University of Iowa Libraries Staff Publications

6-23-2018

Leveraging Python to Improve Quality of Metadata of Engineering Faculty Publication Records (Board 96)

Qianjin Zhang
University of Iowa

Leveraging Python to Improve Quality of Metadata of Engineering Faculty Publication Records

THE UNIVERSITY OF IOWA LIBRARIES
LICHTENBERGER ENGINEERING LIBRARY

Qianjin Zhang

University of Iowa Lichtenberger Engineering Library

Background

Faculty profile systems that capture and showcase faculty scholarly activities and accomplishments are emerging in many institutions. The platforms of faculty profile systems include commercial platforms such as Activity Insight's Digital Measures, Elsevier's Pure and Symplectic Elements and open-source platforms such as Profiles and VIVO [1].

The University of Iowa has started migrating faculty information to Activity Insight's Digital Measures, locally branded as Academic and Professional Records (APR). The APR project is a collaborative initiative of the Office of the Provost, Information Technology Services and the University colleges to capture faculty information on teaching, research, grants, service, as well as records on professional accomplishments and interests. Since a record of publication would make a strong case for faculty excellence in scholarship especially for promotion and tenure, accuracy of publication records is significantly important.

Purpose

Since early 2017, the College of Engineering has steadily migrated their faculty data to the APR. Upon request by the College of Engineering and the APR project leader, we were to review engineering faculty publication records to improve the quality of metadata, especially focusing on Digital Object Identifier (DOI), ISSN, PubMed ID (PMID) and PubMed Central ID (PMCID).

Based on a rough evaluation of metadata quality, we realized that thousands of yet-to-be-identified records with erroneous and missing metadata would make a routine manual review time-consuming and costly. However, some libraries have implemented Python scripts in managing metadata for library resources. For example, the University of Minnesota Libraries used Python scripts to evaluate MARC record completeness for e-books [2] and librarians at the University of Virginia utilized Python to perform quality control on MODS records for digital collections [3]. Both examples indicate that Python would increase efficiency in quality control of metadata. In consideration of the challenges we are facing, scripting with Python would be an appropriate approach over the manual approach.

Workflows

Given that faculty publication records are imported from different sources such as PubMed, Scopus and manual input, our strategy was to evaluate record completeness, break down the large set of records into several subsets of records with some common patterns and then manipulate subsets, especially with a focus on identifying DOI, ISSN, PMID and PMCID.

Adding DOIs

We extracted the engineering faculty publication records in a csv file from the APR and briefly evaluated record completeness through checking for the presence or absence of DOI. We also sorted records by checking for the presence or absence of PMID or PMCID because PubMed records have PMID or PMCID, while Scopus records and manual input records have neither of them. For records that contain PMID and/or PMCID, we retrieved DOIs using PubMed's online DOI query tool.

For records that do not contain PMID or PMCID, we formulated references in the following formats and retrieved DOIs using the CrossRef Simple Text Query.

- Last Name, First Name (Year of Publication) Title. Journal title/Conference Name Volume# (Issue#): Start Page# - End Page#
- Last Name, First Name (Year of Publication) Title. Journal Title/Conference Name Volume#: Start Page# - End Page#
- Last Name, First Name (Year of Publication) Title. Journal Title/Conference Name Volume# (Issue#)

