Annif and Finto AI: Developing and Implementing Automated Subject Indexing

Osma Suominen, Juho Inkinen, Mona Lehtinen

National Library of Finland, Finland

Preprint of article accepted on 23 June 2021 for publication in JLIS.it

Abstract

Manually indexing documents for subject-based access is a labour-intensive process that can be automated using AI technology. Algorithms for text classification must be trained and tested with examples of indexed documents, which can be obtained from existing bibliographic databases and digital collections.

The National Library of Finland has created Annif, an open source toolkit for automated subject indexing and classification. Annif is multilingual, independent of the indexing vocabulary, and modular. It integrates many text classification algorithms, including Maui, fastText, Omikuji, and a neural network model based on TensorFlow. Best results can often be obtained by combining several algorithms. Many document corpora have been used for training and evaluating Annif. Finding the algorithms and configurations that give the best quality is an ongoing effort.

In May 2020, we launched Finto AI, a service for automated subject indexing based on Annif. It provides a simple Web form for obtaining subject suggestions for text. The functionality is also available as a REST API. Many document repositories and the cataloguing system for electronic publications at the National Library of Finland are using it to integrate semi-automated subject indexing into their metadata workflows. In the future, we are going to extend Annif with more algorithms and new functionality, and to integrate Finto AI with other metadata management workflows.

Keywords

automated subject indexing, artificial intelligence, machine learning, metadata

Introduction

Extensive digitization of paper archives and more active archiving of digital material are creating growing collections of data. Subject indexing, i.e. assigning documents with subjects from a controlled vocabulary, is an important method of organizing collections and improving their discoverability. Traditionally, subject indexing is a manual process performed by human experts, but since manual indexing is a very labour-intensive process, automated and semi-automated methods for subject indexing have been developed since the 1960s (Stevens 1965).

Automating some of the subject indexing processes in Finnish libraries and related institutions has long been a goal of the National Library of Finland for several reasons: to reduce the amount of indexing work, to make the subject indexing more consistent, and to expand subject indexing to collections where traditional manual indexing is not feasible. However, from our perspective, the existing tools and services for automated subject indexing suffer from a number of problems.

First, our national languages, Finnish and Swedish, are not well supported by most tools. Second, the tools often rely on their own vocabulary, while we would like to use the General Finnish Ontology YSO[1] (Niininen, Nykyri, and Suominen 2017) as well as other Finnish subject vocabularies. Third, many of the available solutions are commercial services where the customer has little control of the system and is subject to vendor lock-in.

We started the development of Annif[2], our own open source tool for automated subject indexing, in 2017. Three years later, in May 2020, we launched Finto AI - an Annif based automated subject indexing service intended for production use[3]. In this paper, we explain the process of developing Annif, the text classification algorithms it supports, the quality assurance process we use to ensure that the algorithmically produced subject indexing meets expectations, the systems where Annif or Finto AI based automated subject indexing has been deployed, and conclude with some lessons learned.

Development of Annif

The first prototype of Annif was created in 2017, in an experiment to see if it was possible to use freely available metadata from the Finna[4] discovery system to assist in the generation of new metadata (Suominen 2019). After a successful demonstration of the approach, the National Library of Finland decided in 2018 to start development of a new version of Annif built on a more solid technical foundation and a set of goals and principles:

1. The tool should be multilingual, because in Finnish libraries, there is a need to support at least three languages: the national languages Finnish and Swedish, as well as English.
2. The tool should be independent of the indexing vocabulary; although the General Finnish Ontology is the most commonly used vocabulary in Finnish libraries, other special purpose vocabularies and library classifications such as the Dewey-based Public Library Classification YKL are widely used as well.
3. The tool should support different subject indexing algorithms; a general framework that can accommodate different algorithms was seen as more flexible and adaptable to different situations.
4. The tool should have a command line interface, a web user interface, and a REST API suitable for integration with other systems.
5. The tool should be provided as community oriented open source software; the National Library of Finland advocates for the use of open source software, as part of general openness and transparency goals, and the Skosmos vocabulary publishing software is following a similar open development model.

Based on the above goals, we created a modular architecture for Annif (Figure 1). User interaction is handled either through the command line interface (CLI) or the REST-style API that can be used to integrate Annif with other metadata management systems; the Finto AI web user interface, shown on the left in Figure 1, is an example of such a system. An embedded web user interface that relies on the REST API can also be used for interactive testing.

