A case-study:
Using the pdfminer python package and fuzzy matching to triage digitized legacy theses and dissertations for repository ingest

Ingesting digital resources into repositories provides a number of familiar challenges to systems administrators and developers, but dealing with missing or incomplete metadata is a shared challenge. Although developers try to provide automated solutions to create or enhance metadata, oftentimes the solution proves unworkable and the metadata creation must be performed manually.

This poster illustrates a detailed case study of an automated metadata solution using text extraction developed for the digitized legacy theses and dissertation collection into IDEALS, the institutional repository of the University of Illinois at Urbana-Champaign.

IDEALS has been the library's digital access point for theses and dissertations since 2009 when the University began accepting the electronic submission of theses and dissertations. These theses and dissertations are accessible from both a main "Theses and Dissertation" collection, as well as a department specific collection. Since new dissertations are ingested into IDEALS after they have been submitted to and reviewed by the Graduate College, assigning a department specific collection is not an issue. However, digitized dissertations that predate IDEALS have presented a different set of challenges..

As described by Shreeves and Teper, the University purchased 19,375 digitized dissertations dating back to 1949 from ProQuest. The digitized theses and dissertations were delivered with a set of raw MARC encoded descriptive metadata along with PDF files of each dissertation. The PDF files contained plain text data generated by an optical character recognition (OCR) software that represented, albeit imperfectly, the text of the dissertation. Analysis of the raw MARC metadata revealed that it was impossible to determine the original department that granted the degree. This information would need to be determined so that the correct department specific collection could be assigned. .Since

Despite the lack of metadata that could be used to classify the dissertations, the PDF files contained the text of the dissertations, we decided to develop an automated classification scheme. The granting department name, while not included in the metadata, was discernable in boiler plate text that occurred on the adviser's signature form which was included in the first few pages of each dissertation. We realized that extracting the text from this boilerplate would provide the metadata that we needed.. To do this we needed a way to programmatically parse the text that was included with each PDF, and identify the boilerplate text when it was encountered.

To parse the text that underlied the files, a python library called pdfminer (pdfminer 20140328), which acted as an API to extract plain text from pdf files, was utilized. This alone was sufficient to classify many of the dissertations by department. The department string was preceded by a one of a small set of phrases in almost all cases. However, the text, being the product of OCR software, frequently contained imperfections and misdetected characters. Typically these problems manifested as spelling errors where one letter was mistaken for one or two others (e.g. “u” is often identified as a double “l”). Because of these imperfections (often called OCR corruption), the boilerplate text could not be located via an exact match. Instead, we developed a fuzzy matching procedure based on an implementation of edit distance (also known as Levenshtein distance) to locate text that was very similar or identical to one of the phrases we were trying to locate. .When processing the collection of dissertation PDF’s, the program searched for text that had an edit distance of two or less from one of the strings that normally preceded the department name string. In other words, the text string could need up to two character edits (addition, deletion or substitution) and still be considered a match.

Once the department name string was identified, it was keyed to the ProQuest identification number for the dissertation. A list was then compiled of the set of extracted department name strings and appropriate collection handles were assigned to each. In some cases this process turned up terms that were not identifiable as department names. Dissertations with these unrecognizable strings were binned with those for which no string was found during the extraction process. Those that were assigned a collection were packaged and ingested in batches based on the collection handle that had been assigned.

This procedure proved effective in drastically reducing the number of dissertations that needed to be classified by hand. In totality, we were able to successfully match collections to 86.3 percent, 16,721 dissertations among 19,375 that the University purchased. “Fuzzy Matching” or “near matching” has long played a role in spell checking, search-term recommendation and auto-correction algorithms and is a relatively easy to implement technique from the machine learning repertoire. Given the prevalence of OCR processed PDFs in most institutional repositories, this type of “fuzzy matching” presents multiple possibilities for the automation of metadata extraction from noisy but searchable texts and one that costs relatively little in developer time and computational resources.

References

Shreeves, Sarah L. and Thomas H. Teper. “Looking Backwards: Asserting Control over Historic Dissertations.” *College and Research Libraries News* 73.9 (2012): 532-35. Web. 1 Feb. 2014

"pdfminer 20140328." *Python Package Index*. n. p., n.d. Web. 1 Feb. 2014.