Digital Curation and Machine Learning Experimentation in Archives

Teddy Randby
Advanced Information Collaboratory (AIC)
The University of North Carolina at Chapel Hill
Chapel Hill, USA
tedrandby@gmail.com

Richard Marciano
College of Information Studies
University of Maryland
College Park, USA
marciano@umd.edu

Abstract—In this paper, we present a series of experiments we conducted over the summer of 2020 with the FDR Morgenthau Holocaust Collections at the FDR Presidential Library and Museum, in order to unlock hard-to-reach information in the collections and improve access to the public and researchers. We extract detailed Subject Index metadata from Table of Contents images towards creating better finding aids. We demonstrate how digital curation of archival collections are a necessary preparation step for use with supervised Machine Learning algorithms. Finally, we introduce the notion of historical contextualization of Machine Learning models in order to create culturally-aware training models.

Keywords—Digital Curation, Computational Archival Science (CAS), Machine Learning, FDR Presidential Library and Museum, Morgenthau Holocaust Project, Machine Learning

I. INTRODUCTION

In February 2020, the Advanced Information Collaboratory (AIC) [1] was launched with partners from leading academic and cultural institutions, with an emphasis on: Exploring the opportunities and challenges of “disruptive technologies” for archives and records management (Artificial Intelligence (AI), Machine Learning (ML), Computational Archival Science (CAS), etc.), and Leveraging the latest technologies to unlock the hidden information in massive stores of records. AIC goals include: training current and future generations of information professionals to think computationally and rapidly adapt new technologies to meet their increasingly large and complex workloads, promoting ethical information access and use, and pursuing multidisciplinary collaborations to share relevant knowledge across domains. To accelerate investigations in these areas, in October 2020, the AIC launched a targeted AI / ML / CAS initiative called the Future of Archives and Records Management [2].

This paper represents the first FARM Initiative case study exploring the use of Machine Learning strategies to predictive modeling in the extraction of valuable hidden archival materials, using the Morgenthau Holocaust Collections Project, a privately funded project of the Roosevelt Institute (RI), the nonprofit partner to the Franklin D. Roosevelt Presidential Library & Museum (one of 14 Presidential Libraries operated by the National Archives and Records Administration). Other FARM Initiative case studies will follow, exploring the automation of historical archival collections, including: (1) Densho’s WWII Japanese American Incarceration Camp project, and (2) the Maryland State Archives Legacy of Slavery project. This work is partially funded through a 2-year (2020-2022) Laura Bush 21st Century Librarian Program grant called “Piloting an Online National Collaborative Network for Integrating Computational Thinking into Library and Archival Education and Practice”. The Piloting project is concerned with training the next generation of MLIS students. Hence, this paper tries to be very didactic and transparent in walking through an entire ML workflow. We eventually hope to capture this workflow through Jupyter Notebooks that will lead to reusable lesson plans.

While the use of AI and ML with historical archives is in its exploratory stage, it represents an emerging trend as seen by its investigation by a number of cultural organizations, including:

• Yad Vashem, the World Holocaust Remembrance Center, is hosting workshops on the changing relationship between the archivist and the users [3] in the age where large amounts of source materials and information are accessible online, with an emphasis on how this influences changes in methodology and tools.

• The Smithsonian Institution Data Science Lab [4], in collaboration with the United States Holocaust Memorial Museum (USHMM) is focused on enhancing the discoverability of its digitized collections through Machine Learning tools, with a particular interest in Deep Learning techniques to classify unknown document types and natural language processing techniques to delve into document contents, leading to interactive visualizations of results.

• The European Holocaust Research Infrastructure (EHRI) has a project on understanding memories of the Holocaust using neural networks [5].

• The Library of Congress “Newspaper Navigator” Dataset [6], has processed over 16 million pages from “Chronicaling America” using visual content recognition of newspaper articles, cropping of the visual boxes, OCR extraction within predicted bounding boxes, and extraction of metadata in JSON format.

• The National Archives (UK) and its AI for Digital Selection project [7] is focusing on duplicate detection, entity extraction, and classification.

• Virginia Tech University Libraries, through a project called “Ensuring Scholarly Access to Government Records and
Archives”[8], are exploring ways for Machine Learning to help improve access to government records at the US National Archives records, with workshops in the Spring 2021.

• The AURA Network[9] (Archives in the UK/Republic of Ireland & AI), has a goal is to unlock cultural assets that are preserved in digital archives closed to the public or difficult to access, with workshops on AI in 2021.

