BAM! Lessons Learned and Potential Future Work

Original document

Usability/User evaluations

- BAM! Mockup evaluations
  - Mockups
  - Results
- High level takeaways:
  - Different kinds of browsing experiences were shown to undergrad and graduate students at Green Library at Stanford. Of these, subject, call number, and timeline browse generated positive reactions. A few participants noted how geographic browse might be useful if linked to where the books were about.
  - Subject descriptions tied to Wikidata pages. A few noted the Wikidata pages did not seem helpful or informative. Some participants mentioned Wikipedia as a potential starting point or source and seemed more familiar with Wikipedia than Wikidata.

Data, APIs, and Indices

- For author and subject headings, we set up separate indices enabling quicker navigation of relationships and/or discovery based on specific fields.
  - Subject heading and LCCN index
    - Querying the backup Fuseki endpoint set up by Dave Eichmann, we retrieved a list of all LCSH subject headings that also had a LC classification number assigned. There are 89,068 entries in this index, where each entry has the title of the subject heading, subject heading URI, full classification numbers/codes and classification facet that includes the first and first two letters of the classifications for that subject heading. This index enables looking up which subject headings correspond to classifications that start with a given letter or given two letters.
  - Author index with information from Wikidata and LOC for birth and death dates as well as start and end dates for activity
    - There are 1,558,367 entries/documents in this search index. All of these have LOC URIs and 949,446 have Wikidata URIs.
    - Separate scripts were written to populate this index from query results for birth and death dates based on RWO URI and authorized heading strings from LCNAF using queries to a backup SPARQL endpoint version of Dave Eichmann's LOC triplestore. Wikidata URIs, birth and death dates, and start and end dates for activity are added to the index using queries to the Wikidata SPARQL endpoint. Image URLs are added from Wikidata as well.
    - Since URIs were coming from two different sources, there were situations where there were Wikidata results that were not included in the original set of LC identifiers and titles (that were retrieved for only those items that had LC birth or death dates). Development-related results below elaborate on this issue further. At this point, there are 22,910 entries that don't have an LOC label/title.
- Geographic browse: This map-based view used the 500 top level subject region facet values from the production catalog. The authorized heading strings were used to query for FAST URIs that could then be used to query Wikidata for coordinates.
- Item-level subject information: OCLC, LCSH and Worldcat (using OCLC work ID)
- Call number: LCCN classes
- Syllabus: Initially, we used a database with sample data from OpenSyllabus made accessible through a Rails server enabling direct requests for data. After the end of BAM!, the code has been updated to use an OpenSyllabus API.

Development-related results

- Item View Subject Browse
  - The first pass at this produced mixed results. If a work had an OCLC work ID, then we used that do get the FAST data for the work. From there, we extracted the LCSH URIs and used those to get broader and narrower terms. In some cases we got the type of broader and narrower terms we expected. The term "Irish fiction," for example, returned "Irish Literature" as a broader term, and "Romance Fiction, Irish," "Historical Fiction, Irish" and "Children's Stories, Irish" as narrower terms.In other cases, however, the queries produced results that were often so broad or so narrow as to be irrelevant.
  - So the second pass took the subject areas already associated with the work and did LCSH queries to get the broader, narrower and related terms. For those works with CLC work IDS, we used Worldcat (http://experiment.worldcat.org) to get additional LCSH URIs that had been associated with the work. The results were generally better, though some works do produce long lists of narrower topics.
- Call Number Browse
  - This browse relied solely on the Catolog's index, primarily because the LOC only has a subset of the entire Library of Congress Classification system available as linked data. But the indexed call number classifications and labels enabled us to display the labels at the top of the Call Number Browse. So for this call number, D17.T39 C5 1982, the user sees "D - World History > D - History (General) > D1-24.5 - General" above the scrolling area containing works.
  - There were a few issues that provided challenges for this browse feature. First, not all of the indexed works include the field that contains the LC classification labels. As a result, this produced inconsistencies when trying to update the classification labels during scrolling. Another issue is that there are instances where multiple works, even those with different titles, share the same call number. Because of this situation, fetching additional works when scrolling would sometimes result in the same set of works being retrieved. And finally, there are some call number inconsistencies in the index such as spelling inconsistencies ("Oversize," "Overszie," "Oersize") that produce unusual results in the browse as well.
- Syllabus browse
  - Presenting a list of works frequently assigned on the same syllabus as a given work in the catalog produces pretty reasonable-looking recommendations. It looks promising as an Amazon-like recommendation engine.
  - Showing the user book covers in the recommended works list made the feature more compelling. The JavaScript function that fills book cover images from Google Books was set up to run upon page load, so it had to be re-run to fill in cover images from a client-side script.
  - We also created a view of syllabus recommendations by subject area. Open Syllabus Project's subject area notation are not very detailed, and thus the feature is of limited use.
- Author timeline
  - Index population: Separate scripts were written to iterate through the set of Wikidata URIs and their matching LC URIs that did not have authorized heading labels/titles within the index. One script iterated through the set of URIs that didn't have titles from LOC and individually retrieved the JSON from id.loc.gov for each entity and then obtained the label and RWO URI and updated the index. Originally there were 667,894 items that needed to have the label added from LOC. After running the scripts (more than once), there are now only 22,910 items that don't have an LOC label/title. It's unclear why the script did not work with these items and a follow up task would be to explore where/why labels were not retrieved for these items.

