

Annif

leveraging bibliographic metadata for
automated subject indexing and classification

Osma Suominen

SWIB18, Bonn, 28 November 2018

Extrablad till ÅBO UNDERRÄTTELSE

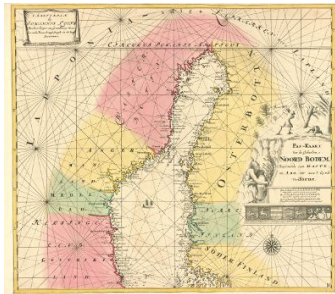
No 2.
Åbo den 10 Mars 1807.

Finlands oavhängighet.

Landtagen omfattar regeringens proklamation om Finlands fullständiga oavhängighet och anseer sig till hädanefter i regeringens program för trygghet av landets nya ställning. Sederst följande med 100 riksdaler 58 skilling till föllo ett av socialdemokraternas förbundet förlag.

Återupplagandet av trykningen Åbo-Stockholm.

Pris 25 penn.



JANNE TILGA
© 2016

Framework for Open Science and Research 1.0.2016 8.196

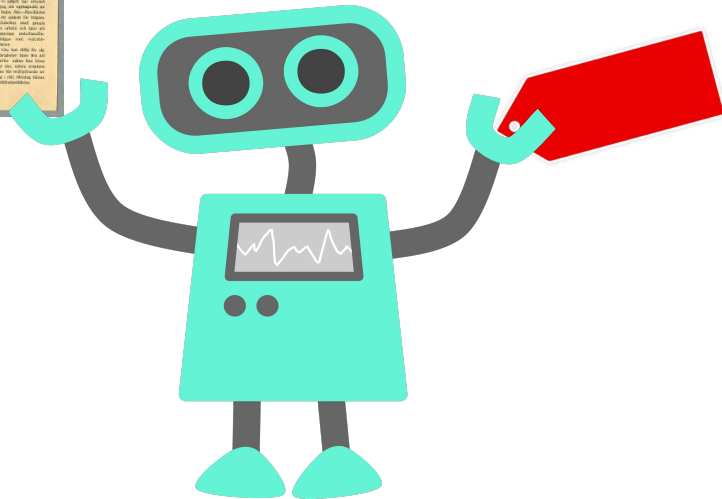
For example, which physical infrastructure will store the digital data resources, who will be responsible for maintaining the infrastructure, and how will the infrastructure be used?

The Karsten EA method used in the higher education sector (described below) has been explained in this work.

Open Level - WHAT	Activities	Information and Knowledge	Applications	Technical infrastructure and software
Level 1 - WHAT	Open Access	Open Access	Open Access	Open Access
Level 2 - HOW	Open Access	Open Access	Open Access	Open Access
Level 3 - WITH WHAT	Open Access	Open Access	Open Access	Open Access

In accordance with the Karsten EA method, unlicensed work was conducted by the Karsten EA description model for work in the physical infrastructure for open science. An example of the unlicensed work is provided in the right column. In the diagram, the focus has been on the activities that describe the objectives of the digital open science, in accordance with the Karsten model. The following diagram roughly models the sub-description of the unlicensed work conducted in activities.





Idea of Annif



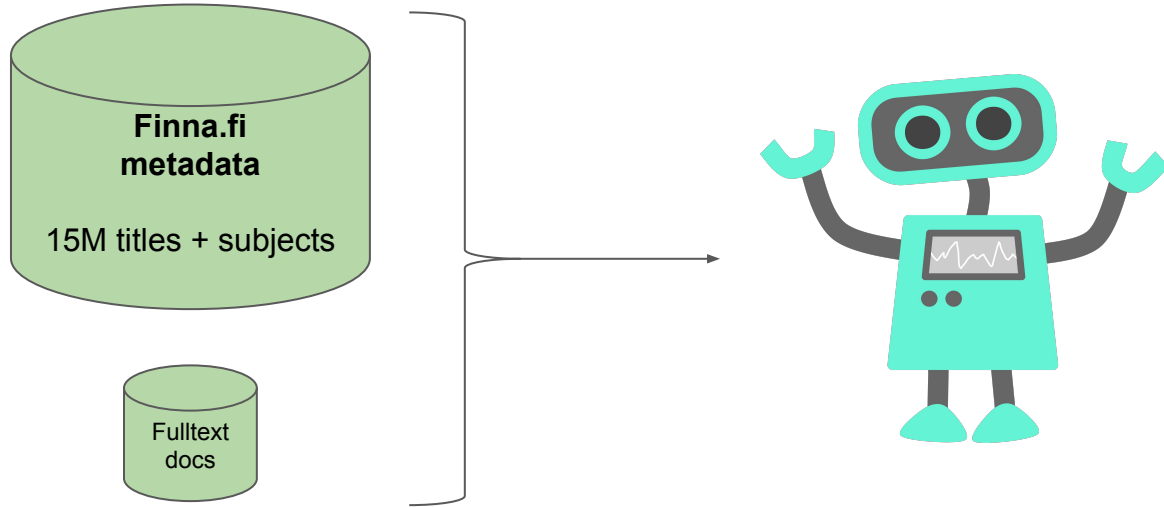
The collections of Finnish archives, libraries
and museums at your fingertips.