[1] https://finto.fi/yso/en/
[2] https://annif.org/
[3] https://ai.finto.fi/
[4] https://finna.fi

The *evaluation module* handles the calculation of various evaluation metrics such as precision, recall and F1 score. Annif is configured using a configuration file, handled by the *configuration module*. The *analyzer modules* support tokenization and normalization (stemming or lemmatization) of many languages. Indexing vocabularies, in either SKOS or a simple text format, are handled by the *vocabulary module*. The subject indexing algorithms are implemented as *backends*. The basic unit of configuration is a *project*, which is defined by specifying an indexing vocabulary, language, analyzer, backend, and project- or backend-specific parameters.

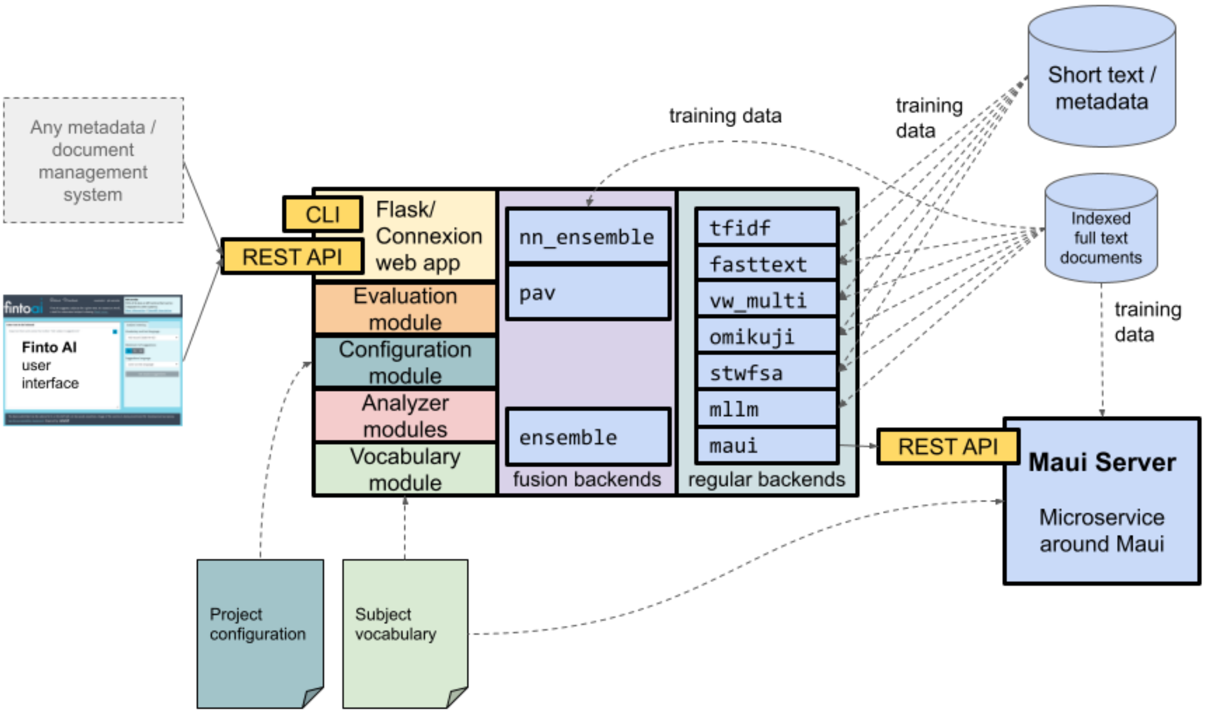

Figure 1. Modular architecture of Annif

Currently all the development of Annif happens on GitHub[5]. Annif is also made available as a Python package[6] and as Docker images[7].

Algorithms in Annif

Annif includes support for several text classification algorithms and thanks to the modular architecture, more can be added over time as backends. Backends can either function as *regular backends* or *fusion backends*. Regular backends work directly on document text and produce a suggestion of possible subjects. The algorithms implemented as regular backends in Annif are based on two main approaches: *lexical approaches* and *associative approaches* (for the distinction, see Toepfer and Seifert 2020). Fusion backends, also called *ensemble* backends, instead use the suggestions from other backends as input and produce a combined suggestion. Backends can thus be stacked and combined in many different ways.

