← Return to SPOKENWEBLOG

# The Bad Batch: Open Refine as a Batch Editing Method in SWALLOW

Ben Joseph, Tomasz Neugebauer and Cole Mash June 8, 2022

Article, Collaborations, DH Design and Tech, SPOKENWEBLOG | audio, batch editing, ben joseph, Cole Mash, data, design, DH, digital humanities, Metadata, openrefine, Sound, SpokenWeb, Swallow, Tech, Tomasz Neugebauer



,Description/identifiers#1,Institution_and_Collection/persistent_URL,Material_I
designation,Material_Description/physical_composition,Material_Description/
,File_Description/duration,Digital_File_Description/size,Digital_File_Descriptic
ote,Item_Description/title_source,Item_Description/language,Item_Description,
Rights/rights,Rights/notes,Notes#1/note,Notes#1/type,Content/contents,Co
le#1,Creator#1_role#2,Creator#2,Creator#2_role#1,Creator#2_role#2,Cre
#2,Creator#5,Creator#5_role#1,Creator#5_role#2,Creator#6,Creator#6_ro
or#8_role#1,Creator#8_role#2,Creator#9,Creator#9_role#1,Creator#9_role
reator#11_role#2,Creator#12,Creator#12_role#1,Creator#12_role#2,Creato
role#2,Creator#15,Creator#15_role#1,Creator#15_role#2,Creator#16,Creat
ator#18,Creator#18_role#1,Creator#18_role#2,Creator#19,Creator#19_rol
,Description/identifiers#1,Institution_and_Collection/persistent_URL,Material_I
designation,Material_Description/physical_composition,Material_Description/
,File_Description/duration,Digital_File_Description/size,Digital_File_Descriptic
ote,Item_Description/title_source,Item_Description/language,Item_Description,
Rights/rights,Rights/notes,Notes#1/note,Notes#1/type,Content/contents,Co
le#1,Creator#1_role#2,Creator#2,Creator#2_role#1,Creator#2_role#2,Cre
#2,Creator#5,Creator#5_role#1,Creator#5_role#2,Creator#6,Creator#6_ro
or#8_role#1,Creator#8_role#2,Creator#9,Creator#9_role#1,Creator#9_role
reator#11_role#2,Creator#12,Creator#12_role#1,Creator#12_role#2,Creato
role#2,Creator#15,Creator#15_role#1,Creator#15_role#2,Creator#16,Creat
ator#18,Creator#18_role#1,Creator#18_role#2,Creator#19,Creator#19_rol

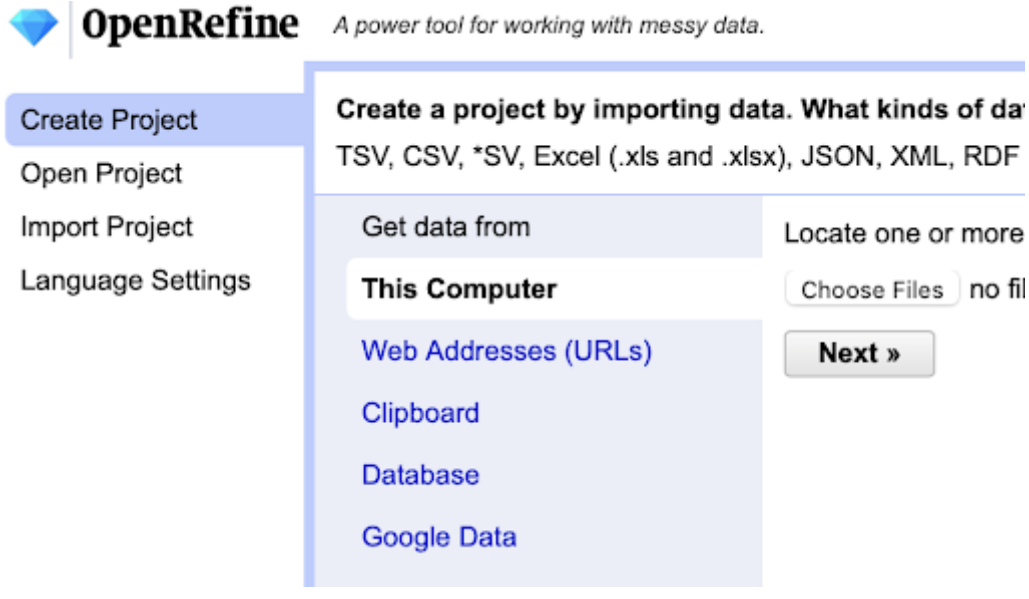


Figure 1. Open Refine GUI.

