Automatisierte Datenverarbeitung mit OpenRefine

Hands-On-Tutorial, DINI AG KIM Workshop am 2./3.4.2019 in Mannheim

Teil 1: OpenRefine über die Kommandozeile steuern

Im ersten Teil des Hands-On-Tutorials steuern wir OpenRefine über die Kommandozeile. Dazu verwenden wir auf unseren PCs einen Client, der als Programm für Windows, MacOS und Linux bereitsteht und ohne weitere Installation ausgeführt werden kann.

Installation (14:40-14:50)

Download der Daten von https://github.com/felixlohmeier/openrefine-kimws2019

- Grüner Button `Clone or download` > `Download ZIP`
- Zip-Archiv entpacken und den Ordner `openrefine-kimws2019-master` z. B. auf den Desktop schieben

Kommandozeile starten und in den Ordner wechseln

- Windows: Programm Eingabeaufforderung öffnen und in den heruntergeladenen Ordner wechseln
  - Beispiel: `cd Desktop\openrefine-kimws2019-master`
- MacOS/Linux: Terminal öffnen und in den heruntergeladenen Ordner wechseln
  - Beispiel: `cd Downloads/openrefine-kimws2019-master`

Da die Startdatei des Programms openrefine-client auf jedem System anders heißt, steht im Folgenden *openrefine-client* stellvertretend für

- Windows: `openrefine-client_0-3-4_windows.exe`
- MacOs: `./openrefine-client_0-3-4_mac`
- Linux: `./openrefine-client_0-3-4_linux-64bit`

Zum Test die Projekte auf OpenRefine Server listen

- *openrefine-client* `-H` *ip* `-P 80 --list`
  - statt *ip* die IP-Adresse des Servers, z.B. 207.154.255.93
  - wenn keine Fehlermeldung erscheint, ist der Server erreichbar :)

Hilfe des Programms aufrufen

- *openrefine-client* `--help`

Die grafische Oberfläche von OpenRefine ist unter `http://` *ip* erreichbar.

1. Aufgabe DOAJ (14:50-15:15)

Wir bearbeiten einen Auszug (1001 Datensätze) aus dem Directory of Open Access Journals (DOAJ). Die manuellen Schritte aus dem Library Carpentry Tutorial (https://librarycarpentry.org/lc-open-refine/) wollen wir hier automatisiert erledigen. Wir verwenden dabei zwei der vorhin heruntergeladenen Dateien:

- Quelldaten: `doaj-article-sample.csv`
- Transformationsregeln: `doaj-openrefine.json`

a) OpenRefine-Projekt über die Kommandozeile anlegen

Verwenden Sie den *openrefine-client*, um die Quelldaten `doaj-article-sample-csv` als neues Projekt in OpenRefine zu laden.

Hinweise:

- Schlagen Sie in der Hilfe den benötigten Befehl nach: *openrefine-client* `--help`
- Bei jedem Befehl müssen die Verbindungsdaten mit angegeben werden: ... `-H` *ip* `-P 80`
- Die Daten sind in UTF-8 kodiert, das muss als Parameter angegeben werden: `--encoding=UTF-8`
- Bitte passen Sie den Projektnamen an, damit Sie sich mit den anderen Workshop-Teilnehmer*innen nicht in die Quere kommen.
    - Beispiel: `--projectName=doaj-`*vorname*

b) Daten transformieren

Schauen Sie sich zunächst die Daten in der grafischen Oberfläche von OpenRefine an:

- Öffnen Sie OpenRefine im Browser (Beispiel: http://207.154.255.93) und rufen Sie Ihr Projekt auf.
- Im Reiter `Undo/Redo` drücken Sie den Button `Apply` und fügen Sie den Inhalt der Datei `doaj-openrefine.json` ein.
- Machen Sie die Änderungen mit einem Klick auf Schritt `0. Create Project` rückgängig.

Versuchen Sie es nun mit dem *openrefine-client* über die Kommandozeile. Vgl. *openrefine-client* `--help`

Prüfen Sie anschließend das Ergebnis in der Oberfläche von OpenRefine.

c) Daten im Format CSV exportieren

Laden Sie mit dem *openrefine-client* die Daten im Format CSV vom Server auf Ihren PC. Schlagen Sie in der Hilfe den benötigten Befehl nach.

Prüfen Sie anschließend die Datei auf Ihrem PC in einer Tabellenverarbeitung (Libre Office, Excel).

2. Aufgabe Powerhouse (15:15-15:30)

Zur Übung bearbeiten wir nun noch einen etwas größeren Datensatz, einen Auszug von Metadaten vom Powerhouse Museum aus Australien. Die manuellen Schritte aus dem Tutorial von Programming Historian (https://programminghistorian.org/en/lessons/cleaning-data-with-openrefine) wollen wir hier automatisiert erledigen.

- Quelldaten: `powerhouse.tsv`
- Transformationsregeln: `powerhouse-openrefine.json`

Gehen Sie genauso vor wie in der Aufgabe zuvor und erledigen Sie die drei Schritte:

- a) OpenRefine-Projekt über die Kommandozeile anlegen
- b) Daten transformieren
- c) Daten im Format CSV exportieren

Hinweise:

- Die Ausgangsdaten haben ein etwas spezielles Format. Damit die Daten richtig interpretiert werden, muss die Import-Option `Use character " to enclose cells containing column separators` deaktiviert sein. Der entsprechende Parameter lautet: `--processQuotes=false`
- Denken Sie daran, wieder den Projektnamen anzupassen.
- Die Datei ist etwas größer. Bei langsamer Datenübertragung kann es mehrere Minuten dauern bis beim Import eine Rückmeldung erscheint.
- Der Export als CSV kann insbesondere unter Windows viel Rechenleistung beanspruchen. Nicht wundern, wenn der Lüfter anspringt und es mehrere Minuten dauert.