- For the front-end, we used Histropedia (see http://histropedia.com/histropediajs/documentation.html) to generate the timeline display. Histropedia expects the data in a particular format (i.e. a JSON file with an array of elements with specific fields).
  - When the page loads, a portion of the author index is requested and then transformed into a JSON structure which is then loaded into Histropedia. We further divided the times represented on the timeline into segments that were added to the display to (a) enable clicking on the segment to scroll the timeline to that year and (b) load the data from the index for that time period.
  - Additionally, we added code to generate a mini-knowledge panel as well as search results from the catalog for a particular author heading. The knowledge panel shows label/name and date information from the author index and supplements the information using a dynamic call to Wikidata to retrieve occupation information.
  - Clicking on the author name or "Search results" heading for the search results will open up an author facet search for that string in the catalog.
  - We used the "DENSITY_HIGH" option which is meant to reduce clutter on the timeline by hiding some of the items based on zoom level. Histropedia also uses rank to determine what to show and hide but we didn't assign any rank to the items being displayed.

Possible enhancements and areas for improvement

- Our initial exploration into looking at linked data connections for the subject era facet encountered issues finding equivalent FAST URIs for the subject era headings. When LCSH headings are converted to FAST, they may yield headings such as "1700-1799" for 18th century but that year range will not have an equivalent FAST URI. Although we retrieved some era facet values, we did not pursue looking up FAST URIs for bringing in linked data after encountering this challenge.
- Author and subject level browsing both rely on separate indices. This enabled exploration of what data points were required without updating the production index. Future work should look at how to update or better integrate information from the main production index and authorized heading indices. Alternatively or separately, we could look at filtering the subject or author data to include only those headings that are in use within the catalog.
- Author index: As noted above, there are 22,910 entries that still don't have any LOC labels. We would need to review the scripts populating the label to see why these particular items continued to not retrieve labels.
- Subject index: We are currently pulling broader and narrower relationships dynamically from id.loc.gov once a subject is selected. Future exploration could incorporate these relationships into the index or in a form that allows for analysis of the hierarchy.
- The subject browse page allows for selection of a top level classification facet and a subject indexed with the facet. The other information is pulled in dynamically so selecting one of those subjects using the "browse" button from the search page won't populate the browse page with that subject. Future work could look at enabling the page to be given a subject heading at any level of the hierarchy and having that subject be selected and displayed on the browse page.
- The author timeline currently uses a subset of the entire author index when first displayed. Clicking on a subsequent year/range then moves the timeline to that year and then loads data from the author index for display. Additional work may include more streamlined loading of data, perhaps depending on when the user navigates to another section of the timeline. Further work could be done to also link search results into the browse view by allowing the authorized heading name to be provided to the timeline which could then select and highlight that author.
- At the time we started BAM!, there were only six schedules from LCCN that were available as linked data. Conversations with LOC developers indicate additional conversions or re-conversions of schedules may occur in the future and these conversions may capture more of the broader and narrower relationships between classifications based on the LOC records themselves. If this is the case, we should revisit call number and subject browsing to incorporate this information.
- Both the item view subject browse and call number browse could be improved. The former only needs some UX refinement in how the additional subject areas get displayed. The call number browse would need more work, but there are possible ways to get around the issues described previously, and enhancements to improve the feature: for example, additional browsing/navigation using the LC classification labels.
- Our syllabus browse proof of concept was implemented with a temporary, custom built recommendation server but Open Syllabus Project has more recently released an API that could be used for the same purpose. The proof of concept implementation was all front-end code, but the new API has an auth key that will necessitate back-end code to keep the key secure. Since the ending of this phase, we have incorporated the use of this API into the code.
- Additional possibilities for future exploration of browsing include intersections of temporal and geographic browse and extensions or combinations of many of the browsing areas we investigated in this phase.
- We didn't include any tests within the code and if this code were to be made production-ready, we should incorporate testing.
- Additional user research and usability evaluations could provide directions for improving the UI.