The short report export script

Jonathan Hooper, Leeds. May 2022

Contents

Setup

Collect options, config etc

simpleExport.php circa lines 98 - 115

- Set threshold for including data in report (80% similarity)
- Get command-line options (initialise or append)
- Get config from config.ini - output CSV or TXT, whether to include byte-order-mark (BOM), whether to include a separate table of country counts in the short report
- Collect country name/code mappings
- Define column headings to include in report (i.e. the keys of $outputRecord to include)

Stage 1:

Loop over each citation in the input JSON data

simpleExport.php circa lines 161 - 789

In this loop:

- The input citation is **$citation**
- A variable **$outputRecord** is created to hold data that might go into the report
- At the end of each pass (at the end of Stage 1b) **$outputRecord** is appended to **$outputRecords** (the list of output records)

Stage 1a:

Assemble the data from the different external sources

simpleExport.php circa lines 248 - 713

- It tries to collect data on matched resources from Scopus, WoS and VIAF and saves this in locations like:

$outputRecord["SCOPUS-AUTHORS"]
$outputRecord["SCOPUS-COUNTRIES"]
$outputRecord["WOS-AUTHORS"]
$outputRecord["WOS-COUNTRIES"]
$outputRecord["VIAF-AUTHORS"]
$outputRecord["VIAF-COUNTRIES"]
etc

- For Scopus and WoS, all the harvested records are saved in **$outputRecord**, *regardless of the similarity score* - low-scoring results will be filtered out later
- For VIAF, harvested data is *only* saved for individual authors with a similarity >= 80
  There is a fundamental difference between Scopus+WoS and VIAF. From Scopus+WoS we retrieve a single resource record which may have multiple authors: either all the author-data is relevant, or none of it is, and so we can leave the filtering until later. From VIAF we may retrieve multiple separate author records, and the relevance of each is independent of each other. It's easier in practice to do the filtering now, as we look at each record, than later on.
- A similarity score is saved in **$outputRecord["SCOPUS-SIMILARITY"] etc** - this is the **maximum** similarity for **any** author in the citation
  *For Scopus and WoS, we may only have a single author in Leganto, but the source may contain multiple authors. It is reasonable to say that if **any** of the source authors is a good match to our author, we should treat **all** the source authors as reliable matches*
- We record whether ot not the successful search against the external source was carried out by **DOI**, in the Boolean **$outputRecord["SCOPUS-SEARCH-DOI"]** etc
  A successful search by DOI is almost certain to be a correct match, regardless of similarity score, so it is useful to know this
- Country data is harvested from suitable fields in the source data and saved to two lists-of-lists:
  - **$outputRecord["SCOPUS-COUNTRY-CODES"]** etc
    *The primary data, using **2-letter ISO codes** (e.g. "GB")*
  - **$outputRecord["SCOPUS-COUNTRIES"]** etc
    Calculated from the primary data using array_map() to apply the code->name mappng to all the country codes
- Different sources, and different fields in different sources use different forms for countries, and some sources are inconsistent, so the code attempts to translate from 3-letter ISO codes or names to 2-letter codes, using map and alias files in **Config/CountryCodes** (see above)
- Country data that cannot be converted to a 2-letter ISO code is ignored (this includes foreigh-language country names, unrecongised names. and invalid 2- and 3-letter codes)
- We can provide aliases for names and 2-letter codes (see above)
- Both $outputRecord["SCOPUS-COUNTRY-CODES"] and $outputRecord["SCOPUS-COUNTRIES"] are broken down by author
  - e.g. $outputRecord["SCOPUS-COUNTRY-CODES"]**[0]** contains a list of country codes associated with the **first** author from Scopus, etc

- Fields used from Scopus are (first choice) "**contemporary affiliation**" (affiliation at the time of publication) or (if that is not present) "**current affiliation**"
- The field used from WoS is the **address** associated with this author
    - In WoS, some or all of the addresses may not be associated with any individual author (especially pre-2008). Addresses not associated with any individual author are grouped togather as belonging to a final anonymous "author"
- The field used from VIAF is **nationalities**
- Duplicates (for a given author) are removed using **array_unique()** i.e. if an author has three addresses in WoS (GB, GB and DE), we will only record GB, DE
- The short report collects a simpler set of author-title data than the long report does (see below) e.g. it selects a single citation title rather than including all the different forms that appear in Leganto/Alma
    - This has the advantage of readability, but the disadvantage that it may not be obvious why a high similarity score has been given (we might have a good match between titles in the reading list and the external source other than those that appear in the report)

