Indexing process details
Overview

- Higher level goals and objectives
- Use cases: Data and UI
- Data processing overview
- Solr index overview: Example document and configuration
- How the index support suggestions
  - Simple examples
  - See also issues and special handling
  - Client-side only vs indexing approach
- Related approach (as demonstrated by Frances Webb)
Higher level goals

- Enable identification and highlighting of related entities
- Typeahead suggestions for persons, locations, subjects, and genres based on user query
- Integrating variant labels, see also and pseudonyms
- Data sources: catalog, LCNAF, FAST, Wikidata, LCGFT
### Authors
- Tunnell, Harry D. (Harry Daniel), 1961- (5)
- Tunnell, Kenneth D. (4)

### Genres
- Tunnel books (2)

### Locations
- Europe > Channel Tunnel (Coquelles, France, and Folkestone, England) (14)
- Massachusetts > Hoosac Tunnel (12)
- Nevada > Sutro Tunnel (5)
- United States > Holland Tunnel (Jersey City, New Jersey and New York, New York) (4)
- West Virginia > Hawks Nest Tunnel (4)
- Colorado > Harold D. Roberts Tunnel (4)

### Subjects
- Tunneling (159)
- Tunnels (153)
  - aka: Highway tunnels
- Wind tunnels (119)
- Tunneling (Physics) (63)
  - aka: Quantum mechanical tunneling
- Scanning tunneling microscopy (56)
- Railroad tunnels (29)
Autosuggest Use Cases

Mark Twain ➔ Pseudonym

See Also

Mark Twain ➔ Samuel Clemens
Mark Twain ➔ Tuwen, Make

Solr Index

<table>
<thead>
<tr>
<th>WD URI</th>
<th>LOC URI</th>
<th>Variants</th>
<th>See Also</th>
<th>Pseudonym</th>
</tr>
</thead>
</table>

See Also

Tuwen, Make
Autosuggest Use Cases

- Motivating questions
  - When should the information result in a match for a query?
  - When should the information enable a separate search in the catalog?
- Use cases outlined here:
  - https://docs.google.com/document/d/1bDJFYXrgaXg3huKwLJgt0WD7TGoUUMKY5lFsn-PYu7U/edit
Use case overview

- If an entity’s preferred label or variant label starts with the query text, then that entity should be displayed as a suggestion. For example, the following queries should match the following labels (separated by commas):
  - “Alb” -> Albert Einstein, Alberta, Ernest Alberto
  - “Eins” -> Albert Einstein
  - “Alb Ein” -> Albert Einstein

- An entity should be displayed as a suggestion only if all the query keywords match. For example
  - “And Ern” -> “Andrew Ernest”
  - But “And Ern Smith” ! -> “Andrew Ernest” (i.e. this should not match)
Use case overview

- An entity may have related headings also present in the catalog. In this case, the related headings should not be shown as separate search results but their information should be displayed as connected to the entity.
- In the case where an entity has related LCNAF headings with distinct URIs (captured with see also relationships), searching for the related labels should result in displaying the main entity as a suggestion.
- For a given heading A in the catalog, if related headings are not present in the catalog (and even if they are present in the authority such as LCNAF), then the text of the related heading (e.g. pseudonym text or see also labels) should show heading A as a suggestion.
Data processing overview

- Authors, subjects, genres, and locations are retrieved from the catalog as JSON.
- These JSON files are processed to generate the Solr documents in the index.
- Vocabulary suggest endpoints are used to resolve the string headings to URIs.
- SPARQL queries are used to retrieve additional information from LCNAF, FAST, and Wikidata.
- A second pass updates the index to handle see also and pseudonym cross-references within the catalog (explained later in the documentation).
Data exported as JSON files

URIs, variants, see also Supplemented with Wikidata

Catalog sources

Author Browse Index

Subject Facet Values

Subject (Region) Facet Values

Genre Facet Values

Authorities/ LD Sources

Author

Faculty

Subject

Location

Genre

LCNAF

FAST

LCGFT

Wikidata

Solr Index

Data exported as JSON files

URIs, variants, see also Supplemented with Wikidata

Catalog sources

Author Browse Index

Subject Facet Values

Subject (Region) Facet Values

Genre Facet Values

Authorities/ LD Sources

Author

FAST

LCGFT

Wikidata

Solr Index

Data exported as JSON files

URIs, variants, see also Supplemented with Wikidata
Twain, Mark, 1835-1910.

