

PKP Scholarly Publishing Conference 2013

The new OJS search features with Lucene/Solr

Overview

Why new search features at all?

Goals and Achievements

Decisions to be made

Installation, Configuration and Maintenance

New search features live

OJS search and Solr

OJS search

- across article metadata, full texts and supplementary files
- simple text analysis for indexing and search (white space “tokenization”, conversion of words to lowercase and elimination of “stopwords”)
- doesn’t work for the languages that use logographic notational systems

Solr

- an open source search server using Lucene (indexing and search software library)
- provides more sophisticated tokenizers, token filters and language-specific analysis components “out of the box”
- supports many other advanced search features, e.g. auto-suggestion, alternative spelling proposal, paging, highlighting, ordering, faceting, “more like this” feature, improved ranking, etc.

→ Search optimization part of the project "Functional extensions and value added services for OJS", founded by the Deutsche Forschungsgemeinschaft (DFG)

Goals

Support multilingualism and all OJS languages

Ensure backward compatibility

Benefit from additional search features provided by Lucene/Solr

Consider different deployment scenarios (ranging from single journal installations to large OJS provider deployments):

- search across articles of a single journal
- search across multiple journals of a single OJS installation
- search across various OJS installations
- search across various applications

Design simple, flexible and user friendly

→ Concept: <http://pkp.sfu.ca/wiki/index.php?title=OJSdeSearchConcept>

→ Result: Lucene plug-in for OJS

Decisions to be made

OJS or Lucene/Solr search?

- Are journal languages supported by OJS search?
- Is the use of the additional search features desired?
- Is the support of additional document formats or faster indexing needed?
- Is there the required knowledge and the possibility to maintain another software (Solr)?
- Is PostScript required?

Deployment requirements and possibilities - Embedded or Remote Server Mode?

- one or more independent OJS installations on one server → embedded mode, fully preconfigured and supported by the plug-in and thus easier to use and maintain
- several servers, search across OJS installations or even across different applications → further configuration, implementation and understanding of Solr needed

Push or Pull indexing?

- Push: up-to-date index and simple configuration, works out-of-the-box
 - Pull: server load can be balanced and it is more resilient against Solr server downtime
- Use the simpler push configuration by default but check its performance and reliability

Installation, Configuration and Maintenance

Embedded scenario:

- install Java
- download the appropriate version of Jetty (servlet container Solr runs with) and Solr binaries, extract them, create symbolic links
- check the installation
- secure server
- start Solr server
- enable and setup the preconfigured plug-in
- rebuild indexes

Remote scenario:

- can be installed and deployed in many different ways
- there is no one best deployment

→ See README:

<https://github.com/pkp/ojs/blob/master/plugins/generic/lucene/README>

Solr server settings

Solr server settings

The Lucene plugin accesses the Lucene search index through a Solr server. This configuration page allows you to configure access to the Solr server. **Please make sure you read the plugin's README file (plugins/generic/lucene/README) before you try to change the default configuration.** If you are using the embedded scenario behind a firewall as explained in the README file then you may probably leave all configuration parameters unchanged.

Search Endpoint URL *

The Solr search endpoint consists of the server URL and a search handler. See the default setting as an example. Only change this if you are using a central search server.

Username *

The Solr search server uses HTTP BASIC authentication. Please enter the username.

Password *

Please enter the Solr server password.

Unique Installation ID *

If you use a central search server then you'll have to provide a unique installation ID for every OJS installation sharing the same search index. This can be any arbitrary text but it must be different for every participating OJS server (e.g. the server's static IP address if you have one OJS installation per server).

Search Feature Configuration

Search Feature Configuration

The Lucene plugin provides several optional search features. Most of these features are enabled by default but can be disabled or fine-tuned.

- ☒ **Auto-Suggest (show a dynamic drop-down with search term suggestions while entering a search query)**

Check terms for results

Check terms for results: Only suggest terms that will actually produce search results. Suggestions will be cross-checked against the current journal and terms already entered into other search fields.

Use global dictionary: This is faster, consumes less resources on the search server and therefore scales better for large installations. Suggestions may contain irrelevant terms, though, e.g. from other journals or terms that produce no search results.