All fields ▾

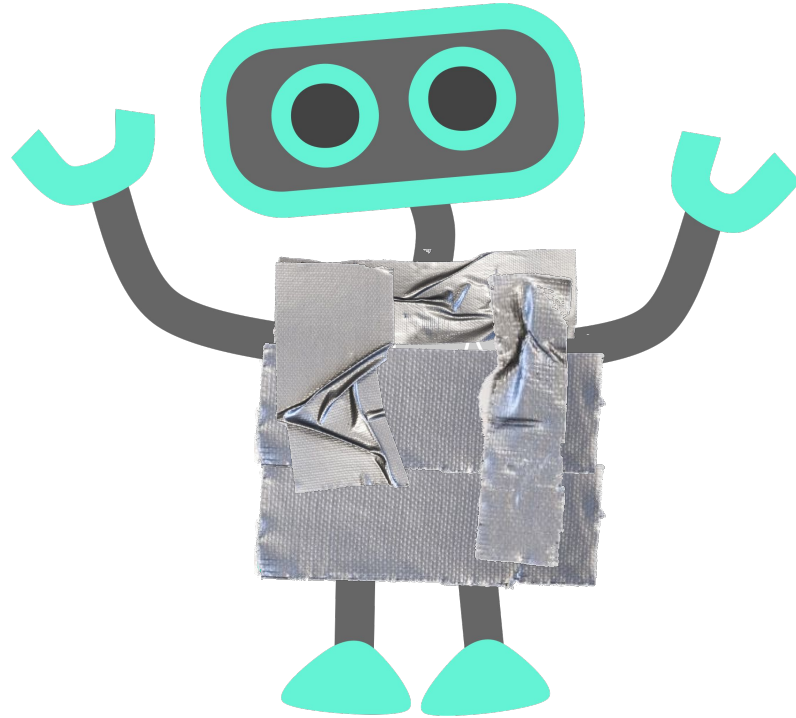
 Advanced search

We have **a lot** of LAM metadata, e.g. **15M records** in [Finna.fi](https://finna.fi) discovery service

Machine learning using library data



Annif prototype (2017)



Annif prototype vs. new Annif

	Prototype (2017)	New Annif (2018→)
<i>architecture</i>	loose collection of scripts	Flask web application
<i>coding style</i>	quick and dirty	solid software engineering
<i>backends</i>	Elasticsearch index	TF-IDF, fastText, Maui ...
<i>language support</i>	Finnish, Swedish, English	any language supported by NLTK
<i>vocabulary support</i>	YSO, GACS ...	YSO, YKL, others coming
<i>REST API</i>	minimal	extended (e.g. list projects)
<i>user interface</i>	web form for testing	http://dev.annif.org
<i>mobile app</i>	HTML/CSS/JS based	native Android app
<i>open source license</i>	CC0	Apache License 2.0

Algorithms for automated subject indexing

Lexical vs. Associative approaches for subject indexing

Lexical approaches

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.”

yso:p14146
“renewable natural resources”

Associative approaches

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



For more information, see:

Toepfer, M., & Seifert, C. (2018). **Fusion architectures for automatic subject indexing under concept drift: Analysis and empirical results on short texts**. *International Journal on Digital Libraries*. DOI: [10.1007/s00799-018-0240-3](https://doi.org/10.1007/s00799-018-0240-3)

Algorithms used in Annif

Statistical / Associative

- **TF-IDF similarity**

Baseline bag-of-words similarity measure. Implemented with the [Gensim](#) library.

- **[fastText](#)** by Facebook Research

Machine learning algorithm for text classification.

Uses word embeddings (similar to [word2vec](#)) and resembles a neural network architecture.

Promises to be good for e.g. library classifications (DDC, UDC, YKL...)

Lexical

- **Maui** using MauiService REST API

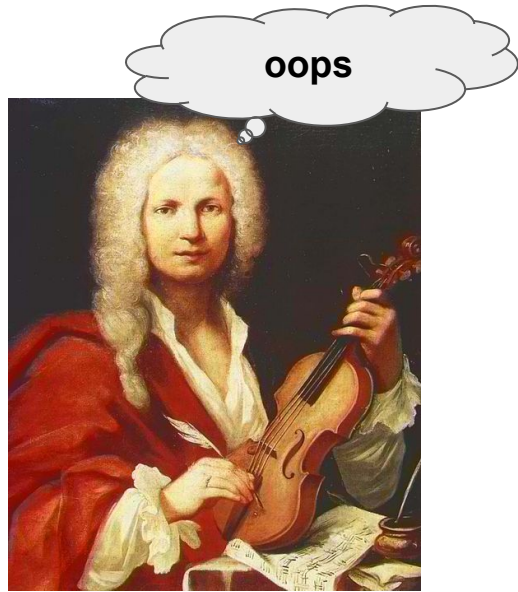
[MauiService](#) is a microservice wrapper around the [Maui](#) automated indexing tool.