3. Aufgabe Templating (15:30-16:00)

Um die Möglichkeiten der Automatisierung zu demonstrieren, schauen wir uns noch eine spezielle Export-Funktion von OpenRefine an, den sogenannten "Templating-Export". Dabei können beliebige Vorlagen geschrieben werden, um weitere als die direkt unterstützten Dateiformate zu generieren. Es ist etwas umständlich, aber ermöglicht den Export aus OpenRefine in Formate wie JSON, XML oder MODS.

Schauen Sie sich zunächst die Funktion in der Oberfläche von OpenRefine an, indem Sie ihr Projekt aus Aufgabe 1 aufrufen. Klicken Sie oben rechts auf den Button `Export` und dann auf `Templating...`

- Links gibt es vier Eingabefelder: `Prefix`, `Row Template`, `Row Separator` und `Suffix`
- Rechts wird in einer Vorschauansicht das Ergebnis gezeigt.
- Voreingestellt ist ein Template für JSON, wobei Null-Werte mit ausgegeben werden.

a) Projekt DOAJ als JSON exportieren

Der *openrefine-client* verfügt über Funktionen für den Templating-Export. Schauen Sie sich die benötigten Befehle und Optionen in der Hilfe an: *openrefine-client* `--help`

Aufgabe: Versuchen Sie nur die drei Spalten DOI, Title und Authors als (valides) JSON auszugeben.

Hinweise:

- Wenn Sie nicht oft mit der Kommandozeile arbeiten, werden Sie Schwierigkeiten haben, mit den Zeilenumbrüchen im Template umzugehen. Ersetzen Sie diese durch Leerzeichen, so dass der ganze Befehl in einer Zeile steht.
- Die Anführungszeichen innerhalb des Templates müssen escaped werden (Windows) oder der ganze Parameter mit einfachen Hochkommata `'` umschlossen werden (Linux).
- Hier die benötigten Parameter als Starthilfe:

```
--export "projektname"
--template="{ \"DOI\" : {{jsonize(cells[\"DOI\"].value)}} }"
--prefix="{ \"rows\" : ["
--rowSeparator=","
--suffix="] }"
```

- Die Trennzeichen in der Autorenspalte können beim Templating aufgesplittet werden, indem die Transformationsregel `split("|")` ergänzt wird (muss mit einem `.` an value anschließen).
  - Achtung: Unter Windows muss das `|` Zeichen mit `^` escaped werden (also zu `^|`).
- Nutzen Sie ein Validierungswerkzeug, um das Ergebnis zu prüfen. Beispiel: https://jsonformatter.org

b) Nur Datensätze in spanischer Sprache als JSON exportieren

Testen Sie den Parameter `--filterQuery` in Verbindung mit `--filterColumn`, um nur diejenigen Datensätze auszugeben, die in spanischer Sprache geschrieben wurden

c) Alle Datensätze als einzelne JSON-Dateien exportieren

Testen Sie den Parameter `--splitToFiles` in Verbindung mit `--output`, um die Datensätze nicht als eine große JSON-Datei, sondern als einzelne JSON-Dateien (1001) zu speichern.

Lösungen

Aufgabe 1: DOAJ

- *openrefine-client* `-H` *ip* `-P 80 --create doaj-article-sample.csv --encoding=UTF-8 --projectName=doaj-` *vorname*
- *openrefine-client* `-H` *ip* `-P 80 --apply doaj-openrefine.json doaj-` *vorname*
- *openrefine-client* `-H` *ip* `-P 80 --export --output=doaj.csv doaj-` *vorname*

Aufgabe 2: Powerhouse

- *openrefine-client* `-H` *ip* `-P 80 --create powerhouse.tsv --processQuotes=false --projectName=powerhouse-` *vorname*
- *openrefine-client* `-H` *ip* `-P 80 --apply powerhouse-openrefine.json powerhouse-` *vorname*
- *openrefine-client* `-H` *ip* `-P 80 --export --output=powerhouse.csv powerhouse-` *vorname*

Aufgabe 3: Templating

- Windows Beispiel:

```
openrefine-client_0-3-4_windows.exe -H 207.154.255.93 -P 80 --export "doaj-vorname"
--template="{ \"DOI\" : {{jsonize(cells[\"DOI\"].value)}}, \"Title\" :
{{jsonize(cells[\"Title\"].value)}}, \"Authors\" :
{{jsonize(cells[\"Authors\"].value.split(\"^|\"))}} }" --prefix="{ \"rows\" : [" --
rowSeparator="," --suffix="] }" > doaj.json
```

- MacOS/Linux Beispiel:

```
./openrefine-client_0-3-4_linux-64bit -H 207.154.255.93 -P 80 --export "doaj-
vorname" --template='{ "DOI" : {{jsonize(cells["DOI"].value)}}, "Title" :
{{jsonize(cells["Title"].value)}}, "Authors" :
{{jsonize(cells["Authors"].value.split("|"))}} }' --prefix='{ "rows" : [' --
rowSeparator=',' --suffix='] }' > doaj.json
```

- Nur Datensätze in spanischer Sprache:
  - Zusatz `--filterColumn=Language --filterQuery=ES`
- Alle Datensätze als einzelne JSON-Dateien exportieren:
  - `--output=doaj.json --splitToFiles=true` als Ersatz für `> doaj.json`