Indexing Repositories

Pitfalls and Best Practices

Web search and Google Scholar

Web search	Google Scholar
Indexes all documents	Indexes scholarly articles
Needs to document text	Needs to document text but ALSO needs bibliographic information
Indexes each URL independently	Groups all versions of a work together
Has no notion of "an article"	Scholar result corresponds to an entire group

Indexing how-tos

Web search: Webmaster Console

Covers broad range of topics

Provides detailed coverage information

Provides info on crawl errors, server error, breakages, etc.

Google Scholar: Inclusion Help Pages

Linked from homepage

Detailed guidelines

FAQs

What does indexing need?

Web search	Google Scholar
List of all article URLs	List of all article URLs
Ability to fetch article URLs	Ability to fetch article URLs
What we index is what the user sees	What we index is what the user sees
	Identify scholarly articles
	Determine article metadata

Overview

- Pitfalls and best practices
- Measuring index coverage
- Indexing analysis for repository platforms
- Finally...

List of articles - I

Pitfall: Search-only interface

- Treesearch (US Forest service repository)
- BCIN (Conservation Information Network)
- No way to list all articles
- What Scholar system doesn't know about, it cannot index

List of articles - II

Pitfall: List-based browse (click "Next")

- Web scale crawlers are designed for volume
- Crawl all sites in parallel, per-site doesn't scale
- Batches of URLs, each batched assigned X hours
- One "Next" is scheduled in each batch
- 25 articles per "Next" => 100s of "Next"s
- DSpace/Fedora default browse

List of articles - III

Pitfall: Hard to find recent additions

- Example: browse only for individual collections
- Collections structure mirrors org structure
- No date sort or recent additions list
- Some DSpace/Fedora instances skip "By Date"

List of articles - IV

Best practice: Year-month browse

- Linked from homepage - EPrints
- Helps crawlers as well as users

Best practice: Article sitemap

- Include urls for ALL articles
- Linked from robots.txt or homepage
- DSpace if sitemaps are enabled

Fetch articles - I

Pitfall: AJAX used to fetch article text

- AGRIS (FAO, fixed), OSTI (US Dept. of Energy, fixed), EuDML (European Math Library, fixed)
- Security issues limit execution within indexer
- Article text not seen by indexer
- AJAX for main content doesn't help UI either
- User needs to wait either way

Fetch articles - II

Pitfall: Full text hosted elsewhere

- Articles elsewhere not part of repository
- If indexed, provide visibility to hosting site, not repository
- URLs may or may not be available to crawlers
- Remote site may be roboted of restricted
- Embedded metadata can be associated only with on-site full text (Google Scholar)

Fetch articles - III

Best practice: Include text directly on the page

- Avoid Javascript for fetching indexable text
- Javascript better for user interaction or auxiliary features (stats, related articles, etc...)
- For main content, need to wait either way

What we index is what you see - I

Pitfall: Interstitial when clicking on full text

- Terms of use, registration
- Users expect to see article
- If shown other pages, click back immediately
- Learn to avoid clicking on repository in the future
- Seen as cloaking and removed from web search

What we index is what you see - II

Pitfall: Redirect PDF to landing page

- Possibly to help with usage analytics
- Users clicking on PDF links are looking for full text
- If no PDF, they click back, learn to stay away
- Seen as cloaking and are removed by web search

What we index is what you see - III

Best practice: Skip interstitials for users clicking on search results

- One-time terms-of-use unfortunately doesn't work either
- Search users see few articles from repository

Best practice: PDF URLs get full-text PDF document

- For analytics, server API can replace Javascript

Scholar-specific guidelines

Scholar indexes scholarly articles, books, reports, theses, etc...

- Need to identify bibliographic information
- Title, authors, where/how published, when
- Need to determine if in-scope for Scholar

Is it scholarly - I

Pitfall: No machine-readable metadata

- Need article metadata for determination
- Automated analysis of HTML/PDF, formats vary
- HTML with CSS is, ahem, versatile
- Analysis of scanned articles depends on OCR
- Machine-readable metadata via metatags

Is it scholarly - II

Best practice: Embed machine-readable metadata as metatags on record landing page

- We recommend HighWire Press metatags ("citation_XX")
- Provide sufficient detail for scholarly articles
- Structured fields for journal name/volume/issue/pages/year
- citation_pdf_url to associate data with PDF full text
- Dublin Core as last resort (key fields missing)

Article metadata - I

Pitfall: Drop authors from other institutions

- Usually caused by interaction with CRIS
- CRISs tend to focus on local authors

Pitfall: Reorder author list

- Often due to treating authors as a set, not a list

Article metadata - II

Pitfall: Include all contributors as authors

- Advisors, thesis committees common case

Article metadata - III

Pitfall: Use upload date as publication date

- Often via bulk uploads (no date specified)
- "Some date is better than no date..."
- Missing data can be inferred from elsewhere
- Wrong data is much harder to override
- Scholar tries to auto-identify problem sites
- Drops sites with large number of broken dates

Article metadata - IV

Pitfall: Add cover pages to full-text PDF

- Usually branding, download timestamp, etc.
- Often breaks automated metadata extraction
- Article titles don't usually appear on 2nd/3rd page
- have seen up to three leading pages inserted into PDF
- Can result in a systematic drop in coverage

Article metadata - V

Best practice: Use author list as in article

- Other versions are not suitable for repository
- Local-authors: suitable only for CRIS context
- Only authors are "authors", others are acknowledged

Best practice: No default publication dates

- Publication date is either specified or empty
- Add separate field for upload date

Article metadata - VI

Best practice: Host PDF articles "as is"

- Avoid cover pages
- Full-text articles match many more queries
- Systematic drop of full text has huge impact on visibility

Measuring coverage - I

Pitfall: Using result count for site: queries

- Does NOT work for any web search service
- Result count is a broad approximation
- Intended to help with query formulation
- Version grouping in Scholar is another issue
- site: on scholar applies to main links
- Doesn't cover "all versions"

Measuring coverage - II

Pitfall: Using result count of filetype: queries

- Counts for all queries broad approximations
- Filetype: queries not suitable for Scholar
- Scholar groups all versions
- Individual versions not returned as results
- Not possible to limit to particular version type

Measuring coverage - III

Best practice: Random sampling

- Pick a small, random sample of article titles
- Use intitle:"<TITLE>" as the query
- Web search: check matching results
- Scholar: also check "all XX versions" link in search results on page

Analysis of repository platforms

Indexing features

- Article list, fetching articles, identifying scholarly articles, article metadata

Platforms

- EPrints, DSpace, Digital Commons

Finally...

A few key features enable indexing

- Repositories with these features are well indexed

Indexing features should be on by default

- All repositories want to be well-indexed

Shared goal: make it easy to find research

- Contact us if you run into issues
- We'd love to help identify/fix problems

Thank you!