[5] https://github.com/NatLibFi/Annif

[6] https://pypi.org/project/annif/

[7] https://quay.io/repository/natlibfi/annif

Lexical approaches

In the lexical approach, words within document text are matched with the terms contained in the subject vocabulary. For example, if the vocabulary includes the term *gross national product* and its abbreviation *GNP* (for example as an alternate label for the same concept), then that concept will be suggested as a potential subject for a document containing either the full term or the abbreviation. Since a long document will contain many such matches, lexical algorithms also need to filter and select the most promising candidates; this is typically implemented using heuristics and machine learning.

Maui (Medelyan 2009) is an example of a lexical algorithm, and is supported in Annif by integration with Maui Server[8]. **STWFSA** (Toepfer and Seifert 2020) is another lexical algorithm supported in Annif by integration with its Python implementation[9]. It has been designed specifically for extracting the maximum information from short text such as metadata records for academic publications. We have created **MLLM**[10] (Maui-like Lexical Matching), a Python reimplementation of many of the ideas in the Maui algorithm, with some adjustments such as a different string matching method and new heuristics. All the previously mentioned lexical algorithms must be trained with a sample of manually indexed documents.

Associative approaches

In the associative approach, a statistical or machine learning model is trained on a large number (typically hundreds of thousands or more) of manually indexed documents in order to find words or expressions that correlate with particular subjects. For example, the subject *renewable energy sources* could be correlated with expressions such as "energy", "solar power", "fossil free", "zero carbon", "smart grids" and "battery technology" that appear frequently in documents indexed with that subject, even though not all of them are strictly related to the subject and may not appear at all as terms in the indexing vocabulary. When a well trained associative algorithm is given a new document containing such expressions, it is likely to suggest that it could be about renewable energy sources.

As a baseline method, Annif provides a simple associative backend called **TFIDF** that calculates a vector representation for each subject based on the words that appear in documents about that subject. When given a new document, the model suggests the most similar subjects for the words in that document, based on vector similarity. **fastText** (Joulin et al. 2016) is a fast and versatile machine learning algorithm for text classification created at Facebook Research and is supported in Annif by integration through its Python bindings. **Vowpal Wabbit** (VW) is a general purpose online machine learning framework; Annif supports its algorithms for multi-class and multi-label classification, which are generally best suited for relatively small vocabularies. Finally, **Omikuji**[11] is a reimplementation of a family of efficient tree-based machine learning algorithms for multi-label classification, including Parabel (Prabhu et al. 2018) and Bonsai (Khandagale, Xiao, and Babbar 2020); it is currently the most versatile and generally best performing associative algorithm in Annif.

[8] https://github.com/TopQuadrant/MauiServer
[9] https://github.com/zbw/stwfsapy
[10] https://github.com/NatLibFi/Annif/wiki/Backend:-MLLM
[11] https://github.com/tomtung/omikuji

Fusion approaches

A fusion approach, i.e. combining different kinds of automated subject indexing algorithms, can be an effective way of improving overall performance (Toepfer and Seifert 2020). Annif provides three fusion backends: a **simple ensemble** backend, which calculates a weighted average of suggestions from several sources; and two more advanced ensemble backends which require separate training with collections of manually indexed documents. The **PAV ensemble** (Pool Adjacent Violations) uses *isotonic regression* to estimate probabilities of particular subject suggestions being correct, based on the documents the ensemble has been trained on (see Wilbur and Kim 2014), and combines the estimated probabilities to calculate an overall suggestion. The TensorFlow based **neural network ensemble** combines the simple averaging method of the simple ensemble with a multi-layer perceptron network that learns how to adjust the combined suggestions so that they best match the manual indexing that the ensemble was trained on.

Quality of automated subject indexing

As we have developed tools and services for automated subject indexing, we have assessed the quality of the automated subject indexing process along the way. According to the framework presented by Golub et al. (2016), the quality of automated subject indexing can be approached from multiple perspectives:

1. Evaluating indexing quality directly through assessment by an evaluator or by comparison with a gold standard.
2. Evaluating indexing quality in the context of an indexing workflow.
3. Evaluating indexing quality indirectly through retrieval performance.

We have so far focused on the first two perspectives, as the retrieval systems affected by the automated subject indexing processes (e.g. Finna) are quite far removed from the subject indexing processes and affected by numerous other factors as well.

