SMASH! Lessons learned and future work

Link to original doc

Usability/User evaluation High Level Takeaways

- Auto-suggest is a useful feature which could benefit from additional support for misspellings. Furthermore, the label “author” could be revisited to clarify what the list indicates (i.e. works by versus works about). Potential exploration could be undertaken around how to provide a mechanism to distinguish between results (especially if they have the same string value) to enable users to know which result to select.
- Related person and subject suggestions have the potential to be useful, but additional steps could be taken to clarify how these entities are related as well more clearly lay out the knowledge panel design to display supplemental information such as contemporaries for authors. Furthermore, labeling could be improved for related concepts. For example, we could use the term “similar” instead of “close” for closely related concepts. Additionally, clarifying that the suggestions are clickable could be clearly indicated using a pointer above the item.
- Most participants indicated zero-results suggestions were a useful feature. Some participants indicated that the highlighted snippets provided useful context around how these suggestions related to the person’s query. Additional labeling changes could be made to clarify that the subjects on the left are suggestions and that the results displayed are related searches.
- For details, please see Tasks, SMASH! Usability Testing notes with more information around high level takeaways and screenshots of the system

Data, APIs, and Indices

- Autosuggestions were based on results from the author and subject headings used in the catalog, with facet counts being used for scoring. Earlier experiments included connections to the VIAF AutoSuggest API, to the VIAF data in DAVE, and to the LCNAF API.
- Zero results pages use data from Google Books API to get results based on full text search using the entered query. Additionally, we retrieved HathiTrust XML for a specific query to display a proof of concept using subject headings and search results based on full text search in HathiTrust based on the entered query. Since then, HathiTrust has provided IP address-based access and access for specific user accounts to an XML version of results for any given query. We have integrated this URL into the code.
- Subject and author suggestions displayed on the left-side of the page relied on the following:
    - Subject search:
        - LCSH search through QA (cached) with partial or complete string matching. This search takes into account both preferred and variant labels.
        - Subject facet values from the catalog for search results for that query. Queries to id.loc.gov retrieved the URIs for the subjects based on the authorized subject headings in the facet values.
        - Annif recommendations based on the query which return both the label for the LCSH heading as well as the URI. (Details on our use of Annif are recorded here).
        - For LCSH and FAST headings, requests for the URL for that subject to retrieve information about that entity including broader, narrower, and close match/related headings. If any of the subjects have a close match with a Wikidata URI, the Wikidata link is also included.
    - Author search:
        - LCNAF search through QA (cached) with partial or complete string matching. This search takes into account both preferred and variant forms of names.
        - Author facet values from the catalog’s search results for that query. Queries to id.loc.gov retrieved the URIs for the authors based on the authorized heading strings in the facet values.
        - Contemporaries within the person details view were retrieved from a query to the author index used in BAM! which captured birth dates, death dates, start activity dates, and end activity dates. When the user clicks on the author result, the first call to this index is to retrieve the appropriate Solr document containing birth/death info. The subsequent call is to the index is to retrieve any individuals whose birth date or death dates, whether from the Library of Congress or Wikidata, fall within the range of the birth to death dates for the author. For the dates for the main author (or the author whose details are being viewed), Wikidata dates are preferred over Library of Congress dates where available.
        - Wikidata queries to retrieve image and schema:description for authors using the Library of Congress URI to query for the corresponding Wikidata entity.