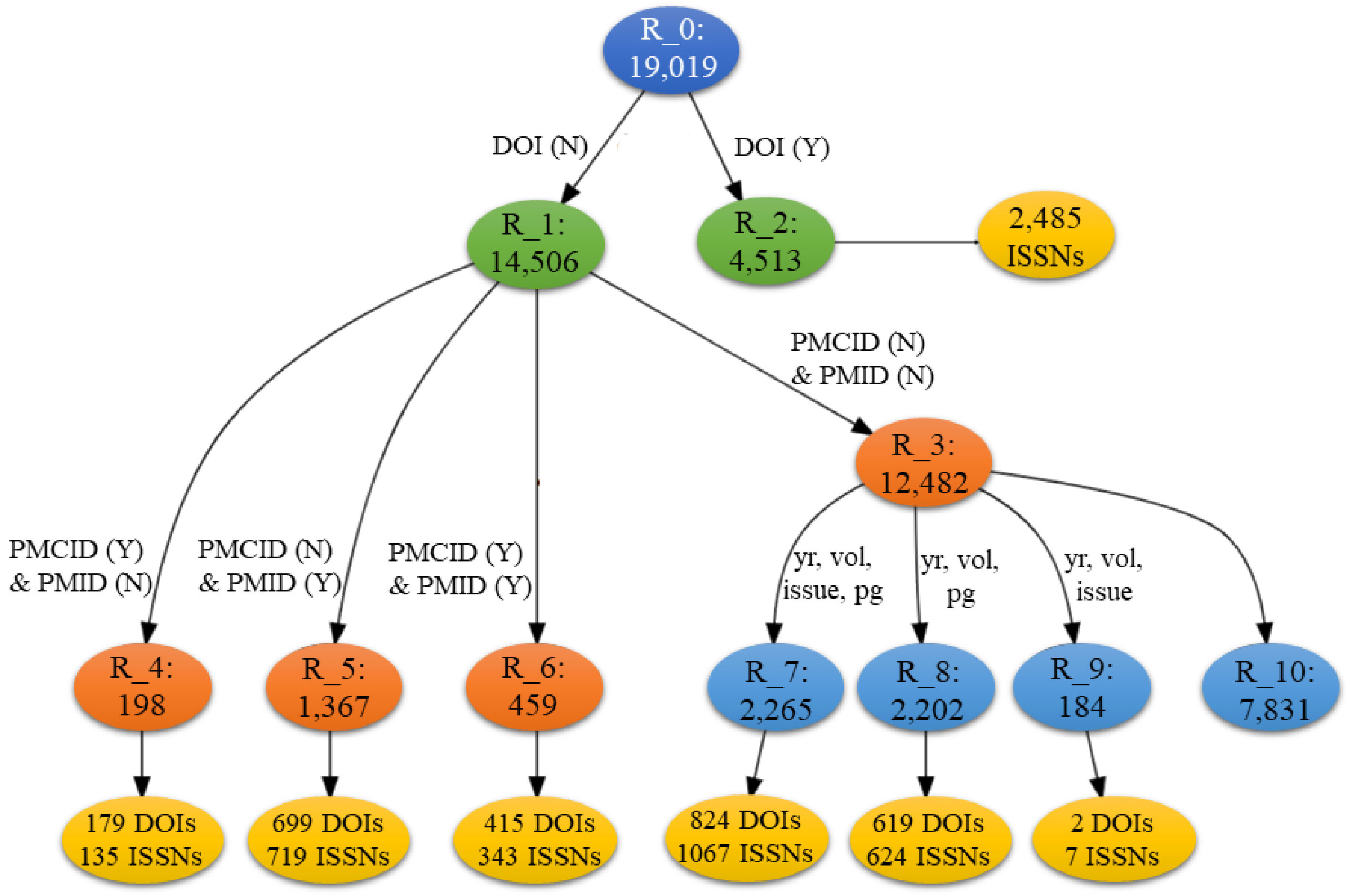

Figure 1: The Workflow of Batch-Processing Records

Adding ISSNs

Titles, title abbreviations and ISSNs were extracted from PubMed and Scopus and then formatted into a csv authority file. The authority file included title abbreviations because some records might have used title abbreviations other than full titles. For records that do not have an ISSN, we filled in the empty ISSN column with journal titles or journal abbreviations. If journal titles or title abbreviations were found as an exact match against the authority file, then they were substituted with an ISSN.

Results and Limitations

Figure 1 shows the exact number of records that we processed in the workflow. We found that 14,506 records in R_1 of the whole records do not have DOIs while 4,513 records in R_2 have DOIs.

Among the records with no DOIs, only a small portion of records have PMID and/or PMCID. In other words, the records in R_4, R_5 and R_6 are from PubMed while the records in R_3 are from Scopus or manual input. With regard to the records in R_4, R_5 and R_6, we identified 1,293 DOIs and 1,197 ISSNs.

For the records in R_3, we identified 1,445 DOIs and 1,698 ISSNs. We also identified 2,485 ISSNs for the records in R_2. As a result, we successfully identified 2,738 DOIs and 5,380 ISSNs for records that have missing DOI or ISSN.

However, this approach could not handle the remaining 7,831 records in R_10 partially due to incompleteness of metadata. We provided the results and discussed further about the project with the College of Engineering and the APR project leader.

Conclusions

We found the implementation of Python in engineering faculty publication records review process improves the effectiveness and efficiency of the review process, saves our library staff's time and makes a contribution to the College of Engineering's APR migration as the University of Iowa Libraries is increasingly involved in this campus-wide initiative.

References

1. Givens, M., L.A. Macklin, and P. Mangiafico, *Faculty Profile Systems: New Services and Roles for Libraries.* Portal-Libraries and the Academy, 2017. 17(2): p. 235-255.
2. Thompson, K.T., and S. Traill. *Leveraging Python to improve ebook metadata selection, ingest, and management*. Code4Lib Journal, 2017 (38).
3. Bartczak, J., and I. Glendon. *Python, Google sheets, and the thesaurus for graphic materials for efficient metadata project workflows*. Code4Lib Journal, 2017 (35).