API service configurations to evaluate

While we have performed many evaluations of individual algorithms during the development of Annif, the most thorough evaluations have been performed on the combinations of projects, backends, configuration settings, and training data sets that have been provided for public use in the API service for Annif and (since May 2020) Finto AI. The first public API service, after the initial prototype, was published in January 2018, with support for suggesting subjects from the General Finnish Ontology YSO for documents in Finnish, Swedish or English. We set up an ensemble project combining results from three different algorithms for each language. The associative algorithms were trained using metadata extracted from the Finna discovery system, while lexical and ensemble backends were trained on various collections of full text documents. Subsequently we have updated the API service with newer versions of the YSO vocabulary (including YSO Places from January 2020 onwards) and switched the backend algorithms and the ensemble type as new options have been developed. The changes to the API service configurations have been summarized in Table 1.

Date	YSO version	Ensemble type	Backends

2018-01	2017-03 snapshot	Simple ensemble	TFIDF, fastText, Maui
2020-01	2019-03 Cicero	Simple ensemble	Omikuji-Parabel, Omikuji-Bonsai, Maui
2020-03	2020-01 Diotima	Neural network	Omikuji-Parabel, Omikuji-Bonsai, Maui
2020-12	2020-10 snapshot	Neural network	fastText, Omikuji-Bonsai, Maui
2021-04	2021-03 Epikuros	Neural network	fastText, Omikuji-Bonsai, MLLM

Table 1. API service configurations.

Comparison to gold standard

A gold standard is a collection in which each document is assigned a set of subjects that is assumed to be complete and correct (Golub et al. 2016). Once a gold standard has been developed, it is easy to evaluate automated subject indexing methods against it by measuring how well the algorithmic suggestions match the gold standard. However, creating a good quality gold standard takes a significant amount of effort and requires input from many experts. In practice, existing manually indexed documents are often used as a substitute for a properly constructed gold standard, as in the evaluation of the Maui algorithm (Medelyan 2009). Such collections are readily available and they enable easy experimentation and comparison of different algorithms, but as the indexing process is susceptible to many kinds of bias, they are best used as ballpark estimates of quality and must be complemented with other types of evaluation.

We have used the following manually indexed corpora for evaluation. The first three include documents in Finnish, Swedish and English, the last two are only in Finnish.

1. **JYU theses**: Master's and doctoral theses from the University of Jyväskylä (n=7,400) published in the years 2010 to 2017. These are long, in-depth academic documents that cover many disciplines.
2. **Electronic deposits**: Non-fiction electronic books (n=9832) published between 1998 and 2019 that have been deposited to the National Library of Finland and indexed in the national bibliography Fennica.
3. **Book descriptions**: Titles and short descriptions of non-fiction books (n=51309) collected from the database of the book distributor Kirjavälitys Oy, covering the time period from approximately 2000 to 2019. The book descriptions were originally created by publishers for marketing purposes. The subject indexing for these works was obtained separately from the national bibliography Fennica.
4. **Ask a Librarian**: Question and answer pairs from the Ask a Librarian service run by public libraries in Finland. The original database consisted of over 25,000 documents but we extracted the subset with a minimum of 4 subjects per document (n=3,150). These are short, informal questions and answers about many different topics.
5. **Satakunnan Kansa**: Digital archives of Satakunnan Kansa regional newspaper. The archives consist of over 100,000 unindexed documents. Out of these, a random sample of 50 documents was manually indexed by four librarians working independently.

We split these collections into train, validate and test subsets. Only the test subsets were used as gold standard sets for the evaluation of algorithms. We mainly used the F1@5 metric for the evaluation: that is, the F1 score similarity (harmonic mean of precision and recall) between the manually assigned subjects and the top 5 suggestions of the algorithm. The results are summarized in Figure 2. We can see that the overall F1 scores have generally improved with successive API service configurations. The best F1 scores of around 0.6-0.7 were obtained with Swedish language documents from the JYU theses and electronic deposit collections; however, these measurements are also the least reliable, since due to the small number of Swedish language documents in these collections, we had reused some of the same documents for both training and evaluating the Maui, MLLM and neural network ensemble models. If we exclude the two Swedish language collections with unrealistically good results, we have reached F1 scores ranging between 0.3 and 0.5.

