Strategies for Correcting and Extracting Fields from Optical Character Recognition Products

Carver Bray, Collin Dixon, Robert Kelly, Liudmila S. Mainzer

Abstract and Background

Radiocarbon dating was invented nearly 70 years ago, and continues to be a crucial method for determining the age of historical objects, fossils and geological sites. Early records were compiled in the form of notched 5×8-inch cards, which still contain valuable information to modern researchers. Fred Johnson (1904-1994), an archaeologist at the Peabody Museum of Andover Academy, compiled 45,000 such cards for the dates 1959-1972 from all over the world, based on the reports and data published in the journal Radiocarbon. To make this information accessible to the scientists in our modern digital world, the University of Wyoming Libraries digitized the cards, and applied Optical Character Recognition (OCR) to the output. Our project focused on correcting and extracting the relevant fields from these records and organizing them for upload to the Canadian Archaeological Radiocarbon Database (CARD). Our Python codes automate this process, which can be used for other batches of cards of similar nature.

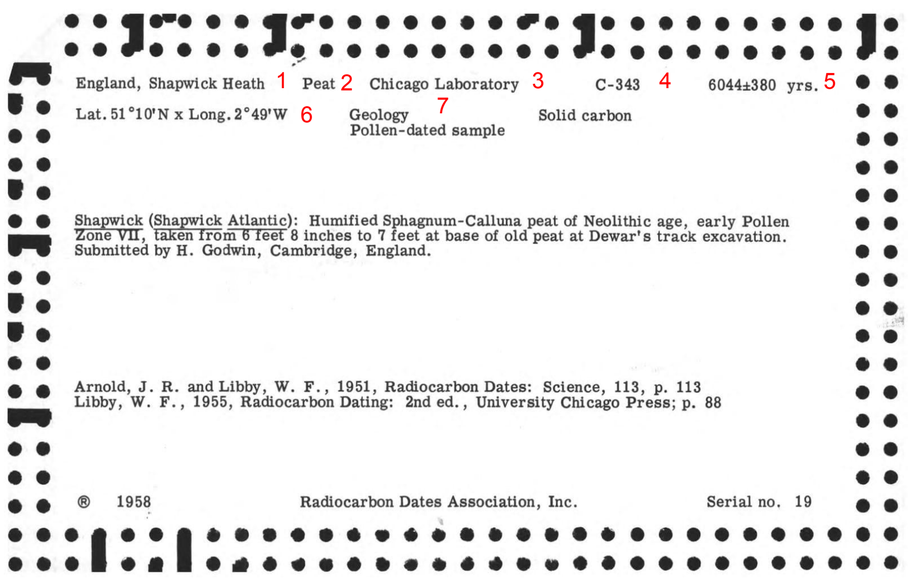
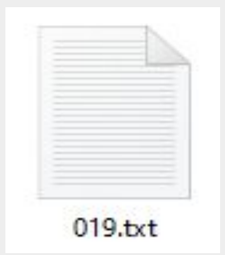


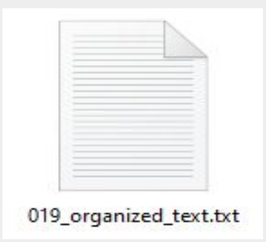
Figure 1. An example of our digitized Radiocarbon dating cards. In red we marked fields relevant for our analysis (also see Figure 2A).

Approach

We focused on exploitable patterns in the data fields: ages had symbols you could search, lab numbers have a specific format, etc. Regular expressions were applied to search for these patterns, then code was developed to correct errors for every data type. Two python packages were used for this purpose: Pdfminer.six for OCR and OpenPyXL for writing data to Excel. Due to the consistent font and spacing on the cards, the OCR was able to read in with few errors, but the size of the dataset still made errors relatively common. Many cards cannot be uploaded as a result.

Implementation and Workflow





The first step in our process was to run OCR on the input pdfs of the card images. The results can be seen on the Figure 2A, which marks the recognized fields as they correspond to the locations on the physical card from Figure 1.

Next we used our Python codes to organize this output into text files (Figure 2B) that can be easily converted into Excel spreadsheet. The codes made use of the exploitable patterns described below..