Catalog: Author.json

Get URI
1. BAM! Author index
2. If not found, LCNAF

http://id.loc.gov/authorities/names/n79021164

Query Wikidata:
- URI, Description, Pseudonyms

Query LCNAF:
- Variants, See Also

Wikidata Info

LCNAF Info

Generate Solr Document
Index Overview

- Solr’s autosuggest endpoint
  - Separate request handler and URL
- Blacklight built-in autocomplete
  - Expects Solr autosuggest endpoint and data format
- Used separate Solr index
  - Tried out autosuggest endpoint
  - Relying on select handler (i.e. regular search endpoint)
- Field type and reserved field for matching
<table>
<thead>
<tr>
<th>id</th>
<th><a href="http://id.loc.gov(authorities/names/n79104234">http://id.loc.gov(authorities/names/n79104234</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type_s</td>
<td>author</td>
</tr>
<tr>
<td>label_s</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
<tr>
<td>uri_s</td>
<td><a href="http://id.loc.gov/authorities/names/n79104234">http://id.loc.gov/authorities/names/n79104234</a></td>
</tr>
<tr>
<td>variants_t</td>
<td>[&quot;Edems, Dzheyn, 1860-1935&quot;, &quot;Addams, Laura Jane, 1860-1935&quot;]</td>
</tr>
<tr>
<td>rank_i</td>
<td>81</td>
</tr>
<tr>
<td>wd_uri_s</td>
<td><a href="http://www.wikidata.org/entity/Q180989">http://www.wikidata.org/entity/Q180989</a></td>
</tr>
<tr>
<td>wd_description_s</td>
<td>pioneer settlement social worker</td>
</tr>
<tr>
<td>label_t</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
<tr>
<td>id</td>
<td><a href="http://id.loc.gov/authorities/names/n79104234">http://id.loc.gov/authorities/names/n79104234</a></td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>type_s</td>
<td>author</td>
</tr>
<tr>
<td>label_s</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
<tr>
<td>uri_s</td>
<td><a href="http://id.loc.gov/authorities/names/n79104234">http://id.loc.gov/authorities/names/n79104234</a></td>
</tr>
<tr>
<td>rank_i</td>
<td>81</td>
</tr>
<tr>
<td>label_t</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
</tbody>
</table>

Main vocabulary URI stored in “uri_s”. “id” used by Solr to uniquely identify documents based on URI by replace slashes with underscores.
<table>
<thead>
<tr>
<th>id</th>
<th><a href="http://id.loc.govAuthorities/names/n79104234">http://id.loc.govAuthorities/names/n79104234</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>type_s</td>
<td>author</td>
</tr>
<tr>
<td>label_s</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
<tr>
<td>uri_s</td>
<td><a href="http://id.loc.gov/authorities/names/n79104234">http://id.loc.gov/authorities/names/n79104234</a></td>
</tr>
<tr>
<td>rank_i</td>
<td>81</td>
</tr>
<tr>
<td>label_t</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
</tbody>
</table>

Type of entity stored in “**type_s**” field: author, subject, location, and genre. “**rank_i**” stores count from browse index (for authors) and facet values (for subjects, locations, and genres).
Main vocabulary preferred label is saved in the “label_s” field, which is used only for display purposes, and the “label_t” field which is of the type “text_general” and is used in search. Subsequent slides will talk about search in more detail.
<table>
<thead>
<tr>
<th><strong>wd_uri_s</strong></th>
<th><a href="http://www.wikidata.org/entity/Q180989">http://www.wikidata.org/entity/Q180989</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>wd_description_s</strong></td>
<td>pioneer settlement social worker</td>
</tr>
</tbody>
</table>

Where querying Wikidata with the vocabulary URI yields a match, the URI and description of the Wikidata entity are copied over to the Solr document in the fields above.
<table>
<thead>
<tr>
<th>variants_t</th>
<th>[&quot;Edems, Dzheyn, 1860-1935&quot;, &quot;Addams, Laura Jane, 1860-1935&quot;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>label_t</td>
<td>Addams, Jane, 1860-1935</td>
</tr>
</tbody>
</table>