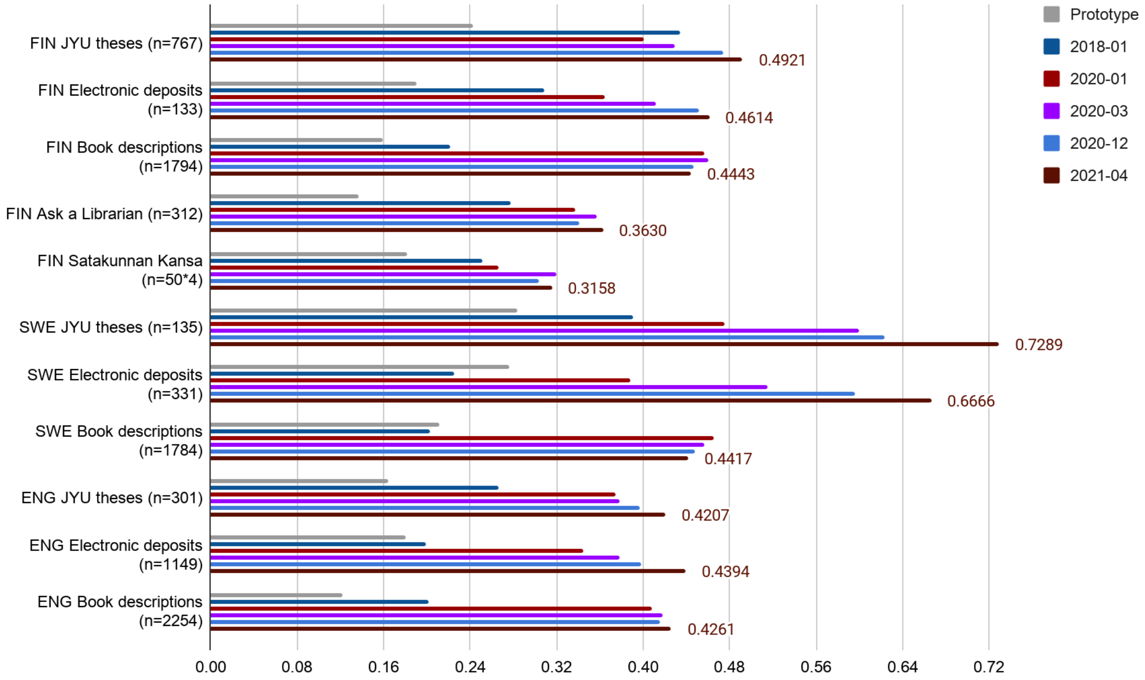

Figure 2. F1@5 scores for the test collections, by API service configuration. The most recent numeric scores for the 2021-04 API service configuration are also shown; these are preliminary numbers, as the 2021-04 configuration was not yet deployed by the time the article was submitted.

Assessment by evaluators

Having human evaluators assess the suggested subjects is another way to measure the quality of automatic subject indexing. In 2019, we organized a workshop where 48 participants (mainly librarians and informaticians) were given 50 example documents, with on average more than 10 sets of subjects assigned to each document. The indexing had been created either by humans (professional or lay) or by different Annif algorithms, but the participants did not know which was which. The participants used a scale from 1-5 to evaluate the indexing from three viewpoints: overall quality, meaningfulness and coverage. In general the human assigned subjects got higher scores, but the difference wasn't very large. Figure 3 shows the evaluation results. Indexing by the best performing Annif PAV ensemble model usually received a grade of around 3 out of 5, while human indexers scored between 3.5 and 4 out of 5, with professionals performing the best. Annif-assisted semi-automatic indexing landed in between. (Lehtinen, Inkinen, and Suominen 2019)