## Explaining the Case and Software Tool

In 2019, SpokenWeb SFU Project Manager Cole Mash (SFU) and SpokenWeb Systems Task Force member Tomasz Neugebauer (Concordia) began work on editing SWALLOW entries. SWALLOW is an open-source metadata ingestion system developed by the SpokenWeb team to describe and manage the project's object of study: literary audio. Since the implementation of SWALLOW in 2018, SpokenWeb team members have ingested over 4,700 entries into the system.

With so many literary artifacts, the team needed a way to edit entries in larger swaths, rather than having to make all edits to the system individually, by hand. SWALLOW had been built with a way to export multiple entries out of the system, edit them, and import them back in. But how exactly would that process work?

Exploration, cleaning, transformation and reconciliation of data is an increasingly common task, with many scripting environments available to help with the work of rigorously implemented descriptions of audio artifacts and events. To explore the possibility of making changes to multiple entries at once, Neugebauer pointed Mash in the direction of OpenRefine because it offers robust functionality via a graphical user interface (GUI, or the visual field that an online user interacts with). OpenRefine's GUI has a sparse but user-friendly front end that opens on one's internet browser, providing users with a number of options to create and manage projects, as well as to import their data, including from the computer, a URL, a database, or to paste data into the provided clipboard. OpenRefine is an open source tool for cleaning up and homogenizing large amounts of raw data. Its website boasts that "OpenRefine (previously Google Refine) is a powerful tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data" ("Welcome"). Basically, the tool allows users to parse and transform large amounts of data into alternative usable forms. OpenRefine has a number of parsing and transformation options such as "TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. [And] [s]upport for other formats can be added with OpenRefine extensions." ("OpenRefine"). In

this way, OpenRefine is a really useful tool for more than just the kind of metadata work that Mash was seeking to perform.

Mash began to experiment with the program as a method for batch editing. SWALLOW, SpokenWeb's metadata ingest system, exports data as a JSON file, that is, JavaScript Object Notation, a "lightweight data-interchange format" that is both easily human readable and machine readable. JSON.org describes JSON as "a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages" ("Introducing JSON"). So first, Mash exported SWALLOW JSON for SFU's Radio Free Rainforest Collection out of SWALLOW and into OpenRefine by selecting export, refining his selection, and then copying and pasting the JSON code. After that, Mash did a lot of fiddling and experimenting. With no previous experience with the program, it was difficult to know exactly what options to use. How did we want to use this new tool and its myriad options for parsing data? Did we want to parse the data as a JSON file or as a CSV file? Did we want to preserve empty strings or trim the white space? A number of experiments followed in which Mash simply tried a number of different options. The goal was to create an output of metadata in the form of a CSV file, with the necessary metadata fields organized clearly into columns and rows. After experimenting, Mash chose what he thought were the correct options and met with Neugebauer, who 'refined' (see what we did there?) Mash's process a little further. Ultimately, the JSON from SWALLOW was parsed as JSON into organized columns and rows of cells. This worked because the exported JSON was consistent in its structure since we worked with items encoded in one SpokenWeb schema version at a time. The rows and cells were then formatted, rearranged, and edited down to produce a clear spreadsheet containing the data from SWALLOW in a curated and organized manner (see **Figure 2**). JSON format makes it possible to have more than tabular data. It is a richer format than CSV, allowing for hierarchical data structures. CSV is just a table of rows and columns of data. For that reason, depending on what the data is, of course, a conversion from JSON into CSV requires some re-interpretation of the data into columns. Fields that allow for multiple entries, such as the creator and contributor fields in SpokenWeb Schema, result in "multi-line" CSV, where each "record/item" consists of many rows of data in the CSV file. OpenRefine does make it possible to "transpose" rows into additional columns, if a one row per record CSV file is required.