Based on traditional Natural Language Processing techniques - finds terms within text.

Algorithms may be used **alone**, or in combinations, **ensembles**



Algorithms make silly mistakes



Some reasons for mistakes:

- errors and skew in training data
- correlation \neq causation
- homonyms (e.g. rock)
- misinterpreted names (e.g. Smith, AIDS)
- random noise

In an ensemble, each algorithm makes different mistakes



Solution: If we have some more training documents, we can perform **second order learning!**

Isotonic regression, implemented using the Pool Adjacent Violators (**PAV**) algorithm, is a good way of assessing trustworthiness of individual algorithms and turning raw scores into final probability estimates.

Wilbur, W. J., & Kim, W. (2014). [Stochastic Gradient Descent and the Prediction of MeSH for PubMed Records](#). *AMIA Annual Symposium proceedings. AMIA Symposium, 2014*, 1198-207.

[Annif Fusion experiment](#) demonstrates PAV

Evaluation of algorithms

Test corpora for evaluating algorithms

Full text documents indexed with YSA/YSO for training and evaluation

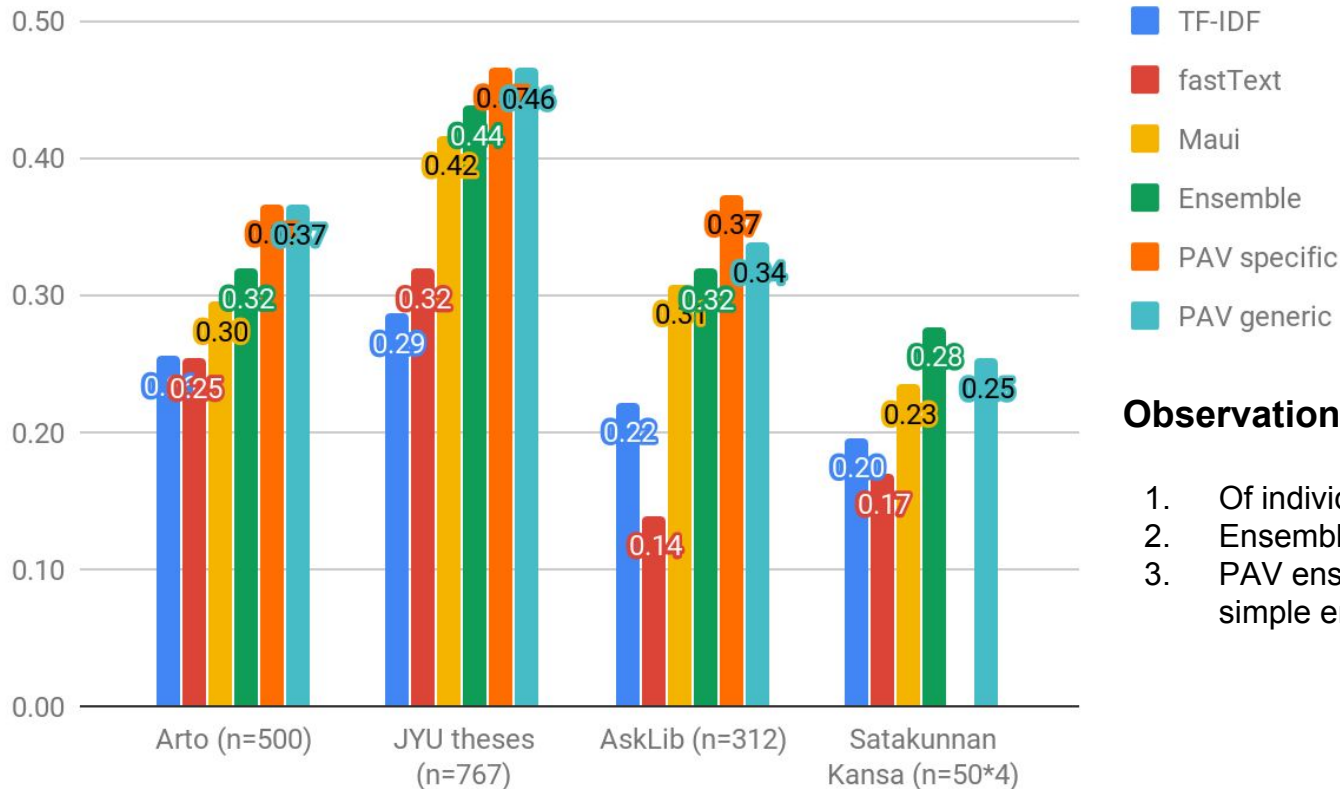
1. **Arto:** Articles from Arto database (n=6287)
Both scientific research papers and less formal publications. Many disciplines.
2. **JYU theses:** Master's and Doctoral theses from University of Jyväskylä (n=7400)
Long, in-depth scientific documents. Many disciplines.
3. **AskLib:** Question/Answer pairs from an Ask a Librarian service (n=3150)
Short, informal questions and answers about many different topics.
4. **Satakunnan Kansa:** Digital archives of Satakunnan Kansa regional newspaper.
Over 100k documents, of which 50 have been indexed independently by 4 librarians.