• The Croatian State Archives in Zagreb is exploring a WWII food rationing index card project, where digitized documents are shown to be a valuable source for Machine Learning (slides shared by Hrvoje Stancic from a Sep. 2020 presentation at the ENTRENOVA conference in Split, Croatia).

Our paper describes the Morgenthau Holocaust Collections Project, highlighting the need for digital curation to help prepare the collections for ML processing. We experiment with supervised training to develop a predictive model that allows the automation of deep metadata extraction.

II. THE MORGENTHAU HOLOCAUST COLLECTIONS PROJECT

The Morgenthau Holocaust Collections Project[10], is a privately funded project of the Roosevelt Institute[11]. Of particular interest are a number of key areas including digital curation and Machine Learning, each furthering the Morgenthau Project’s mission to develop new pathways for digital scholarship on Roosevelt’s response to the Holocaust.

During Henry Morgenthau, Jr’s nearly 12 years as FDR’s Secretary of the Treasury from April 27, 1933 to July 27, 1945, a daily record of his official activities was compiled, including transcripts of his meetings and telephone conversations as well as copies and originals of the most important correspondence and memoranda that passed over his desk. “Henrietta Klotz who served as Henry Morgenthau Jr.’s primary personal secretary supervised the curation and indexing of the Diaries, a collection of 864 bound volumes, with each volume typically covering 1-3 days. Treasury Library Isabella Diamond executed the binding process and systematic subject indexing.”[12].

We focus on Series 1, the Morgenthau Diaries, and Series 3, the Subject Index to the Morgenthau Diaries.[13].

Series 1 consists of 881 .PDF files, corresponding to the 864 physically bound Diary Volumes arranged for the most part in chronological order. Each of the Volumes contains a Table of Contents (TOC).

![Fig. 1-a: Bound Diary Volumes from Series 1 and part of the TOC for Diary Book 696, Jan. 22-26, 1944](image)

Series 3 consists of 65 .PDF documents arranged by alphabetical topics and corresponding to over 30,000 physical Subject Index cards.

Next, we show the “War Refugee Bd.” Subject Index card that refers to the above TOC Volume page.

![Fig. 1-b: “War Refugee Bd.” Subject Index card from Series 3](image)
III. Digital Curation in Preparation for ML

We demonstrate how the Morgenthau Project website, which is structured as a finding aid, is crawled and restructured into a “Content Tree” that can then be used to drive the Machine Learning algorithms.

A. Series 3 Index Card Processing using traditional OCR

Series 3: Morgenthau Diaries Index

Henrietta Klotz, personal secretary to Henry Morgenthau, Jr., compiled an extensive, card-based Subject Index to Morgenthau's bound Diaries. The cards are alphabetically arranged, and each Subject Entry cites to a Diary volume and its subsequent page number (top right corner, numbers separated by a colon), the given document's date, along with any cross references that may apply (usually indicated by "see also:" note). For example, 187:26 means Volume 187 at Page 26.

We started by processing the Series 3 Subject Index cards themselves. From the FDR Library Series 3 Webpage [14], we created a workflow with the following five steps:

1. Parse the HTML for Series 3 of this Webpage (using a Python script).
2. Crawl the 65 Index Card multipage .PDF files (we used the HTTrack software).
3. Explode the PDF files into individual images resulting in over 30,000 .JPG files.
4. Run traditional OCR on each Image Header applying a static recognition Area box (this is done with the ABBYYFineReader OCR software [15]).
5. Collect all resulting Subject Entries.

Using the above card, we extract Subject Index “War Refugee Bd.”. While processing in this manner all 30,000+ cards, we extract 6,361 unique Subject Headers (uniqueness was facilitated by the use of the OpenRefine data wrangling software).

This produces a systematic and comprehensive list of Subject Index entries that can be subsequently used during the ML validation process.