| _ - _ - schema | _ - _ - swallow_id | _ - _ - schema_version | _ - _ - partner institution | _ - _ - Creators - _ - id | _ - _ - Creators - _ - url |
|---|---|---|---|---|---|
| Swallow JSON | 1514 | 3 | Simon Fraser University | 5ddc8d5ab40ad4.61291086 | |
| | | | | 5ddc90ed145ab1.18582239 | |
| | | | | 5dee8d9f980f47.35662404 | |
| | | | | 5ddc8f758ce900.19296529 | https://viaf.org/viaf/46807530 |
| | | | | 5ddc8fe0d1f4e0.37163133 | http://viaf.org/viaf/75765123 |
| | | | | 5ddc9022b1bda5.87998035 | http://viaf.org/viaf/56801571 |
| | | | | 5ddc90b225f910.25465066 | https://viaf.org/viaf/90734890 |
| | | | | 5ddc91246e55f3.33718287 | http://viaf.org/viaf/34469976 |
| | | | | 5dee8aa086ee81.04629532 | https://viaf.org/viaf/75102538 |
| | | | | 612696c9398659.31810605 | |
| | | | | 5dee8b857899d0.59114625 | http://viaf.org/viaf/56616559 |
| | | | | 5ddc916bc55586.58260170 | |
| | | | | | |
| | | | | 5ddc8daa1d4d76.09168486 | http://viaf.org/viaf/13554305 |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Swallow JSON | 1539 | 3 | Simon Fraser University | 5de58455a67147.35383903 | http://viaf.org/viaf/38243 |
| | | | | 5de5884d6132b1.12943864 | http://viaf.org/viaf/75225856 |
| | | | | 5de583b5516ef3.53974466 | http://viaf.org/viaf/46615553 |
| | | | | 5de5856d3462a8.51762063 | http://viaf.org/viaf/105947148 |
| | | | | 5de8ae1363e992.61199843 | http://viaf.org/viaf/13554305 |
| | | | | | |

**Figure 2.** CSV file of Radio Free Rainforest JSON parsed through OpenRefine.

Once a method had been developed for using OpenRefine as an editing tool, we needed to test it. Enter SFU RA Ben Joseph. Joseph is a second year computer science student at SFU, who works on the SpokenWeb project. He is SFU's go-to RA for working with more complex digital tools. At the time, Ben was working on editing the metadata for Gerry Gilbert's radiofreerainforest (rfr) collection. This collection was almost complete, and ready for a final edit. Consisting of audio files of a radio show, the collection's metadata was well documented and more straightforward than some of the maverick modes of recording and unmarked tapes that we find in other collections. This made it the perfect collection to test our new method for editing using OpenRefine.

Joseph and Mash began editing using OpenRefine, though ultimately they would not do a batch edit. You see, batch editing requires exporting a large amount of files from SWALLOW, changing data, and then importing it back in. This process is tricky and can lead to errors in the files that are imported back into SWALLOW if not done correctly. For Neugebauer, it was important that if

an export/import was to be done, a significant amount of data should be changed, or it would not be worth engaging in this complex and risky process. Neugebauer asked the question, does this collection need to be batch edited? Or can the errors simply be edited by hand (that is, edited in SWALLOW itself in each individual entry)? The answer was emphatically the latter—all of the errors that were found could be done by hand and a batch edit was likely overkill.

But all was not lost in this experiment. It illuminated that our question was not simply how to do a batch edit, but a question that needed to be asked of each collection we edit: is a batch edit required at all?