Corpora 1-3 available on GitHub: <https://github.com/NatLibFi/Annif-corpora>

(for 1-2, only links to PDFs are provided for copyright reasons)

Evaluation of different algorithms in Annif

F1 scores (combination of precision & recall) against gold standard subjects

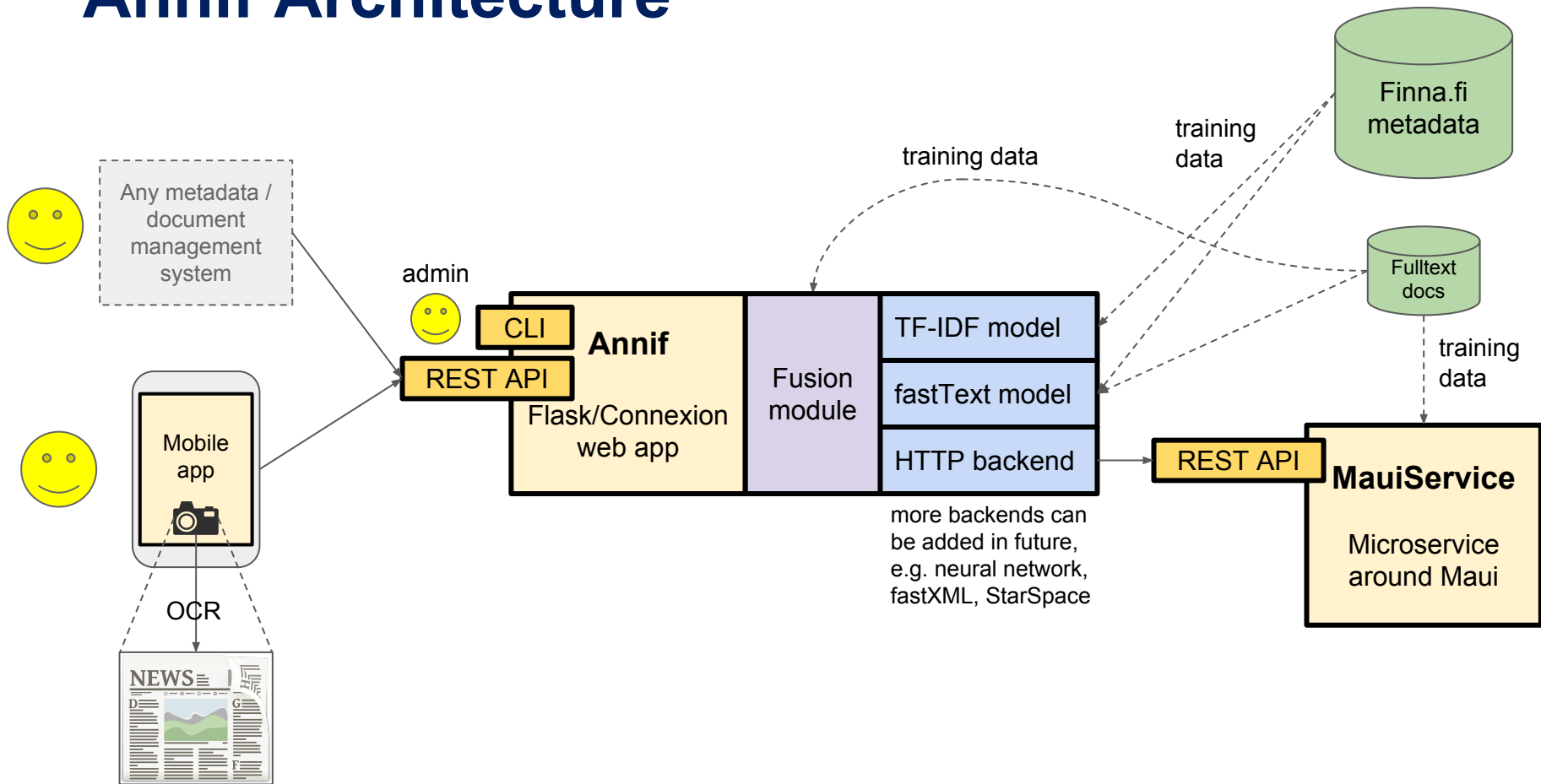


Observations:

1. Of individual algorithms, Maui is the best
2. Ensembles beat individual algorithms
3. PAV ensembles can be better than a simple ensemble (but not always)

Software architecture

Annif Architecture



Form for testing at annif.org

Try Annif!