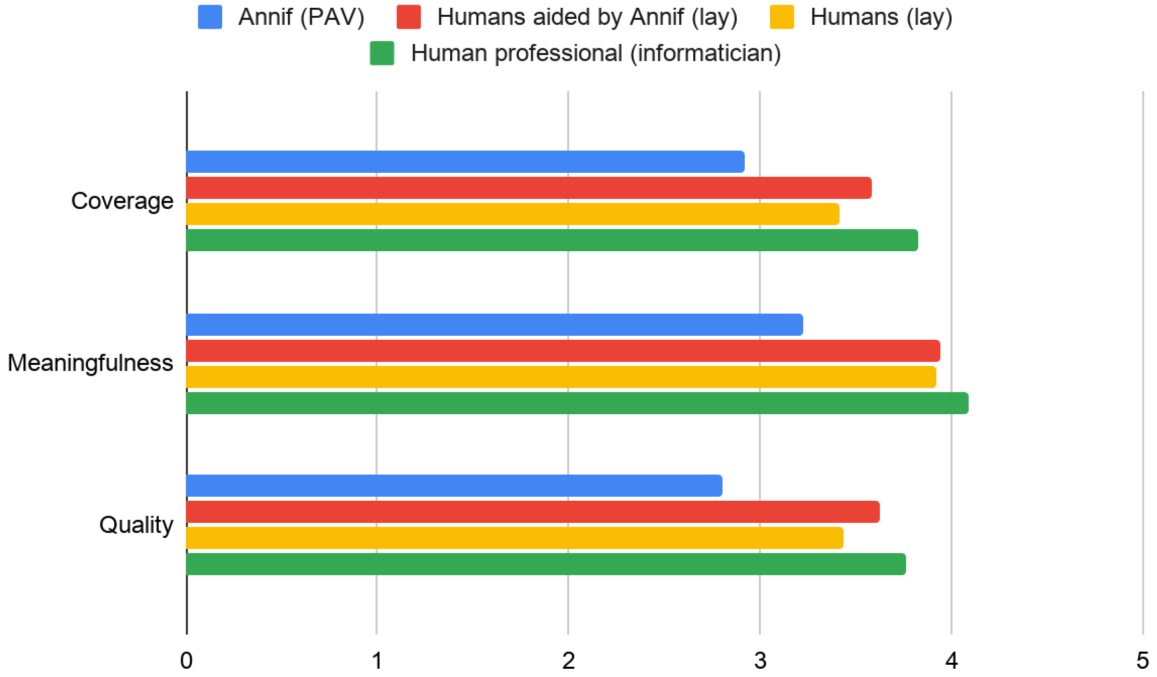

Figure 3. Quality evaluation of intellectually given and Annif-produced subject indices. Data reproduced from Lehtinen, Inkinen, and Suominen 2019.

A similar comparison was performed by the Finnish Public Broadcasting Company Yle. Their tests compared Annif against a commercial document classification service Leiki that they have been using in production for several years. In their results, Annif was rated as slightly better than Leiki for Finnish language documents and as much better for Swedish language documents. The quality of the metadata they used for training Annif might explain the differences between the languages (Suominen and Virtanen 2020; Nikkarinen 2021).

The Research department of the National Library of the Netherlands has also evaluated and used Annif as a part of developing their own larger tool for automated indexing (Haighton and Veldhoen 2020). The German National Library has evaluated Annif as well, comparing it with their current automated indexing system both qualitatively and quantitatively. Seven out of nine Annif's algorithms outperformed the current solution in F1@5 scores. Human evaluators also rated Annif's suggestions as more useful than those of the current system (Uhlmann 2020).

Evaluating in the context of an indexing workflow

We have also evaluated the quality of Annif in the context of the indexing workflow of the JYX[12] institutional repository of the University of Jyväskylä, which was an early adopter of Annif. JYX integrates Annif into its upload form. Students who upload their completed Master's thesis receive suggestions from Annif and can accept or reject the suggestions as well as add their own keywords. Later in the process, informaticians validate the metadata and can make corrections to the subjects.

The system saves the original suggestions by Annif as well as the users' choices, so it is possible to keep track of how many of the Annif suggestions are accepted by the student and the final validated subjects. Figure 4 shows the F1 score similarities between the original Annif suggestions and the student-selected and final subjects, over several generations of API service configurations. There was a marked increase in the similarity after the initial prototype; since then, a small increase in similarity to the Annif suggestions can be seen in both the student-selected and final subjects, suggesting that the acceptability of the automated suggestions has increased over time.