It just so happens that OpenRefine can also be useful for determining whether or not a batch edit is warranted. By exporting the data out of SWALLOW and into OpenRefine, and then a spreadsheet, the data was still more clearly organized and easier to read from a distance than it would be in SWALLOW. SWALLOW's ingestion system requires you to enter into each individual entry in order to edit the metadata, which can be cumbersome (see **Figure 3**). However, when viewing data in a spreadsheet, one can view and sort large amounts of entries and compare data across them (see **Figure 4**). This makes it easy to see what kind of edits need to be done much quicker than if you were just using SWALLOW. Once one decides whether or not a batch edit is necessary based on the amount of errors that appear, then all one needs to do is go into SWALLOW, locate and make each edit by hand, *or* make the edits in the spreadsheet, and then begin the import process.

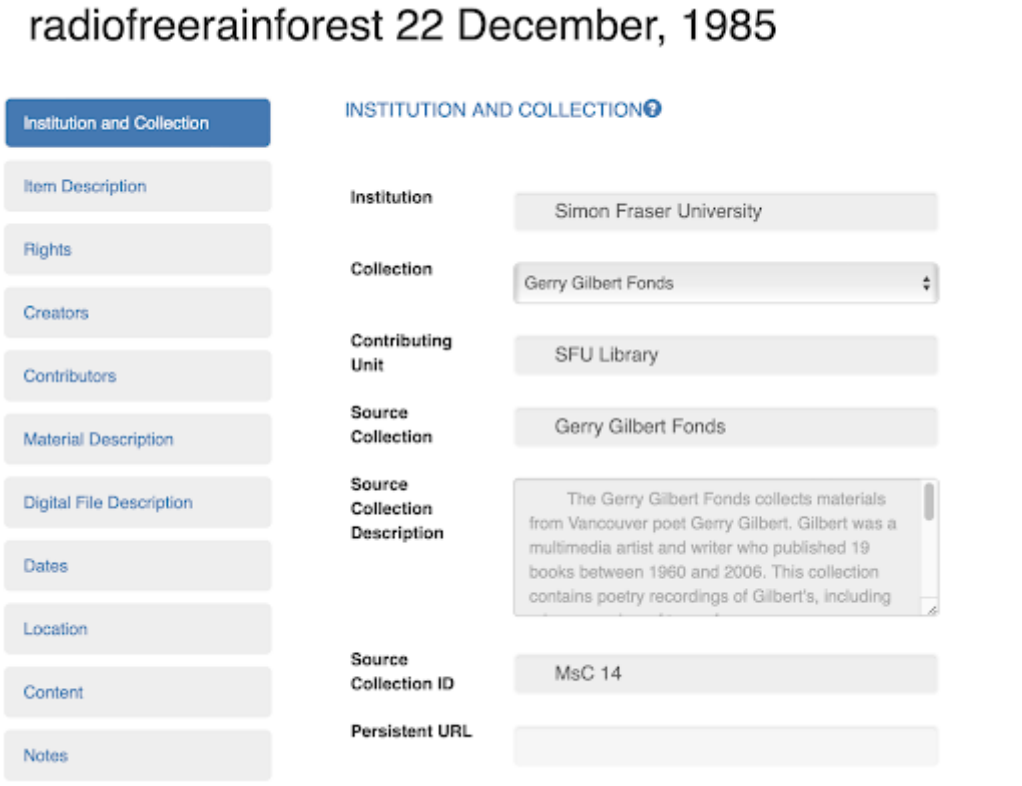


**Figure 3.** Metadata for six fields from artifact #1514 from the Radio Free Rainforest collection in SWALLOW.

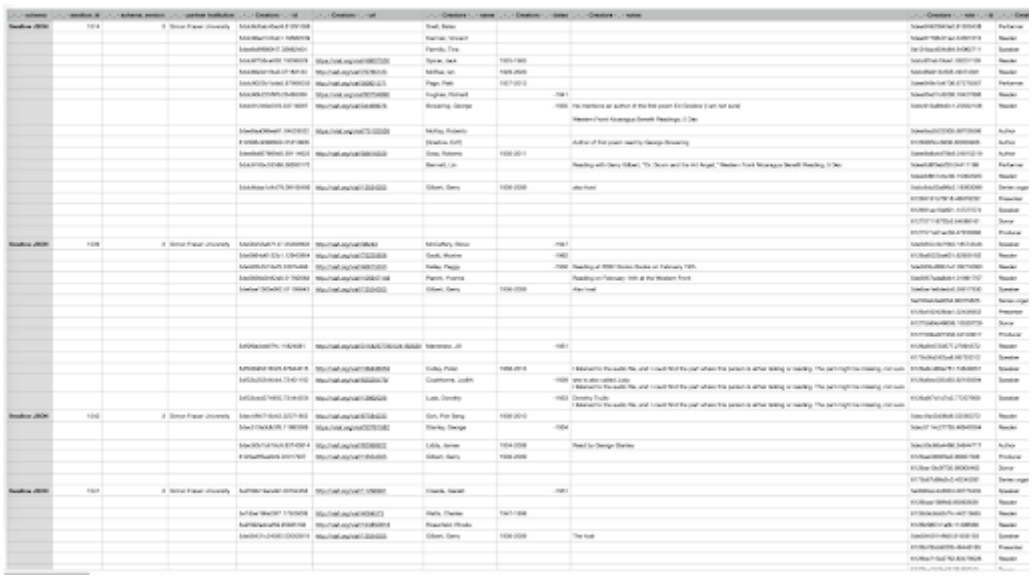

**Figure 4.** Metadata for eleven fields from four different artifacts from the Radio Free Rainforest collection as a CSV spreadsheet.

So, OpenRefine can be used to batch edit large amounts of metadata in a number of useful ways. However, it's not just an export/import editor. It also functions as a method for "distant reading" the metadata of a collection, allowing users to see more easily the errors in a given collection so that edits might be made by hand in SWALLOW.

Joseph, Mash, and Neugebauer have drafted workflows for several tasks relevant to the verification and correction of metadata in SWALLOW, including a process for how to check the metadata to determine whether there are enough errors to warrant performing a full batch edit, or not, as well as instructions regarding how to do a batch edit should that prove necessary. These

workflows and additional comments on the processes are provided in the sections that follow.