B. Series 1 Bound Volume Processing for ML Workflows

From the FDR Library Series 1 Webpage [16], we analyze the structure of the Webpage entries. For example, the entry for Volume 2 is:

Volume 2, July 1-December 31, 1934 View Online (Part 1); View Online (Part 2)

From this entry we extract the following metadata attribute/value pairs (through a simple Python code script):

Volume #: V002
Dates: July 1-December 31, 1934
Number of Parts: 2
Filenames of Parts: md0002.pdf, md0003.pdf
Size of leading TOC pages: 12 pages

A list of all the Parts Filenames is assembled to drive the Web Crawler process that extracts the data from the webpage finding aid (using the HTTrack software). The overall workflow has the following six steps:

1. Parse the HTML for Series 1 of this Webpage.
2. Crawl the 65 Index Card multipage .PDF files (we used the HTTrack software).
3. Extract TOC each from part file.
4. Explode TOC into individual images.
5. Populate Table.
6. Use Table to Create a Content Tree

The Content Tree for Volume 1, looks like this:

The resulting Content Tree for all Volumes is:
The individual TOC JPG files can now be used by the ML algorithms. It is worth noting that we store the entire Content Tree in the cloud in the U. Maryland’s box cloud storage [17]. The ML step makes use of box’s API interface to directly access the TOC content.

IV. 2-PHASE MACHINE LEARNING FOR PREDICTIVE MODELING

While a more traditional approach was to run OCR (optical character recognition) on the TOC images to extract text, and then use NLP (natural language processing) models to label the data, we settled on a second approach, we decided instead to: (1) use object detection as the first step to identify Subject Index (SI) image boxes (object detection within images), and (2) further label the objects contained within the SI image box objects and extract the sub-object underlying text.

Thus, our overall Machine Learning strategy consists of a 2-phase ML approach with: (1) Google AutoML Vision and (2) Nanonets Deep Learning & OCR.

Nanonets is a commercial service which combines Deep Learning and OCR in one phase, meaning that one trains it to locate objects in the images that correspond to one of these 6 unique labels: Book, Page, Header, Content, Date, and Handwritten section. It then locates these types of regions and performs the text extraction in one step (labeling and extraction all in one).

Phase 1 with Google AutoML Vision “Object Detection”:

• Run AutoML Vision on the TOC JPG images. The output is a JSON file with the page SI block locations:

Phase 2 with image processing and Nanonets:

• Use the JSON locations to “carve out” the individual SI block images.

• These SI block images are then fed to the second ML program, Nanonets (https://nanonets.com)

Results:

• This 2-phase ML pipeline processes a total of 3,579 TOC JPG pages, discovers a total of 22,667 SI boxes (an average of 6.69 boxes per page), and produces a final database of tagged entities.

It is the output of this 2-Phase ML process that is meant to produce a more usable finding aid.

The remainder of the paper focuses on the 1st phase of this ML pipeline as work with Nanonets has been validated but is ongoing.

V. SUBJECT INDEX DETECTION USING ML

We used Google Cloud “AutoML Vision” Object Detection on the TOC JPG images. This enables the training of customized machine learning models to detect and extract similar objects and match them to user-defined labels.


In order to train a recognition model for extracting the Subject Indexes from the TOC Diary Volume images, we
defined a “box” label, and manually labeled each SI entry for 351 TOC images (or 10% of the total of 3,579). Google recommends using about 1,000 annotations per label. Our 351 images resulted in 2,630 annotations for the “box” label, which is the reason why 351 is sufficient. In general, the more images per label, the better the model will perform.

When training Machine Learning models, the dataset is typically divided into three separate datasets: (1) a training dataset, (2) a validation dataset, and (3) a test dataset. These are often referred to as “dataset splits”. AutoML Vision Object Detection splits the dataset randomly, with 80% of images used for training, 10% of images used for tuning, and the remaining 10% of images used for evaluating the model, thus our split was 2,104 for training, 285 for validation, and 241 for testing.

The main Google AutoML Vision Object Detection steps [20] are: (1) **Import** data samples, (2) **Train** a model [When you have a solid set of annotated training images with bounding boxes and labels, you are ready to create and train the custom model], (3) **Evaluate** the model [CloudML Vision Object Detection provides an aggregate set of evaluation metrics indicating how well the model performs overall. In particular, precision and recall measures (21)], and (4) **Deploy** and run the model to make predictive analyses

### A. Import Data Samples

We initially used the CloudML Vision interactive user interface.

![Fig. 9: Importing 351 TOC JPG files](image)

### B. Train the Model

Since we had enough images, we were able to launch the training of the model phase, which takes several hours in the background. One is notified when the model is ready.

### C. Evaluate the Model

**Precision and recall** values for this model are:

- **Precision:** 96.5%  
  - The frequency of predictions that were correct (positive). The higher the precision, the fewer false positives predicted.
- **Recall:** 80.08%  
  - The frequency of bounding boxes that were successfully predicted by the model. The higher the recall, the fewer false negatives, or the fewer predictions missed.