Development-related and UI-relevant results

- Auto-suggest: Our initial tests using VIAF, VIAF via DAVE and LCNAF as data sources provided mixed results. Neither DAVE nor LCNAF incorporate scoring, so a search on "einstein" doesn't return "Albert Einstein" at or near the top of the results. In the case of LCNAF, "Albert Einstein" is the 24th result. The VIAF AutoSuggest API does provide scoring and the results were significantly better. There were still limitations, however. For example, a search on "emily dickinson" includes not only the poet and relevant societies, but it includes works by Dickinson as well -- something we did not want to include in the results. To provide a prototype of how the autosuggestion could work, we ended up using a large JSON file (over 38 thousand rows) that was populated using a subset of the author and subject facets from the catalog. The labels in these facets match the LCNAF headings, and the facet counts gave us a method for scoring the autosuggest results.
- The Annif implementation currently relies on a training data set with 194,555 entries matching full titles from the catalog with subject heading URIs. This set was retrieved from an initial section of 300,000 Solr records from the production Solr index which include the full title as well as subject heading strings for that record. The possible enhancements section below refers to different methods for retrieving additional catalog metadata and URIs for associated subject headings.
- The related person and subject suggestions load the very first result into the knowledge panel or item detail cards when the page loads. Since the facet text is already available, the facets were included at the top of the list for both person and subject suggestions. The URIs for the facet values are retrieved through an independent AJAX query, which means that it is possible that the knowledge panel viewed may not have access to the URI and thus to the related information coming from the URI. One option we tried was to have the knowledge panel reload once the URI is available for the first result, but that led to some screen jitter and the panel seemed to reload more than once. Currently, the panel seems to load correctly but future work should look at improving the approach for populating the first panel.
- Google highlights a single entity in the knowledge panel (currently on the top right side of the page). In our case, it wasn’t immediately obvious which of the search suggestions for people or subjects were the most relevant or closest to the query. The current SMASH! design displays a list of potentially related suggestions and then populates the knowledge panel for the first while allowing users to click on subsequent items in the list to be able to see their details. Further UI exploration should be done around how best to display that information.

- The suggested person search knowledge panel queries for five contemporaries. The list is ordered by birth date in ascending order. Showing a range including contemporaries covering the range of birth to death and/or start activity dates to end activity dates would be useful. One possibility is to have multiple queries to the author index to retrieve contemporaries at the beginning and end of the time range, or to get a fuller list of contemporaries and show authors from the beginning and end of the range. The question to ask with the latter approach is how to understand the length of the list (i.e. getting this information probably requires a separate call to the index).

Possible enhancements and areas for improvement

- Since it's currently implemented with a JSON file as the data source, the autosuggest field could be re-engineered to drive the query off of a Solr index. The source for the index could again be the author and subject facets from the catalog, only it would include all of the facet headings from the catalog, not just a subset. Also, we could possibly enhance this by including name variants from LCNAF or VIAF.
- The Zero results pages' initial example integration of HathiTrust relied on a static XML file and a subsequent implementation used an IP address-based and specific user account-based access approach. If HathiTrust develops a more general-purpose API that enables retrieving search results as well as subject headings in a structured data format (such as XML), then the code could utilize that API instead to retrieve and display information from HathiTrust.
- Additional work with Annif could generate a more comprehensive training data set using more (or all) records from the catalog. A possible path is to utilize an internal tool for querying the MARC to get all fields that contain subject headings and also retrieve any associated URIs.
    - Currently, the Annif implementation retrieves URIs for the subject headings by querying against id.loc.gov's LCSH data with the subject heading string. Since not all subject headings within the MARC catalog will have URIs, a combination of retrieving any URIs from MARC where they do exist and the approach of querying strings to retrieve URIs could be used to get a more comprehensive set of LCSH URIs associated with a catalog resource.
    - We started with LCSH headings but we could also use the FAST vocabulary and training data that uses the FAST subject headings associated with titles in the catalog.
    - Our set of LCSH data relies on what was available within LD4P2's QA cache for LCSH. Our catalog appears to use some subject headings that are more recent and did not map to anything within the cache (since there have been additional LCSH headings added after the data used in the QA cache). Future work could include
- For the suggested person and subject searches, we tried to incorporate some suggestions from our colleagues at Cornell as well as from Astrid Usong at Stanford. Suggested person and subject searches would benefit from (a) further exploration into which dimensions of relevance would be most useful for users and how to ascertain and evaluate those dimensions and (b) UI design and implementation that supports users in accessing these suggestions. Of note, here are mockups that could be reviewed when designing updates to the UI.