# Workflow, Other Things, and Bots

### 1. Workflow

### 1.1 Export from SWALLOW

**1.1.1** Decide which collection or group of artifacts for which you will edit the metadata.

**1.1.2** Sign into SWALLOW.

**1.1.3** Click the **Export** button on the left sidebar.

**1.1.4** Choose the **Institution**, **Cataloguer**, and **Collection** you will export. Under **Schema**, select one schema only, usually the most recent version number. If you need to export records with multiple schema versions, you will have to import them in batches by schema version, so it is good to keep the export files separate by schema version.

**1.1.5** Beside **Export current dataset as** select **Swallow JSON**. Click **Export** button.

**1.1.6** Wait for the result to load, then click on "File > Save Page As" in your browser to save a copy of the JSON export as a file. Alternatively, **copy** [CTRL/CMND C] the JSON text provided onto the clipboard. Make sure you select all of this text!

### 1.2 Parse text in OpenRefine

**1.2.1** Open the OpenRefine application on your computer. It will open as an app, and then in your browser.

**1.2.2** Click on **Choose Files** and select the file you saved in step 1.6. Alternatively, select **Clipboard** and **paste**[CTRL/CMND P] the JSON text you copied to the clipboard.

**1.2.3** Click **Next.** OpenRefine will transform data string from SWALLOW into generic columns and rows.

**1.2.4** You now have a number of options for parsing and transforming the text. Under **Parse data as**, select **JSON files**.

**1.2.5** Now, you will have a few options pertaining to parsing the data as JSON.

**1.2.6** Select the button **Parse cells into numbers, dates...**

**1.2.7** Select **Create Project**.

**1.2.8** Click the drop down arrow on the left under **All** and select **Reorder/Remove columns**.

**1.2.9** Reorder columns into a way that will be easiest for you to engage with visually.

**1.2.10** Delete any columns that you won't be focusing on in your editing.

**1.2.11** In the top right, click the **Export** drop down menu, and select either **Excel (.xls)** or **Comma-separated Value**, depending on if you use excel specifically or another spreadsheet program.

