Indexing Repositories: Pitfalls & Best Practices

Anurag Acharya

Web search & Scholar

- Web search indexes all documents – Scholar indexes scholarly articles
- Web search needs document text – Scholar also needs bibliographic info

- Web search indexes each url independently – Scholar groups all versions of a work
  – Scholar result corresponds to entire group

Indexing how-tos

- ♣ Web search: webmaster console – Covers broad range of topics
  – Provides detailed coverage information – Crawl errors, server errors, breakages, etc
- Scholar: inclusion help pages
  - Linked from homepage – Detailed guidelines, FAQs

What does indexing need?

- List of all article urls
- Ability to fetch article urls
- What we index is what the user sees

Web search Scholar

- Identify scholarly articles
- Determine article metadata
Overview

- Pitfalls and best practices
- Measuring index coverage
- Indexing analysis for repository platforms
- Recommendations 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 we don’t know about, we can’t 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 batch 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
  – Eg: 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 –
Fetch articles - I

❖ Pitfall: AJAX used to fetch article text – AGRIS (FAO), OSTI (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: Fetching fulltext requires POST
– Eg: POST for download button
– Possible reason: tracking downloads
– Dynamic urls with GET are just as easy to track – POST forms mostly used for update ops – Update account, upload article, delete info etc – Crawlers skip POST to avoid causing updates

Fetch articles - III
Pitfall: Splitting theses into chapters – Theses are large, can take a while to download – Few years ago, network speeds were slower –
Less of an issue these days
– Indexer can’t know how to put pieces together – Individual chapters aren’t citable
– Theses available as chapters indexed only in web search, not indexed in Scholar

Fetch articles - IV

• Pitfall: Fulltext 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 or restricted
– Embedded metadata can be associated only with on-site fulltext (Scholar)

Fetch articles - IV

⭐ Best practice: Include text directly on 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

❖ Best practice: HTTP GET for article text – Reserve POST for repository updates

Fetch articles - V

❖ Best practice: Include full thesis versions – Mark the full version (Scholar)
❖ Best practice: Host fulltext locally – Maximize visibility of
repository
– Ensure availability to crawlers
– Ensure association of metadata with fulltext

What we index is what you see

❖ Pitfall: Interstitial when clicking on fulltext – Terms of use, registration
– Users expect to see article
– If shown other pages, click back immediately – Learn to avoid clicking on repository in future – Seen as cloaking and removed by web search
What we index is what you see

- ♦ Pitfall: Redirect PDF to landing page
  - Possibly to help with usage analytics
  - Users clicking on PDF links are looking for fulltext
  - If no PDF, they click back, learn to stay away – Seen as cloaking and removed by web search

What we index is what you see
✧ Best practice: Skip interstitials for users clicking on search results

– One-time terms-of-use doesn’t work either – Search users see few articles from a repository

✧ Best practice: PDF urls get fulltext PDF – 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 – PURE, Islandora, VTLS, Treesearch
Is it scholarly - II

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

– We recommend Highwire Press metatags – Provide sufficient detail for scholarly articles – Structured fields for jrnл/vol/iss/pages/year – citation_pdf_url to associate with PDF fulltext – Dublin Core as last resort (key fields missing)

-------------------------------------------------------------------------------

Article metadata - I

• ♣ Pitfall: Drop authors from other institutions – Usually caused
by interaction with CRIS – CRIS’s tend to focus on local authors

▪️ Pitfall: Reorder author list
  – Often due to treating authors as a set, not list

▪️ Pitfall: Include all contributors as authors – Advisors, thesis committees common case

----------------------------------------

Article metadata - II

▪️ 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 - III

• ♣ Pitfall: Add cover pages to fulltext PDF – Usually branding, download timestamp etc – Often breaks automated metadata extraction – Article titles don’t usually appear on 2nd/3rd pg – Have seen up to three leading pages inserted – Can result in systematic drop in coverage

______________________________________________
Best practice: Use author list as in article – Other versions not suitable for repository – Local-authors: suitable only in CRIS context – Only authors are “authors”, others are ack’ed

Best practice: No default publication dates – Publication date is either specified or empty – Add separate field for upload date

Best practice: Host PDF articles as-is – Avoid cover pages
– Fulltext articles match many more queries
– Systematic drop of fulltext has huge impact on visibility

Measuring coverage

♣ Pitfall: Using result count for site: queries – Does NOT work in any web search service – Result count is an broad approximation – Intended to help with query formulation – Version grouping in Scholar 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” page

Analysis of repository platforms

❖ Indexing features
– Article list, fetching articles, identifying scholarly articles, article metadata

❖ Platforms
– EPrints, DSpace, Digital Commons, PURE
EPrints

- ♠ Indexing features: zero config since 2007 – Almost all instances have indexing features
- ♠ List all articles: year-month browse
- ♠ Machine-readable metadata as metatags
  – Metadata model handles articles & theses
- ♠ EPrints repositories well-indexed

DSpace
- ♣ Indexing features: require configuration – Highwire press metatags default since 1.7

- ♣ List of articles: “Next” clicks by default ♣ ♣ Metadata model is general

- Journal article details require customization

- ♣ Instances with recent release well-indexed – Large new repositories can take a while

_____________________________________________________

Digital Commons
▪️ Indexing features: some configuration ▪️ List of articles: by collection

– Recent additions by default, no sitemap
▪️ Machine-readable metadata as metatags

– Metadata model handles articles & theses ▪️ DC repositories often well-indexed

– Large new repositories can take a while

_______________________________________________________________

PURE
• Indexing features: require custom upgrade
• List of articles: no crawl-friendly browse
  – No sitemap
• No machine-readable metadata by default
• Limited coverage for PURE-only repositories
  – Some sites use PURE for CRIS + a repository

------------------------------------------

Recommendations for platforms
• Indexing features that just work
– No configuration needed to enable
– Features wanted by almost all repositories – Blocking indexing is easy via robots.txt

User-agent: *
Disallow: /

– Auto-enable huge success for OJS!

Recommendations for platforms -II

- ♣ Comprehensive & efficient browse – Year-month browse linked from homepage – OR sitemap linked from robots.txt – Timely indexing of large repositories – Rapid pick up of new additions
Recommendations for platforms - III

- ✿ Embed machine-readable metadata
  - Decouple UI from content
  - Customize HTML pages without losing coverage
  - Use citation_pdf_url to associate metadata with fulltext

 Separate upload date & publication date – No default publication date

Recommendations for platforms - IV

 Author lists exactly as in the article itself – Separate CRIS and repository features – Separate fields for non-author contributors
Server-side analytics API support – Enables analytics for non-HTML items

Recommendations for platforms - V

Automated analysis to help identify metadata problems

– Too many articles with same publication date

– Too many PDFs with sparse covers

– Too many titles with common prefix/suffix • “Analysis of Magic Rites – University of X”
Author names with known affiliation keywords • “John Doe, University of Y”

Finally...

♣ A few key features enable indexing – Repositories with these features indexed well

♣ 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
   – Would love to help identify/fix problems