Supporting Subject Librarians with AI Solutions

Osma Suominen
IFLA Subject Analysis and Access WG on Automated Indexing Webinar

9 November 2022
Osma Suominen
Information Systems Specialist, National Library of Finland

Doctoral thesis “Methods for Building Semantic Portals”
Semantic Computing Research Group, Aalto University, 2013
Supervisor Professor Eero Hyvönen

Joined the National Library in 2013
to set up the Finto.fi thesaurus and ontology service

Team leader for automated cataloguing project (since 2019)
Working on automated subject indexing (Annif, Finto AI)

Open source software projects e.g.:
- Skosify - Validation and QA tool for SKOS vocabularies
- Skosmos - SKOS vocabulary publishing tool
- Annif - Tool for automated subject indexing and classification

Twitter: @OsmaSuominen
LinkedIn: osmasuominen
GitHub: @osma
General purpose open source tool for automated subject indexing and classification

Multilingual, supports many vocabularies

Code on GitHub, website with test form and API

Global development and user community; user forum annif-users on Google Groups

Developed since 2017

Annif.org

Automated subject indexing service for production use, based on Annif

Supports indexing with the General Finnish Ontology YSO in Finnish, Swedish and English language

Web user interface and API service

Intended to support subject cataloguers in Finland regardless of institution (GLAMs, public administration); sister project to the Finto vocabulary service

Launched in 2020

Ai.finto.fi
Outline

1. Preparing the ground for AI solutions
2. Algorithms and data sets
3. Interfacing between developers and librarians
4. Putting AI into production
1. Preparing the ground for AI solutions
Setting expectations, communicating goals

What are you aiming for?

- improvement of subject cataloguing processes?
- indexing of large amounts of documents that humans can’t handle?
- replacement of subject cataloguers by machines?

These are all different goals that you need to communicate to everyone involved.
# Humans vs. algorithms in subject cataloguing

## Humans

- Have background knowledge about the world
- Remember what they've done in similar situations
- Memorize core parts of the vocabulary
- Are creative
- Understand bias and try to avoid it
- Are slow
- Are inconsistent
- Make (human) mistakes

## Algorithms

- May be trained on millions of examples
- See patterns in data that humans miss
- Know all of the vocabulary, but in a shallow way
- Are fast and tireless
- Are easily biased
- Struggle with change
- Don’t really understand what they’re doing
- Make mistakes that don’t make any sense
**Machine-assisted vs. fully automated subject indexing**

**Machine assisted (semi-automatic)**

- Beginner friendly (e.g. student indexing thesis)
- More consistent indexing
- Possibly faster than without assistance

Users like it, but is it actually better?
Can we measure it?

**Fully automated**

- Collections that can’t be indexed manually
- Crucial to set expectations accordingly
- Quality not be as good as professional indexing - but maybe better than non-expert?
Buy or Build?

**Commercial solution**

**Pros:** Apply existing, mature products
  - Access to expertise not available in-house
  - Clear responsibilities: provider & customer

**Cons:** One size fits all solutions
  - Lack of options (e.g. language, vocabulary)
  - Vendor lock-in

**DIY open source solution**

**Pros:** Build solutions based on actual needs
  - Competency building for own staff
  - Community building & sharing

**Cons:** Requires dedicated staff
  - Requires in-house expertise
  - Sustainability?
YSA
YSO
Allärs
KOKO
black box
Required resources

For a successful automated subject indexing project, you will need:

1. a well defined subject vocabulary or classification
2. enough good quality training and evaluation data
3. staff with necessary skills [next slide]
4. computing resources (from laptops sometimes up to big servers)
Required staff skills

Collectively, your team should:

- know the subject vocabulary and how it’s used
- be familiar with subject cataloguing practices processes
- be able to work with data sets, e.g. database dumps of text corpora
- be familiar with the tools for automated indexing
- understand evaluation metrics & methodology
- be able to operate production web services
- **talk to each other & people affected by automation**
Annif tutorial
Hands-on guide - arranged 5 times in 2021

Videos and exercises freely available on YouTube & GitHub!
2. Algorithms and data sets
Classification vs. subject indexing

**Classification**

Goal: Pick the **one correct class** among many defined classes that best fits this document

E.g. DDC, UDC, fields of science classifications

In machine learning: **multiclass** classification

**Subject indexing**

Goal: Pick **a few (3-12) concepts** from a subject vocabulary (subject headings or thesaurus) that best describe the topic of this document

E.g. LCSH, MeSH, AGROVOC

In machine learning: **multilabel** classification; with big vocabularies and messy, real world data sets

→ **extreme multiclass classification (XMC)**
Lexical vs. associative algorithms for subject indexing

**Lexical** approaches (e.g.: MLLM, stwfsa)

match the **terms** in a document to **terms** in a controlled vocabulary

"**Renewable resources** are a part of Earth’s **natural** environment and the largest components of its ecosphere."