**1.2.12** The CSV will download on your computer. Open it with excel or another spreadsheet program.

### 1.3 Spreadsheet-based Editing

**1.3.1** Now that you have a spreadsheet, your next job will be to decide whether you will need to edit this spreadsheet and then import the data back into SWALLOW (batch edit) or simply use OpenRefine's resulting spreadsheet to quickly locate errors and then edit them by hand, so to speak, in SWALLOW itself. But how do you know which method is the best one to use? With the spreadsheet, it will be much easier to view large amounts of data simultaneously than it is in SWALLOW. Use this functionality to locate how many errors there are in the collection and what the nature of those errors are.

**1.3.2** Begin by going through the cells of each field, focusing on areas you know may have errors (VIAF, Name Spelling, Dates and Date formatting, Open Street Maps Link etc.). You will not be able to check the data in every single cell—this would take even longer than it took to enter the data in the first place. As with everything with SWALLOW, you need to decide how much time you have to spend on this task, and use that time wisely. Ask questions to decide what data to check. For instance, which cells commonly have errors? Who was the original cataloguer and what cells do they struggle with or commonly succeed with? You can also click all the links provided to check their validity, and use the search function to check on data that is repeated. Check as much data as you can, making note of errors along the way.

You may need to get creative to discover errors in the spreadsheet. If you know a name or date or place that comes up quite frequently, you might want to search to ensure it has been done correctly. For example, you could search the name Daphne Marlatt, then look through all the entries with her name to ensure the right VIAF has been entered.

### 1.4 Correcting inconsistent details of creators or contributors (Name, date, VIAF URL)

**1.4.1** This is also an important step as some creators have aliases or different spellings of their name within the same collection, and there is dissenting information on the web that might lead to incorrect dates, etc.

- The simplest solution to this would be to temporarily sort your rows by one column: the creators/ contributors list alphabetically, and look through the instances of the names, making sure they're all identical.
- This could be done within OpenRefine, making use of the "Sort" dropdown menu in the toolbar, followed by the Ctrl+H shortcut to search for specific instances of a name.

### 1.5 Deciding how to proceed with Batch Edit

**1.5.1** Make a list of the errors you find, including which catalogue entry they are in. Ask yourself how long it might take to correct these errors by hand?

**1.5.2** Decide if it will be easy to edit these by hand in SWALLOW, or if a larger batch import is needed.

**1.5.3** If the former, then edit these by hand.

**1.5.4** If the latter, edit these in the spreadsheet itself (either individually, or through search, and replace if there are many entries).

**1.5.5** A larger batch import would usually be easier for repeated instances of the same error within a collection.For instance, if the Brands of every single audio tape in every entry within a collection needed to be changed.

**1.5.6** If you're going through with a batch edit, before importing the changed file, Use Ctrl+H to Open the find and replace option within the CSV/json/excel file you will be using. Replace every instance of the collection name with "CollectionName-Duplicate".

### 1.6 Import Back into SWALLOW

**1.6.1** Duplicate the collection that you are working on, so that you have a backup inside of Swallow in case something goes wrong. Go to **Collections** and click on Duplicate button to the right of the collection name you want to import back into. This should create a collection "-COPY".

**1.6.2** If you have made changes inside of OpenRefine, you need to export the updated content from OpenRefine as a CSV file and then Import the collection back into SWALLOW. To export the updated content from OpenRefine, click on Export and select CSV. You can import this CSV file by going to **Import** and selecting "Swallow CSV to V3" under Mapping Function, or another custom CSV mapping function. The mapping functions rely on custom matching column names in the CSV file, so you will have to ensure that the column headings in your CSV files correspond to these. For example, the following is a list of column headings for a University of Alberta import, listed in CSV format. Select "UofA CSV to V3" under Mapping Functions when importing a CSV file that uses these headings:

cataloguer_last_name_first_name,partner_institution,contributing_unit,Collection.source_collection,Institution_and_Collection/item ID,Item_Description/identifiers#1,Institution_and_Collection/persistent_URL,Material_Description/recording_type,Material_Description/AV_type,Material_Description/material_designation,Material_Description/physical_composition,Material_Description/tape_brand,Material_Description/playing_speed,Digital_File_Description/filename