Variant labels are stored in the “variants_t” field. All text fields (i.e. “_t”) fields are copied over to the label_suggest field.
<table>
<thead>
<tr>
<th>variants_t</th>
<th>[&quot;Tvėn, Mark, 1835-1910&quot;]</th>
</tr>
</thead>
<tbody>
<tr>
<td>label_t</td>
<td>Twain, Mark, 1835-1910.</td>
</tr>
<tr>
<td>pseudonyms_t</td>
<td>[&quot;Snodgrass, Quintus Curtius, 1835-1910&quot;, &quot;Conte, Louis de, 1835-1910&quot;]</td>
</tr>
<tr>
<td>wd_pseudonyms_t</td>
<td>[&quot;Sieur Louis de Conte&quot;]</td>
</tr>
</tbody>
</table>

In a separate Solr example, see also labels from LCNAF are saved in the “pseudonyms_t” field and Wikidata pseudonyms are saved in the “wd_pseudonyms_t” field. The contents of all these “_t” fields are copied to the “label_suggest” field.
In addition to matching against queries, we also want to display see also information under a heading that matches. Information for this display is saved in the “pseudonyms_ss” field as a serialized string version of a JSON object. The parsed version of the string is also displayed above to show that the JSON string captures label, uri, and rank information for display.
Search configuration

- Label_suggest is of the type “text_suggest” which allows for the query to be broken into words and to be matched against the beginning of the words in the field.
- The type “text_suggest” was defined added to the Solr configuration as shown here: https://github.com/LD4P/discovery/blob/master/solr_config/wham/suggest/managed-schema.xml#L394
- To enable the autocomplete functionality, we added a search request handler which queries the label_suggest and label_t fields as shown here: https://github.com/LD4P/discovery/blob/master/solr_config/wham/suggest/solr_config.xml#L751
How the index supports suggestions

● As noted, the index is configured to enable matches against preferred labels, variants, pseudonym text, and see also labels stored in the appropriate text fields.
● Data processing takes into account whether see also relationships for an entity are represented by separate headings in the catalog or not.
● The indexing examples are broken into two sets:
  ○ Matching that relies on the data from sources as retrieved.
  ○ Matching that relies on the second indexing pass for see also and pseudonym headings.
How the index supports suggestions: Simple cases

- Preferred label
- Variant labels
- Pseudonym not represented by a separate catalog heading
Query and Results: High level

Query: Little Dorrit

Title: Little Dorrit

Cornell University Library
LIBRARY CATALOG

little dorrit

Limit your search

Looking for more?
- Request from Libraries Worldwide (1,849+)
- Search Articles & Full Text ()
- Recommend a Purchase

Access

1. Little Dorrit
   Book [51]: Project Gutenberg.
   Online

2. Little Dorrit 2009
Query and Results: High level

Query: Jamie

Label: Campell, Jamie
Title: Goode, Jamie
Match against catch-all “bucket” field

Query: Jamie

Label: Campbell, Jamie  
Variants: Saratoga Ceviche  
Pseudonyms: Kambella

*label_suggest:* Campbell, Jamie  
Saratoga, Ceviche  
Kambella

The “bucket” field is the label_suggest field defined in the previous slides explaining index configuration. In this case, “Jamie” matches one of the words in the label_suggest field for the entity Solr document representing Jamie Campbell. The information for the entity, such as preferred label, catalog count, and Wikdiata description (if it exists in the index), is retrieved and displayed as a suggestion.
In this example, “Edems” matches the variant label text added to the “label_suggest” field for the entity Solr document for “Addams, Jane, 1860-1935”. The preferred label is shown in the suggestion, along with “aka” which shows the text the search actually matched on i.e. the variant label.
Pseudonym (without separate catalog heading)

Query: Monica Hill

Pseudonyms: Hill, Monica, 1915-2004


The Solr document for “Watson, Jane Werner, 1915-2004” has “Hill, Monica, 1915-2004” stored as a pseudonym. (The real Solr document shows that this information is coming in from LOC see also connections.) In this case, although LCNAF has a separate URI for Monica Hill, the catalog does NOT have a separate heading. We treat this exactly the same way as we would a variant label.
How the index supports suggestions: Two pass approach

- See also URIs that are separate headings in the catalog
- Once the index is populated with information from the data sources used, a second pass is conducted
  - All Solr documents which have see also relationships are retrieved
  - For each of these see also relationships, the index is checked to see if a Solr document exists for that URI
  - If the URI exists, this means the catalog contains this heading as well. The Solr document is updated in the manner explained in the next few slides.
  - Additionally, if the second heading exists, the Wikidata pseudonym information is also updated by removing any labels that contain the second heading’s label from the Wikidata text that is used for matching.
See also and pseudonym info

