989
- Data Munging Tools in Preparation for RDF: Catmandu and LODRefine http://journal.code4lib.org/articles/11013
- MODSXML http://www.utsc.utoronto.ca/digitalscholarship/content/blogs/converting-spreadsheets-modsxml-using-open-refine

Bildnachweis

- Miele 3240, https://commons.wikimedia.org/wiki/File:Mielew3240.jpg, Sanekmoskow
- OpenRefine Logo, http://openrefine.org
- Laundry Symbols, Public Domain, https://commons.wikimedia.org/wiki/Category:Laundry_symbols
- Dirty Laundry, http://tommy.ismy.name/wp-content/uploads/2011/06/dirty-laundry.jpg
- Washings whites bright in the sun, https://commons.wikimedia.org/wiki/File:Peru_-_Salkantay_Trek_104_-_clothesline_%28734317481 6%29.jpg, Magnus Manske

Danke für Ihre Aufmerksamkeit!