,Digital_File_Description/duration,Digital_File_Description/size,Digital_File_Description/bitrate,Digital_File_Description/encoding,Item_Description/title,Item_Description/title_note,Item_Description/title_source,Item_Description/language,Item_Description/genre#1,Item_Description/genre#2,Dates/date,Dates/type,Dates/notes,Location/venue,Rights/rights,Rights/notes,Notes#1/note,Notes#1/type,Content/contents,Contributor#1,Contributor#1_role,Contributor#2,Contributor#2_role,Creator#1,Creator#1_role#1,Creator#1_role#2,Creator#2,Creator#2_role#1,Creator#2_role#2,Creator#3,Creator#3_role#1,Creator#3_role#2,Creator#4,Creator#4_role#1,Creator#4_role#2,Creator#5,Creator#5_role#1,Creator#5_role#2,Creator#6,Creator#6_role#1,Creator#6_role#2,Creator#7,Creator#7_role#1,Creator#7_role#2,Creator#8,Creator#8_role#1,Creator#8_role#2,Creator#9,Creator#9_role#1,Creator#9_role#2,Creator#10,Creator#10_role#1,Creator#10_role#2,Creator#11,Creator#11_role#1,Creator#11_role#2,Creator#12,Creator#12_role#1,Creator#12_role#2,Creator#13,Creator#13_role#1,Creator#13_role#2,Creator#14,Creator#14_role#1,Creator#14_role#2,Creator#15,Creator#15_role#1,Creator#15_role#2,Creator#16,Creator#16_role#1,Creator#16_role#2,Creator#17,Creator#17_role#1,Creator#17_role#2,Creator#18,Creator#18_role#1,Creator#18_role#2,Creator#19,Creator#19_role#1,Creator#19_role#2,Creator#20,Creator#20_role#1,Creator#20_role#2,Creator#21,Creator#21_role#1,Creator#21_role#2

**1.6.3** If you have made changes directly to the JSON file using VisualStudio Code, you can import the updated JSON file by going to **Import** and selecting "Swallow JSON V3 to V3" under Mapping functions. Select the "Preview" checkbox first, and correct any errors displayed. If there are no errors and the only warnings are for duplicate titles, you can batch delete the items inside the collection that you are about to replace (**Items** > use filter for the collection name and click on **Delete** button next to "Batch delete all these items"), to remove the duplicate titles warning. Now run the import with the "Preview" unchecked, this will actually import the updated items into the correct collection.

**1.6.4** Check to make sure it worked and the entries look good. If it is OK, you can delete the backup collection you created in 4.1 by going to Collections and clicking on **Delete** button to the right of the collection name.

## 2. Other Things to Keep in Mind

- It is inevitable that each researcher is going to enter the same data slightly differently. In your batch editing process, ask yourself, "Is the information in this cell wrong, or is it just different than how I would do it?" For example, if someone enters a date wrong, that is just wrong information. But if someone enters that they got the title from the "Tape" vs. entering that they got the title from the "J-Card," (which refers to the paper card inserted in the plastic storage case of an audio cassette) this isn't necessarily wrong, but rather an editorial choice and preference, and does not require editing. The editing process is not static, and requires critical thinking about how the entries can be made to be the most useful to future researchers, with the time we have been given to work on this project.
- Whenever possible, try to edit the entries in SWALLOW rather than importing the spreadsheet back in. This process can be laborious and has a higher risk for error than editing by hand.
- It might help to familiarize yourself with the capabilities of OpenRefine. A good place to start would be the documentation for the application: https://docs.openrefine.org.
- In order to import the spreadsheet back into SWALLOW, you're going to have to request a special Editor status for your SWALLOW account. If you're going through with a batch import, make sure that you make a duplicate collection before you begin. You will be importing your new information into the duplicate collection that you have created.