- LCNAF See also information is stored in two different types of fields
  - Pseudonyms_ss which is used in the “see also” display in the UX
  - Pseudonyms_t whose contents are used to match
- Wikidata pseudonym text is stored in wd_pseudonym_t
The Twain Dilemma

- Mark Twain and Samuel Clemens
  - Separate LCNAF authorities
  - Both have separate catalog entries
- Desired behavior
  - See also links to catalog entry
  - Mark Twain -> See also Samuel Clemens
    - But no separate Samuel Clemens search result
The Twain Dilemma

Query: Twain

Label: Twain, Mark, 1835-1910
Variants: ...
Pseudonyms_ss: {"uri":..., "label": "Clemens, Samuel Langhorne, 1835-1910", "rank":...}

label_suggest: Twain, Mark, 1835-1910

Label: Clemens, Samuel Langhorne, 1835-1910
Variants: ...
Pseudonyms_ss: {"uri":..., "label": "Twain, Mark,, 1835-1910"}

label_suggest: Clemens, Samuel Langhorne, 1835-1910
The Twain Dilemma

- To enable the “see also” display to show “Samuel Clemens” for the “Mark Twain” suggestion
  - The pseudonym_ss field includes information about the headings that will be displayed
- To prevent the query “Mark Twain” from showing “Samuel Clemens” as a separate independent suggestion
  - The pseudonym_t field for “Mark Twain”’s Solr document does NOT include “Samuel Clemens”
The Snodgrass Conundrum

- If Snodgrass is a see also URI
  - But does not appear as a separate catalog heading
- Desired behavior
  - “Snodgrass” query should bring up Twain
  - “Snodgrass” should be indicated in the UX as what was matched on
  - “Snodgrass” should not show a “see also” reference
The Snodgrass Conundrum

Query: **Snodgrass**

Label: Twain, Mark, 1835-1910

Variants: ...
Pseudonyms_t: [“Snodgrass”]
Pseudonyms_ss: {“uri”:..., “label”: “Clemens, Samuel Langhore, 1835-1910”, “rank”:...}

*label_suggest: Twain, Mark, 1835-1910, Snodgrass, Quintus Curtius, 1835-1910*
The Snodgrass Conundrum

- “Snodgrass” query should bring up Twain
  - Pseudonyms_t in the Mark Twain Solr document contains the name for the Snodgrass heading
- “Snodgrass” should not show a “see also” reference
  - Pseudonyms_ss in the Mark Twain Solr document does NOT contain an entry for Snodgrass
Pseudonym (Data as is approach)

Query: Twain

Label: Twain, Mark, 1835-1910
Variants: ...
Pseudonyms_INFO: {“uri”:..., “label”: “Clemens, Samuel Langhorne, 1835-1910”, “rank”:...}
Pseudonyms: Clemens, Samuel Langhorne, 1835-1910
Bucket: Twain, Mark, 1835-1910

This is what would happen without a second pass over the index:
“Twain” would show Clemens as an independent result as well the see also info

Label: Clemens, Samuel Langhorne, 1835-1910
Variants: ...
Pseudonyms_INFO: {“uri”:..., “label”: “Twain, Mark,, 1835-1910”}
Pseudonyms: Twain, Mark, 1835-1910
Bucket: Clemens, Samuel Langhorne, 1835-1910
Twain, Mark, 1835-1910
Client-side solution vs Indexing solution

- Client-side solution incorporates controller-level parsing/munging
  - Relies on whether see alsos show up as separate solr results (i.e. they exist in the catalog) or not (to be treated as variants and not separate entries)
Client-side solution vs Indexing solution

- Indexing side solution requires different handling
  - Once the index is populated, a second pass checks the see also connections against what is in the index to see if the headings
Related work

- Frances Webb demonstrated a left-anchored autosuggest using the existing Cornell production browse indices.
- For author and subject browses, a query will match against the beginning part of the heading.
- Suggestions are provided using a specific request handler that matches against a field that enables matching against the beginning part of the heading.
Related work

- Results for query “ei”
- URL is “suggest?q=ei a”
- Request handler set up for “suggest”
  - Matches against “heading” field
  - “Heading” is of type "textLeftAnchored" which allows left anchored matching against words in the label
  - “mainEntry” is true if the heading has an LOC authority (i.e. is an authorized heading)