2019-08-07 Brown Dog Meeting notes

Date

07 Aug 2019

Attendees

- Sandeep Puthanveetil Satheesan
- Luigi Marini - absent
- Shannon Bradley -
- Mark Fredricksen -
- Rob Kooper - absent
- Kenton McHenry - absent
- Gregory Jansen - absent
- Dukyun Nam - absent
- Deren Kudeki - absent

Discussion items

Time	Item	Who	Notes
		Shannon	Begin testing the Tools Transformation Catalog - what is this? 🙂 What is the link I need to share? https://browndog.ncsa.illinois.edu/transformations-dev/ Will need to fix CSS (size of icons issue) + path issue before we start - Yan can't test locally - possibly can run behind NGINX I will come up with a testing matrix and the follow up mechanism to gather input on bugs and improvement suggestions Make sure be ready for last report for NSF
		Sandeep	Restarted work on CSV aggregator extractor - focusing on unit conversions - current code finds all combinations of the columns and for each value combination it applies the other columns units on that column - ready to started with PINED Started with updating eScience poster paper on XSEDE project by incorporating reviewers comments. camera ready instructions are not out yet Started looking at Mark's PR
		Mark	Transformations-catalog with adjustable URLs has a pull request, but need reviewers. Yan reviewed, and had some issues, but I believe they were documentation issues for how to set up the URL_PREFIX. Added documentation. I also finished a test using nginx as a proxy_pass to a different server and verified that the prefix worked as intended.
		Deren	We created a repo for it - please just push code to Master https://opensource.ncsa.illinois.edu/bitbucket/projects/CATS/repos/extractors-hathitrust/browse technically can clone but not push to it - could be a group permission issue? Read only access? added to cats-dev per Sandeep's request
		Boris	asked questions in an email- Rob answered - see below
		Greg	Cluster has been migrated to new datacenter and networking (finally) restored to normal. I am starting work on aggregation of punchcard extractor output in Clowder. Also working to have it ignore non-punchcard images more often.

Boris's email
From: Kooper, Rob <kooper@illinois.edu>
Sent: Tuesday, August 6, 2019 2:37 PM
To: Capitanu, Boris <capitanu@illinois.edu>; Puthanveetil Satheesan, Sandeep <sandeeps@illinois.edu>; Burnette, Maxwell Amon <mburnet2@illinois.edu>; Bradley, Shannon <sbrad77@illinois.edu>
Cc: McHenry, Kenton Guadron <mchenry@illinois.edu>; Marini, Luigi <lmarini@illinois.edu>; Kudeki, Deren Emre <dkudeki@illinois.edu>
Subject: Re: Progress on Brown Dog Extractor

In general it is best to post these type of technical questions either in slack (maybe spaced over time) or send an email to the clowder mailing list. This will reach the broadest audience that can help with answering the questions.

The first problem about the period in the key name of json is due to the fact that the period is used in mongo to indicate a subdocument.

The second requires the document to be specific written to get the status to propagate correctly. This is all hidden in pyclowder, in your case you will need to send the appropriate message back using rabbitmq, you can see the message that is send in https://opensource.ncsa.illinois.edu/bitbucket/projects/CATS/repos/pyclowder2/browse/pyclowder/connectors.py#910 on the receiving end in clowder (https://opensource.ncsa.illinois.edu/bitbucket/projects/CATS/repos/clowder/browse/app/services/RabbitmqPlugin.scala#1178) you can see that it will look for the message DONE to mark the event as processed.

As for the timestamps they are based on the timestamps you provide in your message to clowder. If you don't provide a timezone we will guess it to be the same as the server, however in case of docker containers they could be using UTC. When sending the timestamp it is important to use ISO-8601 standard including the timezone.

Finally yes, extractors can have parameters. Right now we have it disabled in the GUI, but you can submit to an extractor parameters, that are passed into the RabbitMQ message. We use this for instance to pass in the google api key for specific extractors. You can use https://clowder.ncsa.illinois.edu/swagger/#/files/post_files__file_id__extractions and use parameters in the json body that is send when posting to that endpoint.

Hope this helps,

Rob

From: Boris Capitanu <capitanu@illinois.edu>

Subject: Re: Progress on Brown Dog Extractor

Date: August 5, 2019 at 11:41:01 AM CDT

To: "Bradley, Shannon" <sbrad77@illinois.edu>

Cc: "Kudeki, Deren Emre" <dkudeki@illinois.edu>, "Marini, Luigi" <lmarini@illinois.edu>, "McHenry, Kenton Guadron" <mchenry@illinois.edu>

Hello,

I'm writing to offer an update and ask for some help/clarifications.

The extractor I've been working on is designed to operate on volumes of text (encoded as ZIP files containing text files, one for each "page") and process them to generate what HTRC calls "extracted features". For information about what these extracted features are, please see https://wiki.htrc.illinois.edu/display/COM/Extracted+Features+Dataset

For purposes of this extractor, the "metadata" portion described at the link above does not apply, as that is specific to HathiTrust volumes and thus not available in the Clowder environment.