## 3. Using the Bot (Optional)

Ben has developed a bot to assist in batch editing tasks.

**3.1** Using the EditorBot Script (https://github.com/puppyhearts/EditorBot) to find missing details within entries:

**3.1.2** This application should be used to do a preliminary scan of the collection to point you in the direction of entries that may be flawed. It generates a list of typical errors that you may find in the collection and a list of important information that might be missing in an entry.

**3.1.3** Make sure you have python 3 installed on your computer: https://www.python.org/downloads/.

**3.1.4** Download the python scripts for the EditorBot from the github repository.

**3.1.5** In Finder.py, find the line of python code that says
df = panda.read_excel (r'GG1.xlsx')

**3.1.6** Change 'GG1.xlsx' to the name and file extension of the file you downloaded in step 2.13.

**3.1.7** Execute Finder.py

**3.1.8** The program should give you a list of Swallow IDs in ascending order, along with a list of information that that specific entry may be missing.This information may simply be unavailable rather than missing, but it's probably a good idea to look at every field that the program flags.

**3.1.9** The error message should look something like “XXX info not Mentioned”

**3.1.10** The list of missing information that the program currently checks for includes:

1. Rights
2. Item Description
    1. Genre
    2. Source of Title
    3. Production Context
    4. Item ID
3. Dates
    1. of Recording
    2. Type of Date
    3. Source of Date
4. Creator Details
    1. Creator Name
    2. Creator Dates
    3. Creator URL
    4. Creator Role
5. Contributor Details
    1. Contributor Name
    2. Contributor Dates
    3. Contributor URL
    4. Contributor Role
6. Physical Description
    1. Image of Tape
    2. Physical Condition
    3. Recording Type
    4. Sound Quality
    5. Extent
    6. Physical Composition
    7. AV Type
7. Digital File Description
    1. Size of Digital File
    2. Duration of Audio
    3. File Type
    4. File Name
8. Location

---


**Ben Joseph, Tomasz Neugebauer and Cole Mash**

**Rohan Ben Joseph** is an undergraduate student at Simon Fraser University, majoring in Computing Science and Linguistics. He is interested in Natural Language Processing and Data Analysis. He has experience as a research assistant for the SFU iXLabs working on Human-computer interaction under Dr. Parmit Chilana. His research on Software Daemons and Background Processes was published in the SFU Scientific Undergraduate Research Journal. He is currently working on Metadata processing and data visualisation in the SpokenWeb Project. He enjoys spending his spare time performing spoken word poetry, playing Minecraft and taking care of Artemis, his pet cocker spaniel.

**Cole Mash** is a PhD student Simon Fraser University in the English Department. His poetry has been published in *Forget Magazine*, *The Eunoia Review*, *Papershell*, and *OK Magpie* and his critical work has been published in *Scholarly and Research Communication*. He has worked on a number of major digital research projects as both RA and investigator. He is co-executive director of non-profit organization The Inspired Word Café and Managing Director of Kelowna Poetry Slam. His work centres on Performance Poetry in the traditions of Spoken Word and Slam, and their exclusion from the critical canon. Through digital methods, he hopes to bring that work into the canon whiles simultaneously studying how performance poets notate the paralinguistics of performance onto the page. Currently he is working in the Contemporary Literature Collection (CLC) at SFU's Special Collections, and cataloguing the current collection of sound.

**Tomasz Neugebauer** is the Digital Projects & Systems Development Librarian at Concordia University, where he participates in the design, development, and implementation of various research and library applications. His current research interests include information visualization, linked open data, metadata interoperability, open-source software systems used for digital curation, preservation, and the building of digital repository infrastructure. Tomasz has developed software for the visualization of bibliographic metadata, DNA data, and a number of software plugins to the EPrints digital repository platform. In 2013, he helped to launch the e-Artexte open access digital repository. He has been collaborating with the SpokenWeb research project since 2016.



SpokenWeb is a SSHRC-funded partnership grant.