

ELAG 2010

09/06/2010

Autosuggest in VIAF

Ralph LeVan
Thomas Hickey
OCLC Research



OCLC™

The world's libraries.
Connected.

- Spelling of names can be difficult
- Easier searching

- Source of good names during cataloging bib records
- Source of names to avoid when creating name records
- Source of good names when searching
- Hopefully ranking presents best names first, helping to eliminate keystrokes (and potential typos)

- Fast
- Easy to incorporate into search form
- Extensible to other uses
- Minimal added complication to database creation

- Prototyped in flat file
- Ported to SRU environment
 - SRU layer
 - Text retrieval layer
 - Currently Pears
 - Planning a Lucene implementation
- Using Yahoo! UI AutoComplete on pages

Data Flow: Step 1, make tuples



- From each VIAF record we extract tuples of information for each variant of the name
 - Variant form of name
 - Controlled form of name
 - Score for each variant
 - Single JSON string for the term and any other stuff to be sent in the response (LCCN, etc)

Example: Making Tuples



- `<VIAFCluster> <mainHeadings>Dempsey,
Lorcan</mainHeadings>
<ns2:length>500</ns2:length><ns2:viafID>36978042</ns2:
viafID></VIAFCluster >`
- `dempsey, lorcan\t500\t{" term" : " Dempsey,
Lorcan" }\t36978042`
- Score is the length of the record, assuming that longer records indicate more attention from the contributors. Bach's is 126855

Data Flow: Step 2, make AutoSuggester records from tuples



- Find all the variant forms that start with common initial letters
- Eliminate redundant variants that point to the same record
- Pick the 10 remaining tuples with the highest scores
- Write an XML record with the common letters as the key, and the 10 tuples

Example: AutoSuggester Record



```
lorcan d\t[{"term":"Lorcan  
Dempsey", "ptbnp":"70922", "nla":"000035338539", "n  
kc":"js20080511012", "bnf":"12276780", "bav":"adv1  
1117013", "selibr":"256753", "dnb":"114712638", "lc":  
"n90602202"}, {"term":"Lorcan  
Demsey", "nla":"000035786141"}]
```

Data Flow: Step 3, load database



- Load text database with a single index derived from the common first letter keys.

- Need sub 1/10 second to be invisible
- But need to wait for typing pauses
- Yahoo! code does caching
- Using most of Yahoo!'s defaults
 - Expand/collapse animation speed (0.3 seconds)
 - queryDelay (0.2 seconds)

- Tested every 2-letter combination
- 18ms average response time
- fastest (vf)=12, slowest (ad)=111

What happens in query



- AutoSuggester servlet gets request
- Pulls keystroke string from request and turns into SRU query
- Extract pre-built JSON from XML AutoSuggester record
- Return JSON

Request



- <http://viaf.org/AutoSuggest?query=lorcan d>

Result



```
{ "query": "lorcan d",  
  "result": [  
    { "term": "Lorcan Dempsey",  
      "ptbnp": "70922",  
      ...  
      "lc": "n90602202" },  
    { "term": "Lorcan Demsey",  
      "nla": "000035786141" }  
  ] }
```

Integration into form



```
<div class="yui-ac" id="ac-Target">  
  <input size="40" class="yui-ac-input"      id="searchTerms" />  
  <div class="yui-ac-container" id="ACContainer" />  
</div>
```

```
YAHOO.example.VIAFRemote = function() {
    var oDS = new YAHOO.util.XHRDataSource("/viaf/AutoSuggest");
    oDS.responseType = YAHOO.util.XHRDataSource.TYPE_JSON;
    oDS.responseSchema = {resultsList:"result", fields : ["term"]};
    oDS.maxCacheEntries = 20;
    // Instantiate the AutoComplete
    var oAC = new YAHOO.widget.AutoComplete("searchTerms", "ACContainer", oDS);
    oAC.useIFrame = true;    oAC.autoHighlight = false;
    oAC.doBeforeExpandContainer = function(){
        // code to check whether autosuggest is
        // appropriate for selected indexes}
    return {oDS: oDS, oAC: oAC };
};
```

Demonstration



VIAF Beta
Virtual International Authority File

Search

Select Field: Select Index: Search Terms:

Search

VIAF: The Virtual International Authority File

VIAF is a joint project of several national libraries, implemented and hosted by [OCLC](#). The project's goal is to lower the cost and increase the utility of library authority files by matching and linking the authority files of national libraries, and then making that information available on the Web.