Text to analyze:

SWIB conference (Semantic Web in Libraries) is an annual conference, being held for the 10th time, focusing on Linked Open Data (LOD) in libraries and related organizations. It is well established as an event where IT staff, developers, librarians, and researchers from all over the world meet and mingle and learn from each other. The topics of talks and workshops at SWIB revolve around opening data, linking data and creating tools and software for LOD production scenarios. These areas of focus are supplemented by presentations of research projects in applied sciences, industry applications and LOD activities in other areas.

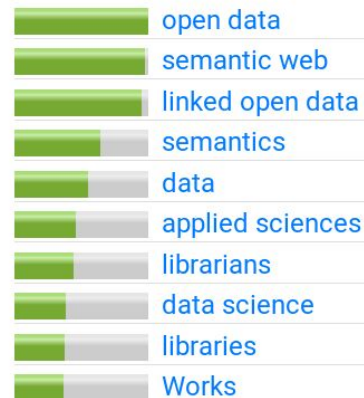
YSO model
trained on Finna data

Project (vocabulary and language):

YSO ensemble English

Analyze

Results



Form for testing at annif.org

Try Annif!

Text to analyze:

SWIB conference (Semantic Web in Libraries) is an annual conference, being held for the 10th time, focusing on Linked Open Data (LOD) in libraries and related organizations. It is well established as an event where IT staff, developers, librarians, and researchers from all over the world meet and mingle and learn from each other. The topics of talks and workshops at SWIB revolve around opening data, linking data and creating tools and software for LOD production scenarios. These areas of focus are supplemented by presentations of research projects in applied sciences, industry applications and LOD activities in other areas.

Wikidata model trained on Wikipedia

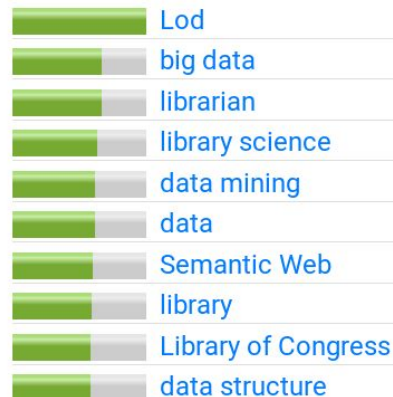
Top 50k items by # of sitelinks

Project (vocabulary and language):

Wikidata TF-IDF English

Analyze

Results



Command line interface

Load a vocabulary to be used by one or more models:

```
$ annif loadvoc tfidf-en yso-en.tsv
```

Train a model:

```
$ annif train tfidf-en yso-finna-en.tsv.gz
```

Analyze a document:

```
$ annif analyze tfidf-en <berries.txt>
<http://www.yso.fi/onto/yso/p772>    strawberry          0.39644203283656165
<http://www.yso.fi/onto/yso/p18109> wild strawberry     0.37539359094384245
<http://www.yso.fi/onto/yso/p25548> stolons            0.3261554545369906
<http://www.yso.fi/onto/yso/p6749> berry cultivation  0.2394291077460799
<http://www.yso.fi/onto/yso/p10631> questionnaire survey 0.22714475653823335
<http://www.yso.fi/onto/yso/p6821> farms              0.21725243067995587
<http://www.yso.fi/onto/yso/p3294> customers          0.216395821347059
<http://www.yso.fi/onto/yso/p1834> work motivation    0.21612376226244975
<http://www.yso.fi/onto/yso/p8531> customership       0.21536113638508098
<http://www.yso.fi/onto/yso/p19047> corporate clients  0.21412270159920782
```

Evaluate a model using several measures (e.g. recall, precision, F1 score, NDCG):

```
$ annif eval tfidf-en directory-with-gold-standard-docs/
```

REST API access example

“The quick brown fox jumped over the lazy dog.”

Analyze this!



```
results=[
  {uri="<http://www.yso.fi/onto/yso/p2228>", score=0.2595, label="red fox"},
  {uri="<http://www.yso.fi/onto/yso/p5319>", score=0.2039, label="dog"},
  {uri="<http://www.yso.fi/onto/yso/p8122>", score=0.1946, label="laziness"},
  {uri="<http://www.yso.fi/onto/yso/p25726>", score=0.1285, label="brown"},
  {uri="<http://www.yso.fi/onto/yso/p4760>", score=0.1220, label="triple jump"}
]
```


What can you do with Annif?