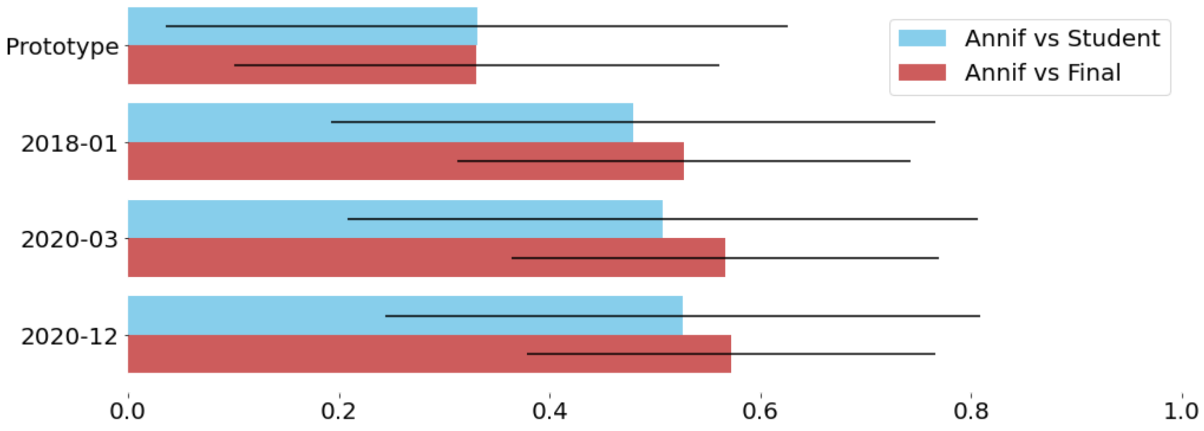

Figure 4. F1 score similarity between Annif suggestions, student-selected subjects and final subjects in JYX, for the Annif prototype and subsequent API service configurations. Data is missing for the short-lived 2020-01 configuration.

Users of the Annif API service and Finto AI

A service for automated subject indexing based on Annif has been existing since 2017 at the annif.org website, but its main purposes have been testing and development. The Finto AI service we launched in May 2020 is intended for production use. The service offers an easy way for introducing automatic subject indexing into information systems, provided that the vocabularies and language support offered by the API service meet local requirements. Some of the systems integrated with Finto AI are shown in Figure 5.

Generally, when the API service is integrated in the indexing workflow of a document repository, the steps in processing a document are:

1. extract the text from the document (typically a PDF file)
2. detect the language of the text (if not already known)
3. send the text to Annif via the *suggest* method of the API; the specific endpoint is chosen based on the text language and the indexing vocabulary

[12] https://jyx.jyu.fi/

4. display the returned subject suggestions to the user
5. the user selects the subjects to be stored in the document metadata; the user can also add subjects that were not suggested by Annif

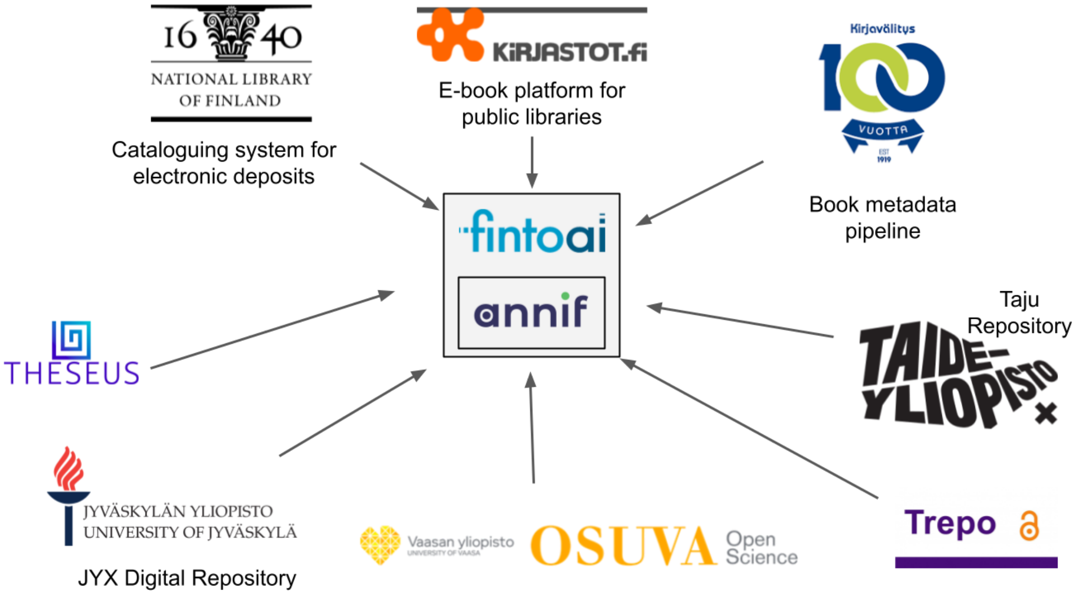

Figure 5. Institutional users of Finto AI.

Institutional repositories

The very first institutional user of semi-automated subject indexing by Annif was the **JYX repository** of the University of Jyväskylä, which is based on DSpace software. Already in 2017 they integrated the API of the Annif prototype system into their pipeline, which is used by students to upload their Master's or doctoral theses. As explained above, a librarian may correct the student-selected subjects when validating the metadata.

Since 2020, four DSpace based university repositories maintained by the National Library of Finland have started using Finto AI in their uploading pipeline: **Osuva**[13] (University of Vaasa), **Trepo**[14] (University of Tampere), **Taju**[15] (University of Arts) and **Theseus**[16] (used by many Finnish universities of applied sciences). Their workflow is similar to JYX, but there is no validation step performed by a librarian.