[Image: Word cloud with terms like "renewable," "natural," "resources," etc.]

Lexical approaches need comparatively little training data. Best suitable for multilabel subject indexing.

**Associative** approaches (e.g.: SVC, fastText, Omikuji)

learn which **concepts** are correlated with which **terms** in documents, based on training data

Associative approaches need a lot more training data in order to cover each subject. Both for multiclass and multilabel classification.
Algorithms may be used **alone**, or in combinations, **ensembles**

**Ensembles are nearly always better** than individual algorithms
Make sure to have enough training and evaluation data

Collect already indexed documents, or metadata about documents, from

- bibliographic catalogues
- discovery systems
- institutional repositories
- digital archives

Ideally you should have

- for lexical algorithms: 1000 or more indexed documents (or abstracts)
- for associative algorithms: \((10 \times \text{size of vocabulary})\) documents (or records)
Text: title, abstract, keywords, fulltext...

Text is the **main, often only**, information fed into automated subject classification algorithms. It is important to have enough good quality text that represents the topic.

<table>
<thead>
<tr>
<th>Title</th>
<th>Often too short to capture the whole topic; can be figurative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title + keywords</strong></td>
<td>Better than title alone, even if keywords are uncontrolled</td>
</tr>
<tr>
<td><strong>Abstract</strong></td>
<td>Very good summary of the document</td>
</tr>
<tr>
<td><strong>Fulltext</strong></td>
<td>Good but may be noisy. Extracting text from PDFs or OCR processes can produce garbage. Often enough to use just the beginning (e.g. first 5000 characters)</td>
</tr>
</tbody>
</table>
Biases, omissions, quality errors

Many quality issues to watch out for:

- too few documents in a collection; skewed towards some topic areas
- existing subject indexing is inconsistent or has many errors
- few or no documents about emerging topics
- only 0-2 documents about many concepts in the vocabulary (long tail)

Some algorithms are more sensitive to these problems than others. **Extreme classification** algorithms (e.g. Omikuji) are better than others.
3. Interfacing between developers and librarians
Workshops

We’ve arranged workshops at the biennial Library Network Days (2017, 2019, 2021) where participants performed subject indexing and/or rated suggested subjects for example documents. The subjects were produced either by human indexers or Annif algorithms.

The workshops have been very successful in spreading awareness about automated subject indexing among Finnish librarians.

2019 workshop. Photo: Mikko Lappalainen.
User testing of AI tools & services

Can be approached from many angles:

1. usability testing of user-facing tools (e.g. screen recording, think aloud protocol)
2. subject librarians make notes during their daily work
3. asking for user feedback via survey forms

We've done a little bit of 1., some more of 2. and 3.
Agile practices: librarians as users

Software & systems development is nowadays often done using agile methods.

Subject librarians can be active users in the process, for example:

- testing prototypes and intermediate versions
- suggesting and prioritizing features
- evaluating results of algorithms
Evaluation approaches (Golub et al. 2016), emphasis mine

1. Evaluating indexing quality directly through assessment by an evaluator or by comparison with a gold standard.

2. Evaluating indexing quality directly in the context of an indexing workflow.


The different evaluation approaches are complementary. Not a good idea to look at just a single measure.

4. Putting AI into production
Deep vs. lightweight integration

**Deep integration**: automated topic suggestions in the cataloguing user interface

**Lightweight integration**: separate web UI, copy & paste strings in the correct format

JYX Dspace repository using Finto AI API service

Copy Finto AI suggestions in Aleph ILS format
Technical infrastructure for production use

You can start with laptops, but production use needs servers!

Development server
48 CPU cores, 512GB RAM

Local Linux install
used for training and evaluating models

Container infrastructure

deploy (copy) models to production

trained models

trained models

api.annif.org
Start by experimentation, move slowly towards production

image credit: @kettudolls (IG)
Thank you!


*JlIs.It*, 13(1), 265–282. [https://doi.org/10.4403/jlis.it-12740](https://doi.org/10.4403/jlis.it-12740)

These slides: [https://tinyurl.com/ifla-supporting-librarians](https://tinyurl.com/ifla-supporting-librarians)