JYX repository, University of Jyväskylä

Students upload their Master's and doctoral theses, Annif suggests subjects

Keywords

<p>Keyword suggestions</p> <p><i>Choose valid keywords by clicking</i></p>	<ul style="list-style-type: none"><input type="checkbox"/> information management systems [YSO]<input type="checkbox"/> metadata [YSO]<input type="checkbox"/> connections (technical systems) [YSO]<input type="checkbox"/> content management [YSO]<input type="checkbox"/> multimedia (information technology) [YSO]<input type="checkbox"/> digital libraries [YSO]<input type="checkbox"/> XML [YSO]<input type="checkbox"/> semantic web [YSO]<input type="checkbox"/> open source code [YSO]<input type="checkbox"/> open data [YSO]<input type="checkbox"/> user-centeredness [YSO]<input type="checkbox"/> archives (memory organisations) [YSO]<input type="checkbox"/> seeking [YSO]<input type="checkbox"/> Works [YSO]<input type="checkbox"/> cloud services [YSO]<input type="checkbox"/> electronic publications [YSO]
<p>Your own keywords</p> <p><i>Comma separated list</i></p>	<input type="text" value="keyword 1, keyword 2"/>

Implemented using
DSpace &
[GLAMpipe](#)
by Ari Häyrinen

Indexing Wikipedia by topics

Finnish Wikipedia has 410 000 articles (620 MB as raw text)

Automated subject indexing took 7 hours on a laptop, using the Annif prototype

1-3 topics per article (average ~2)

Indexing Wikipedia by topics

Finnish Wikipedia has 410 000 articles (620 MB as raw text)

Automated subject indexing took 7 hours on a laptop

1-3 topics per article (average ~2)

Examples: (random sample)

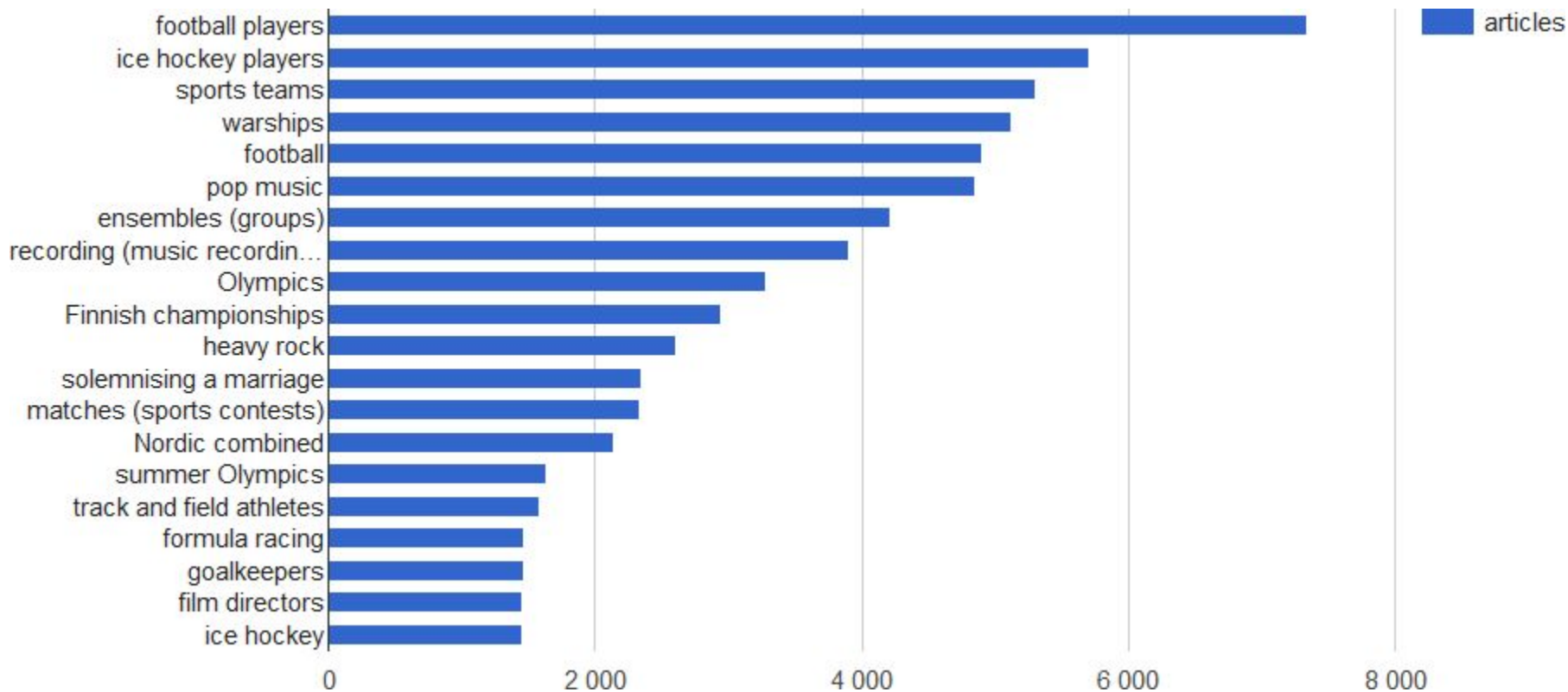
Wikipedia article

Ahvenuslammi (Urjala)
Brasilian Grand Prix 2016
Guy Topelius
HMS Laforey
Liigacup
Pää Kii
RT-21M Pioneer
Runoja
Sjur Røthe
Veikko Lavi