- ☒ **Highlighting (display a short excerpt of each article's full text containing queried keywords)**
- ☒ **Faceting (display a navigation box with additional filters to refine your search)**

You may select specific facet categories (the corresponding metadata must have been selected for indexing in journal setup, step 3.4):

Discipline



Keyword



Method/Approach



Coverage



Journal



Author



Publication Date



- ☒ **Alternative Spelling Suggestions (display alternative search terms)**
- ☒ **More-Like-This (display a link "similar documents" for every search result)**
- ☒ **Custom Ranking (set individual ranking weights per journal section)**
- ☒ **Instant search (return search results instantly when a user types a search query - obs: uses considerable server resources)**
- ☐ **Pull indexing (this is an advanced feature, see README file for more information)**

Index Administration

Index Administration

Rebuild index

Rebuild index for all journals



Rebuild index

Rebuild dictionaries

If your Lucene index became outdated or corrupted, you can re-index your data per journal or for all journals of this installation. Dictionaries must be rebuilt after large index updates when using auto-suggest or alternative spelling suggestions. (See [plugins/generic/lucene/README](#) for details and ways to automate these processes.)

Solr Server Administration

Start Server

Custom Ranking Weight

Custom Ranking Weight

The Lucene/Solr search plugin allows you to adjust the relative weight of articles in the result list of a search query. Setting the ranking weight higher (or lower) does not place articles in this section above (or below) all other articles. But they will rank better (or worse) than they would without the adjustment made. Setting this option to "never show" will completely exclude articles in this section from search results.

Normal	▼
Never Show	
Rank Lower	
Normal	
Rank Higher	

Auto-suggestion

Search

Search for

► Additional

lucene
lunch
lunchtime

Order results by:

ISSUE

TITLE

No Results

Search tips:

- Search terms are case-insensitive
- Common words are ignored
- By default articles containing *any* term in the query are returned (i.e., *OR* is implied)
- Make sure that a word exists in an article by prefixing it with +; e.g., *+journal +access scholarly academic*
- Combine multiple words with *AND* to find articles containing all terms; e.g., *education AND research*
- Exclude a word by prefixing it with - or *NOT*; e.g., *online -politics* or *online NOT politics*
- Search for an exact phrase by putting it in quotes; e.g., *"open access publishing"*. Hint: Quoting Chinese or Japanese words will help you to find exact word matches in mixed-language fields, e.g. "图书馆".
- Use parentheses to create more complex queries; e.g., *archive ((journal AND conference) NOT theses)*

REFINE YOUR SEARCH

▶ Discipline

▶ Keyword

▶ Method/Approach

▶ Coverage

▼ Author

- [mcautomatic, arthur](#) (3)
- [author, another](#) (1)
- [author, some](#) (1)
- [authorname, second a](#) (1)

▶ Publication Date

HOME ABOUT USER HOME SEARCH CURRENT ARCHIVES

Home > Search

Search

Search for

▶ Additional Search Options (click to show)

Order results by:

ISSUE	TITLE	
Vol 1	Lucene Test Article 1	ABSTRACT HTML SIMILAR DOCUMENTS
	<i>Some Author, Second A Authorname</i> "... Lucene Test Article 1 Abstract ..."	
Vol 1	Lucene Test Article 2	ABSTRACT PDF SIMILAR DOCUMENTS
	<i>Another Author</i> "... Lucene Test Article 2 Abstract ..."	
Vol 1	Ranking Test Article 3	ABSTRACT SIMILAR DOCUMENTS
	<i>Arthur McAutomatic</i> "... Ranking Test Article 3 abstract ..."	
Vol 1	Ranking Test Article 2	ABSTRACT SIMILAR DOCUMENTS
	<i>Arthur McAutomatic</i> "... Ranking Test Article 2 abstract ..."	
Vol 1	Ranking Test Article 1	ABSTRACT SIMILAR DOCUMENTS
	<i>Arthur McAutomatic</i> "... Ranking Test Article 1 abstract ..."	

1 - 5 of 5 Items

Thank you!

Thanks Florian Grandel for realizing this!

Božana Bokan
bozana.bokan@fu-berlin.de
Freie Universität Berlin
Center for Digital Systems