The TL;DR; of what the extractor does is as follows:

1. The volume ZIP file is downloaded from Clowder, and each text file inside the ZIP is loaded into memory (it is assume that a text file contains lines of text delimited by line breaks, as normally generated by an OCR engine applied to a page image scan)
2. An algorithm is run that identifies running headers/footers and segments each page into "header","body","footer".
3. An algorithm is run that tries to identify the predominant language on the page
4. A set of NLP algorithms are run to do sentence segmentation, tokenization, and part-of-speech tagging; token counts are also generated
5. A JSON document is constructed containing the extracted features (page-by-page, section-by-section) and aggregate information about the volume.

The extractor code is nearly finished. The main problem I run into now is that it's unclear what the best thing to do is with the output I generate.

As a first attempt, I tried attaching it as metadata to the file that was processed, but this failed because the generated JSON has a subtree which contains as keys the actual tokens extracted, one of which being the "." (period) which seems to break the JSON-LD metadata upload endpoint with the following error:

clowder_1 | ! @7cljj4jna - Internal server error, for (POST) [/api/files/5d465ef1e4b0bd9c93accc01/metadata.jsonld?key=1e7b256b-91bf-46e6-a028-3c486dc6768a] ->

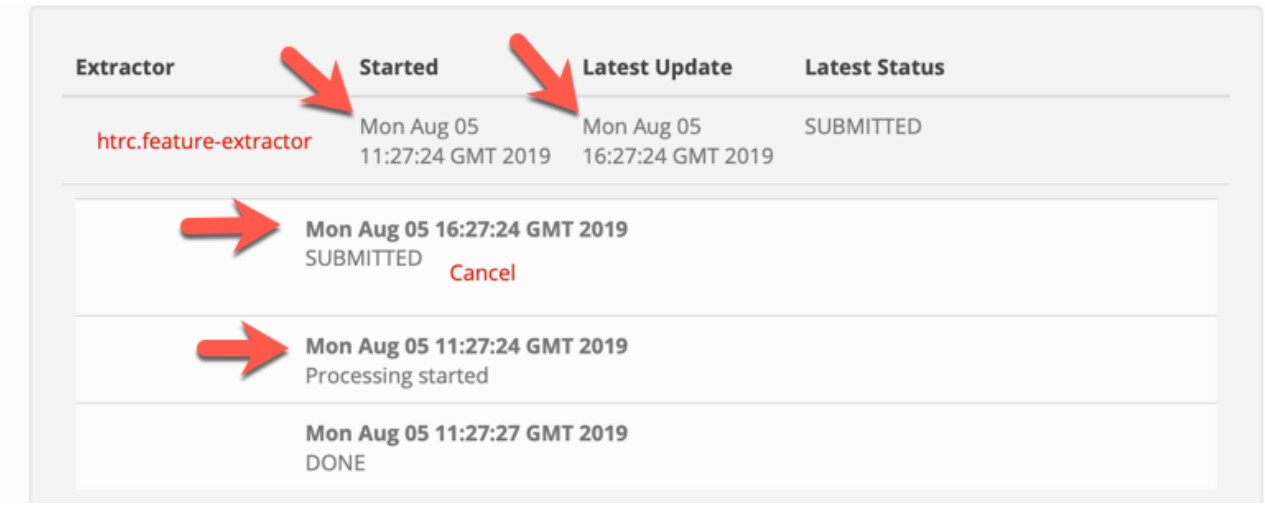
clowder_1 |

clowder_1 | play.api.Application$$anon$1: Execution exception[[IllegalArgumentException: Document field names can't have a . in them. (Bad Key: '.')]]

I am thinking that instead of uploading that as metadata, I should perhaps upload it as a new file in the same dataset, adding ".json" to the filename that was processed. Other options?

Through looking at the code of existing extractors (both Python and Java), I noticed that there's a "status update" mechanism through which an extractor can report on its status. I've implemented this into the extractor I've built, but have a few questions:

1. How does an extractor report that it's "DONE" so that the interface shows the extractor as "DONE" (instead of the current status which says SUBMITTED)
2. Are the status updates optional, or required (and perhaps the mechanism used to inform Clowder of "DONE"?)
3. Is it normal for the timestamps to be "all over the place" for the event stream listed in the Clowder UI for an extractor? (the actions below happened at the same time)



Another question: Can the extractors have parameters that the user can modify? If yes, how?

It would be helpful if it would be possible to set up a time to have a short technical discussion with someone from the Clowder team to address these and potentially other technical questions I have. That should speed up the timeline when this will be finished considerably (rather than me having to look through lots of code to discover myself). Please let me know if/when we might be able to do that.

Thank you,

Boris