Stage 1b:
Select which external source data to use for the report

simpleExport.php circa lines 717 - 780

*The short report only includes data from **one** of the external sources (Scopus/WoS/VIAF) for each citation so we need to identify the best one to use.*

- **For article-like citations** (Leganto material type is CR/E_CR/JR) order of preference of sources is **Scopus > WoS > VIAF**
- **For other citations**, order of preference of sources is **VIAF > Scopus > WoS**
- *These orders of preference were set by the Library, after inspecting some sample data*
- For a citation, using this order of preference, we find the first source (if any) which has *either*:
    - $outputRecord["*SOURCE*-SIMILARITY"] >= 80
      this threshold is set in the variable $inclusionThreshold
      or;
    - $outputRecord["*SOURCE*-SEARCH-DOI"] == TRUE
- We save the label of the selected source (SCOPUS/WOS/VIAF) into **$outputRecord["SOURCE"]**
- We copy the source-specific harvested data into a definitive per-citation location e.g. if we select Scopus, we copy **$outputRecord["SCOPUS-COUNTRY-CODES"]** into **$outputRecord["COUNTRY-CODES"]**

- We map the Boolean **$outputRecord["*SOURCE*-SEARCH-DOI"]** to a String **$outputRecord["DOI-MATCH"]** (Y|N)

Stage 2:
Collect country counts per reading list

simpleExport.php circa lines 803 - 852

This code collects country counts in an associative array **$countryCodeCounts**. Initially this array is empty:

- The keys of this array are report filenames - i.e., in practice, reading list codes (e.g. "0.46351519280917763").
- The values are associative arrays mapping country names to counts (e.g. ["Germany":5, "France":2]).

*NB for historical reasons some of the variables in this section have names like $country**Code**Counts, because initially we were saving country data by code. We are now using country names, so these variables are misnamed, although it makes no significant difference (and we could easily revert to using codes instead).*

The code loops over **$outputRecords**, fetching each **$outputRecord** by reference rather than value (because it will be modified):

- **$outputRecord["SOURCE-COUNTRIES"]** is an array we collected in Stage 1a - each value is a list of countries (one list for each individual author)
- The code loops over these individual authors:
    - Counting each country
    - Normalising these counts so that they total 1 for each individual author
    - Adding the normalised counts to a running total for the citation
- The code then adds the citation running totals for countries to a list-level running total
- This is saved in **$countryCodeCounts[{FILENAME}][{COUNTRY}]**

Stage 3:
Write the output files

simpleExport.php circa lines 855 - 1031

Stage 3a:
Open Summary report file

simpleExport.php circa lines 863 - 873

The Summary report is opened for appending.

*If script is called with **initialise** option, or **without append** option, it is emptied, and the header row is written*
*If the script is called with the **append** option, we will simply add rows to the existing summary report*

Stage 3b:
Write the short report file

simpleExport.php circa lines 878 - 1024

*This is not done if the script has been called with the **initialise** option, since initialise does not output any actual readling list data*

The code loops over the output records in $outputRecords and:

- If this record requires a different report filename from the last one seen (i.e. if we've hit a new reading list):
    - Closes any existing short report file handle
    - Outputs the last list's summary row to the Summary report
    - Opens a new short report file and outputs the header row (with BOM if using)
    - Sets various counters to zero for the new list's row in the Summary report
- Increments various counters for the new list's row in the Summary report
- Assembles data for each of the column headings required, from $outputRecord and **$outputRow**
    - In this process, scalars (e.g. 100 or "Y") are copied as-is
    - Missing values are replaced with the empty string
    - Arrays are flattened - at the top-level with pipe character, and then with semi-colon e.g.:
      ["A", "B", "C"] becomes **"A|B|C"** and
      [["GB", "IE"], ["DE"], ["AU", "DK"]] becomes **"GB;IE|DE|AU;DK"**
- Outputs the row **$outputRow** to the short report file

Once all output records have been processed:

- If we're exporting the additional table of country counts below the main table, do that (using the $countryCodes that we collected in Stage 2)
- Append any remaining summary row from the very last reading list to the summary report
- Close any open file handles