Precision and recall give an indication of how well a model is capturing information, and how much is being left out. **Precision** tells us, from all the test examples that were assigned the “box” label, how many actually were supposed to be categorized with that label. **Recall** tells us, from all the test examples that should have had the “box” label assigned, how many were actually assigned the label.

**Average precision** at 0.5 IoU: 0.892

- A measure of how well the model performs across all score thresholds by calculating the area under the precision-recall trade-off curve (0–1 range).

- **Intersection over Union** (IoU) is a parameter of the evaluation metrics for object detection. It indicates the amount of required overlap between a predicted bounding box and a test set (ground truth) bounding box.

![Fig. 11: Model evaluation](image)

**Fig. 10: Annotate “box” areas on each of the TOC JPG files**
D. Deploy the Model and Run Interactive Predictions

We show the predictive output on 2 pages. The model outputs a series of numbers that communicate how strongly it associates each label with that example. If the number is high, the model has high confidence that the label should be applied to that document.

Fig. 12: Interactive model predictions

VI. ASSESSING THE RESULTS

A closer examination of the results reveals patterns of mistakes that the ML model makes. These can be broadly categorized into two parts. These come in the form of: False Positive: The model predicts an entry where there isn’t one; and False Negative: The model fails to predict an entry where there is one.

These are unavoidable ML mistakes, and they translate into terms often used to describe machine learning models, called precision and recall. A general definition of precision is defined as:

$$ \frac{|\text{Correct SI Entries}| \cap |\text{Predicted SI Entries}|}{|\text{Predicted SI Entries}|} $$

Narrowing this definition to our context:

$$ \frac{|\text{Correct SI Entries}| \cap |\text{Predicted SI Entries}|}{|\text{Predicted SI Entries}|} $$

Thus, precision measures the proportion of all predicted SI-Entries which were correctly drawn. The precision of our model was calculated to be 96.5%. Roughly 96% of the SI-Entries it drew were correct. High precision means a lower chance of producing a false positive: drawing a box where there shouldn’t be one.

A general definition of recall is defined as:

$$ \frac{|\text{Relevant Documents}| \cap |\text{Retrieved Documents}|}{|\text{Relevant Documents}|} $$

Narrowing this definition to our context:

$$ \frac{|\text{Correct SI Entries}| \cap |\text{Predicted SI Entries}|}{|\text{Correct SI Entries}|} $$

Thus, recall measures the proportion of all correct SI-Entries which were drawn. The Recall of our model was calculated to be 80%. Roughly 80.08% of the total correct SI-Entries were drawn. High recall means a lower chance of producing a false negative: failing to draw a box where there should be one.

These numbers are calculated during the training of the model, using the “test” split of the dataset as a ground truth to compare its predictions to. In practicality, these comparisons can be flawed. A closer examination of the false negatives follows:

Fig. 13: False Negative #1

In the figure above, the red label is what our model predicted, and the grey label is the objective truth from the manually annotated dataset. When calculating precision and recall, Google determined that these two labels were distinct
enough to categorize this as a false negative. But upon examination, the predicted label captures all of the relevant information for that entry. For our purposes, this is a successful data capture.

But upon examination, the predicted label captures all of the relevant information for that entry. For our purposes, this is a successful data capture. For our purposes, this is a successful data capture. For our purposes, this is a successful data capture.

The false positives also contain special cases:

- **Fig. 14: False Negative #2**
  
  Above is another instance, where the model predicted a box with an extra amount of whitespace, but still resulted in a successful data capture. Many of the false negatives detected during training were in this category, the other half were not predicted at all. Following is one example:

- **Fig. 15: False Negative #3**
  
  Again, the grey label is the ground truth annotated by hand. The model made no predictions at all in this case, and therefore this is defined as a false negative. In this case however, no data lost. The SI-Entry missed by our model did not have any relevant data, or even an index. It is just a reference to another entry. This entry is not useful to us anyway, so cases like this can also be considered successful data-capture.

- **Fig. 16: False Positive #1**
  
  This false positive is most likely the result of a mistake made during manual annotation. If the label for this entry was not drawn during annotation and is therefore absent as the ground truth during comparison, the model will interpret this as a false positive.

- **Fig. 17: False Positive #2**
  
  Other false positives are true mistakes on the part of the model:
Here the model captures several SI-Entries inside one large label. Although this is technically a false positive, it is a mistake that can be remedied by later processing steps. The second layer of machine learning and our parsing techniques can account for this and separate out the nested entries.