YSO topics

shores
race drivers, formula racing, karting
folk poetry researcher, saccharin
warships
football, football players
ensembles (groups), pop music
missiles
pop music, recording (music recordings), compositions (music)
skiers, skiing, Nordic combined
lyricists, comic songs

Most common topics in Finnish Wikipedia



Most common topics in Finnish Wikipedia



Image credits:

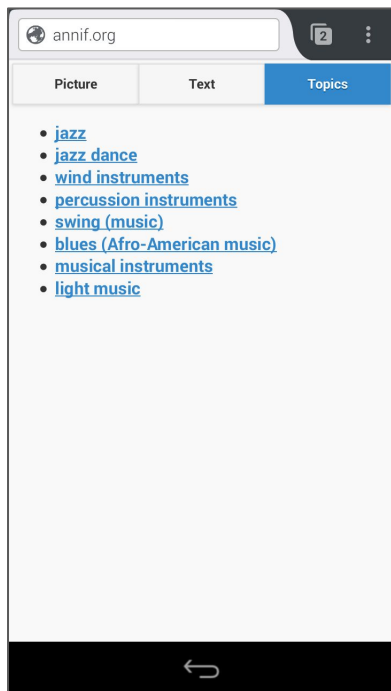
Petteri Lehtonen [CC BY-SA 3.0]

Hockeybroad/Cheryl Adams [CC BY-SA 3.0]

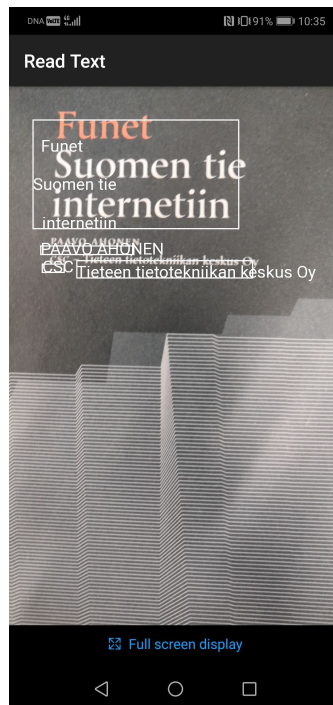
Tomisti [CC BY-SA 3.0]

Tuomas Vitikainen [CC BY-SA 3.0]

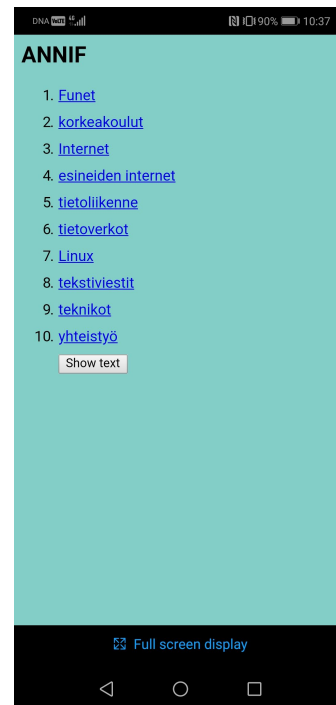
Mobile apps



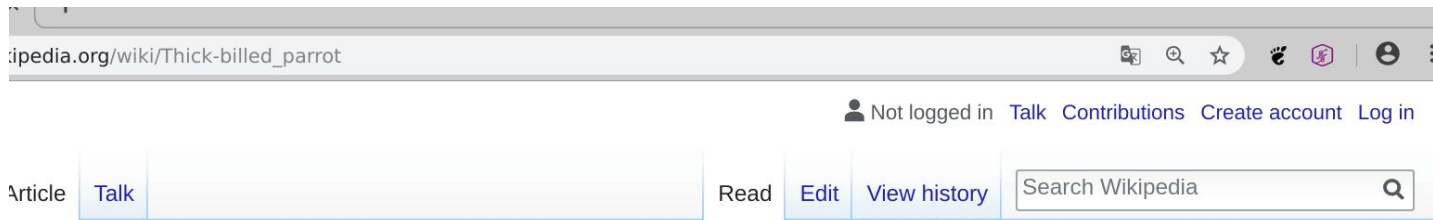
Prototype web app,
ocr.space cloud OCR
m.annif.org



Prototype Android app with OCR on the device
(by Okko Vainonen)



Finna Recommends Chrome browser extension



Analyzes selected text from any web page using Annif API and **recommends books** from Finna.fi

Thick-billed parrot

From Wikipedia, the free encyclopedia

This page is about the species of parrot. For the genus of parrots, see [Rhynchopsitta](#).

The **thick-billed parrot** (*Rhynchopsitta pachyrhyncha*) is a medium-sized green and red [parrot](#) found in Mexico, that formerly ranged into the [southwestern United States](#). Its position in parrot [phylogeny](#) is the subject of ongoing discussion; it is sometimes referred to as thick-billed [macaw](#) or thick-billed [conure](#). In Mexico, it is locally called *guacamaya* ("macaw") or *cotorra serrana* ("mountain parrot"). Classified internationally as Endangered through [IUCN](#),^[1] the thick-billed parrot's decline has been central to multiple controversies over wildlife management.