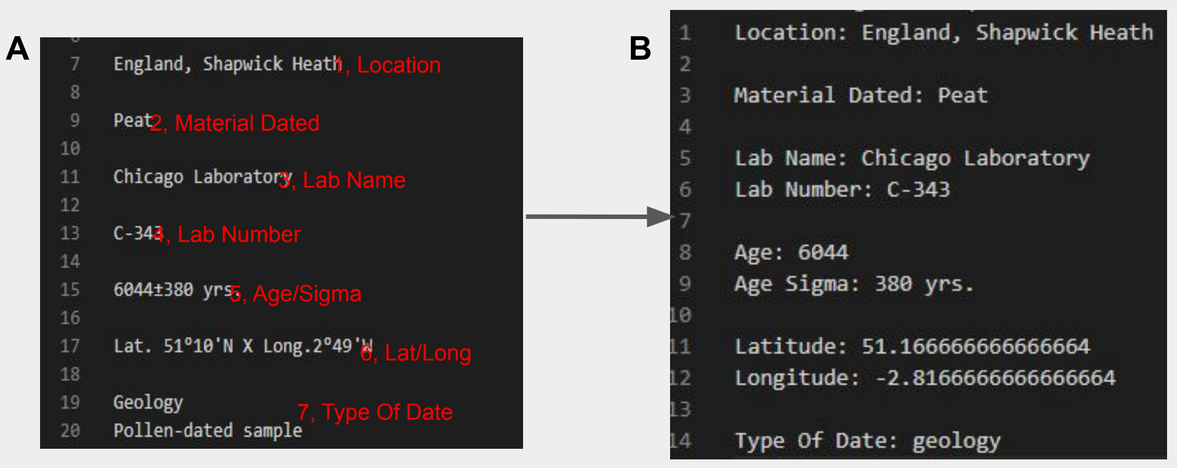


Figure 2. Output of the OCR software (A) and our codes (B) as we organized the digitized information by the relevant fields (in red).

In order to submit to CARD, the information must be in a designated format in an excel document. Data from the above text files were processed into a spreadsheet (Figure 3). From this spreadsheet, several final checks are also run on the fields.

Lab Number	Field Number	Material Dated	Taxa Dated	Type of Date	Locality	Latitude	Longitude
C-343		Peat		Geological		51.16666667	-2.8166666

Figure 3. Headers and an example entry into the Excel spreadsheet for CARD database.

Exploitable patterns

Material Dated:
Using spreadsheet data…
Create a list of valid materials to search for

Lab Name:
____ Lab/Laboratory/Survey
Keywords indicate that they represent the lab name

Lab Number:
Letters-Numbers
Lab Numbers follow the same format on all cards

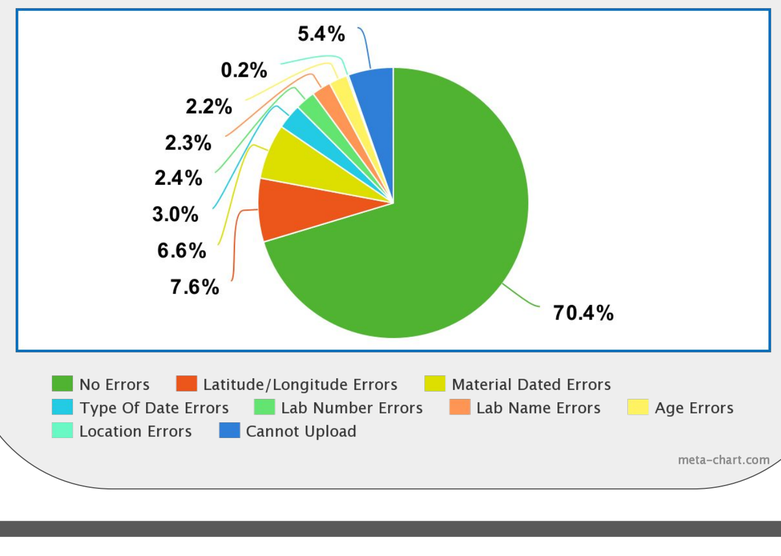
Age:
###±## yrs.
± and other related symbols are easily searchable

Latitude/Longitude:
Lat. ##°##”##’ N/S x Long. ##°##”##’ E/W
Lots of symbols/words to search for

Type Of Date:
Geology/Archaeology/Paleontology
Only 3 supported types in CARD database, search using regex

Results

We organized over 70% of the cards into the submittable spreadsheet. The other 30% are either missing one or two items, or cannot be uploaded due to the data types not being compatible with the CARD database.



Next Steps

1. The data will be imminently uploaded to the CARD database.
2. The errors will be further reduced by improving the initial OCR by exploring multiple OCR software packages.
3. Next we will fix the remaining problematic cards and make them uploadable as well. Most of the remaining issues are with locations and latitude/longitude. Verifying location will also help with latitude and longitude through cross reference. In order to extract locations more consistently, OCR correction may be used.
4. In the future, it may also be required to obtain a site identifier from the site names on the cards. This would most likely involve referencing site names to databases of site identifiers.

References

Pdfminer.six - PyPI. (n.d.). Python Software Foundation. Retrieved from https://github.com/pdfminer/pdfminer.six

OpenPyXL- PyPI. (n.d.). Python Software Foundation. Retrieved from https://foss.heptapod.net/openpyxl/openpyxl

Acknowledgements

Thank you to the UW Advanced Research Computing Center (ARCC) for supporting us on this project. We are also grateful to the UW Libraries for providing the dataset and mentoring us in the analysis. Much of this work was done on the Teton HPC cluster at the University of Wyoming, supported by the admin team at ARCC.