Fig. 18: Other types of False Positives

In this case, the first layer model lumped two SI-Entries together under one bounding box. The second layer correctly identified the individual parts of each nested SI-Entry. While parsing this clip, the parsing technology will detect the presence of two entries and separate them into their own data-entries.

Moving away from false positives and negatives, errors can occur during recognition, even if the model correctly determines the presence of an SI-Entry. To the left is an example of one such case: a truncation. The model draws the bounding box too narrow, and truncates the beginning of the header as well as the end of the entries. This problem can be fixed easily with some clever parsing. Because the SI-Entries we care about never overlap in the x-direction (i.e., there will never be two SI-Entries side by side), we can ignore the left and the right limits of the bounding box. Instead of clipping out boxes, we vertically slice the entire page at the y-values given by the bounding box.

Another error during prediction is an overzealous prediction:

Fig. 19: Overzealous Prediction

This error occurs when the model includes some of the content from a previous or following SI-Entry into a given bounding box. This results in one of two situations:

- The extra line(s) is captured again in the bounding box for the previous entry, resulting in a data overlap. In this case, the line can be thrown out during parsing because it is captured already by a different entry.
- The extra line(s) is NOT captured by a different bounding box, resulting in this label “stealing” another’s data. In this case, in order to not lose the data, the PSM needs to somehow attach this data to a different entry on the same page.

Determining how this situation should be parsed needs to be informed by the commonality of these two situations. If the former is true for greater proportion of cases, then ignoring extraneous data is the desirable behavior. However, if the latter is true, then more work needs to be done during parsing in order to preserve this data.

Given this exploration of mistakes made by our model, we classified the errors into the following figure:

<table>
<thead>
<tr>
<th>Shorthand</th>
<th>Description</th>
<th>Example</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRNC</td>
<td>Some content of the entry is missed by the box in the x-direction.</td>
<td><img src="image1.png" alt="Example Image" /></td>
<td>This can be fixed by clipping out entries with the width of the page.</td>
</tr>
<tr>
<td>EXTRA</td>
<td>Some content of another entry is included in this and NOT in another entry.</td>
<td><img src="image2.png" alt="Example Image" /></td>
<td>This can be fixed with some more complex state while parsing. Each entry would need information about its surrounding entries.</td>
</tr>
</tbody>
</table>
Interpreting the training data shows that both the false-negative and false-positive rate are even lower than calculated, and some of the true shortcomings can even be overcome by smarter parsing on the backend and the second layer of machine learning.

One of AutoML Vision’s suggestions is to “Capture the variation in your problem space”, in other words create a broader data selection to make for a more robust model. The 351 files we selected were actually based on that premise. The figure below shows a red line which captures the number of Books (out of 864) that cover a particular year. For example, in 1941 there are 137 bound Volumes (meaning greater activity). The blue bar section of the graph indicates the number of pages we sampled from that year. You will note that our blue distribution tracks the number of bound volumes for that same year.

Hence the dataset we selected to create the model was historically contextualized. We speculated that this would lead to a more accurate and robust model. This is an assumption we would need to validate. However, we believe that creating culturally and historically aware datasets is an endeavor worth exploring. Future work on this collection will be to connect the 2nd Nanonet-based machine learning phase to produce the enhanced finding aid and build and test a user-interface around this finding aid.

VII. CONCLUSIONS, AND FUTURE WORK

This paper represents a preliminary investigation of applying supervised machine learning techniques to historical collections. The results so far are very promising.
ACKNOWLEDGEMENTS

We wish to acknowledge funding from the 2020-2022 IMLS Laura Bush 21st Century Librarian Program: “Piloting an Online Collaborative Network for Integrating Computational Thinking into Library and Archival Education and Practice.” Also, a deep appreciation for the support of staff from the FDR Presidential Library: Kirsten Carter, Supervisory Archivist, Bill Harris, Deputy Director, and Dr. Abby Gondek, Morgenthau Scholar-in-Residence, Roosevelt Institute at the FDR Presidential Library & Museum. Also, we wish to recognize the contributions of Renee Geary, Librarian and member of the 2020 Digital Curation for Information Professionals Certificate who performed a series of digital curation analyses on this collection that informed the development of this paper.

REFERENCES

Fig. 22: Creating culturally-aware training models