Contents [\[hide\]](#)

- [Taxonomy](#)
- [Description](#)



Created during **WIDE hackathon** by Yazan Alhalabi Samuel Akangbe Steven Nebo

Getting Annif

558 commits 10 branches 34 releases 3 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

osma Bump version: 0.33.0 → 0.34.0 Latest commit 4894a60 7 days ago

annif	break up AnnifProject.initialize() into smaller pieces (and rename of...	7 days ago
swagger	Handle errors in REST API. Part of #187	7 days ago
tests	More REST error handling tests	7 days ago
.codeclimate.yml	more comprehensive Code Climate configuration	a year ago
.codecov.yml	Codecov should ignore setup.py	6 months ago
.coveragerc	Generate Codecov reports	a year ago
.gitignore	Rename projects.cfg into projects.cfg.dist so deployments can use the...	5 months ago
.jgtn.yml	Add LGTM configuration excluding fasttext	26 days ago
.scrutinizer.yml	Try to fix pipenv/pip compatibility issue pypa/pipenv#2924 within Scr...	14 days ago
.travis.yml	use fasttextmirror package from official PyPI instead of fasttext fro...	26 days ago
LICENSE.txt	Switch to Apache license. Fixes #6	a year ago
Pipfile	Enable CORS requests to REST API using flask-cors. Fixes #190	7 days ago
README.md	add LGTM badge, drop Coveralls badge for now	26 days ago
autopep8.sh	refactor: separate merge_hits into a shared utility function	5 months ago
config.py	add tests for the initialize functionality	6 months ago
projects.cfg.dist	Add vocab settings to example configuration file, needed after #180	14 days ago
pytest.ini	add pep8 checks to pytest	7 months ago
setup.cfg	Bump version: 0.33.0 → 0.34.0	7 days ago
setup.py	Bump version: 0.33.0 → 0.34.0	7 days ago

README.md

Annif

license Apache 2.0 build passing codecov 98% maintainability A Scrutinizer 9.95 codebeat A Better Code 9 / 10

code quality: python A+

Annif on GitHub

Python 3.5+ code base

Apache License 2.0

Fully unit tested (98% coverage)

PEP8 style guide compliant

Usage [documentation](#) in the wiki

<https://github.com/NatLibFi/Annif>

annif 0.37.0

✓ Latest version

`pip install annif`

Last released: Nov 21, 2018

Automated subject indexing and classification tool

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

GitHub statistics:

★ Stars: 8

🍴 Forks: 1

🔗 Open issues/PRs: 15

View statistics for this project via [Libraries.io](#), or by using [Google BigQuery](#)

Project description

Annif

License Apache 2.0 build passing codecov 98% maintainability A Scrutinizer 9.77 codebeat A Better Code 9.7.10 code quality: python A+

Annif is an automated subject indexing toolkit. It was originally created as a statistical automated indexing tool that used metadata from the [Finna.fi](#) discovery interface as a training corpus.

This repo contains a rewritten production version of Annif based on the [prototype](#). It is a work in progress, but already functional for many common tasks.

Basic install

You will need Python 3.5+ to install Annif.

The recommended way is to install Annif from [PyPI](#) into a virtual environment.

```
python3 -m venv annif-venv
source annif-venv/bin/activate
pip install annif
```

You will also need NLTK data files:

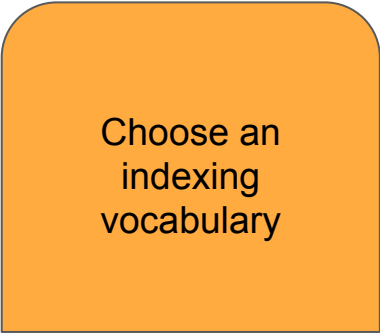
Annif on PyPI

Installing into a virtualenv:

`pip install annif`

<https://pypi.org/project/annif/>

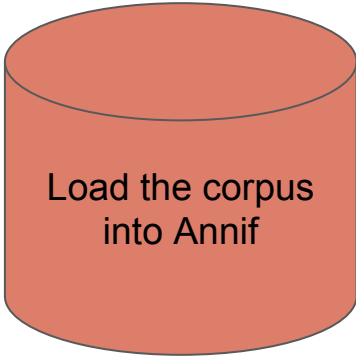
Apply Annif on your own data!




Choose an
indexing
vocabulary



Prepare a
corpus
from your
existing
metadata



Load the corpus
into Annif



Use it to index
new documents

Community group on DIY automated subject indexing?

To discuss applications, algorithms, API standards, corpora, formats etc.

Contact me if interested!

Thank you!

Questions?

osma.suominen@helsinki.fi - [@OsmaSuominen](https://twitter.com/OsmaSuominen)

Website: <http://annif.org>

API: <http://api.annif.org>

These slides: <https://tinyurl.com/annif-swib>