The electronic deposit system at the National Library of Finland

The National Library of Finland maintains an uploading service for individual deposits of electronic publications[17]. The API of Finto AI was integrated in 2020 to the metadata workflow of the internal deposit repository Varsta. The subject suggestions are not shown to the uploader, but to a library cataloger who curates the metadata in the Varsta system. The metadata is then

[13] https://osuva.uwasa.fi/
[14] https://trepo.tuni.fi/
[15] https://taju.uniarts.fi/
[16] https://www.theseus.fi/
[17] https://luovutuslomake.kansalliskirjasto.fi

stored in the Melinda union catalogue. The publication files are stored in the Varia repository, which can be browsed using the computers within the premises of the National Library of Finland. See Figure 6 for an overview of the pipeline.

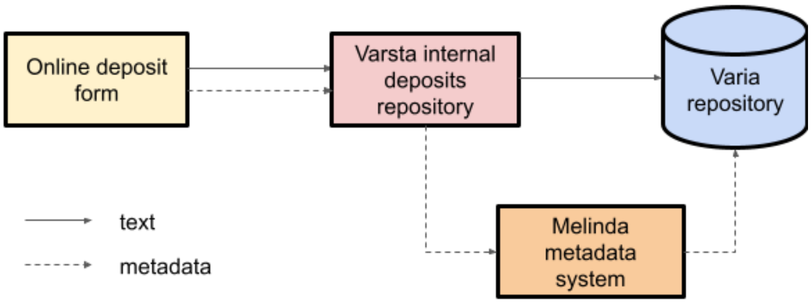

Figure 6. Data flows for individual electronic publication deposits in the systems of the National Library ofFinland.

Book distributor Kirjavälitys Oy

Kirjavälitys Oy[18] is a Finnish book distributor that handles book-sale logistics. They receive information about upcoming titles from publishers and produce metadata used in libraries, booksellers and the union catalogue Melinda, which includes the Finnish national bibliography Fennica (see Figure 7). Kirjavälitys has integrated the API of Finto AI in their system to aid in subject indexing of non-fiction books. They use the back-cover description text of books as the input to Finto AI.

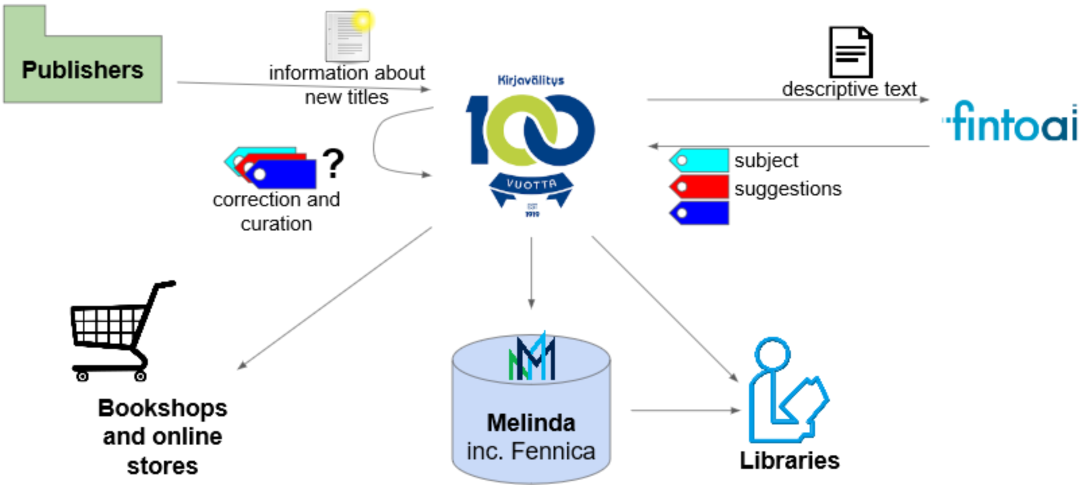

Figure 7. The book distributor Kirjavälitys Oy receives information from publishers, enhances it with subject indexing assisted by Finto AI, and produces widely used metadata.

[18] https://www.kirjavalitys.fi/en/home/

Standalone Annif installations

To have more control on the indexing, e.g., for using a custom vocabulary or achieving better indexing quality on a specific topic area, or to support a language not available in Finto AI, a user can install and set up Annif by themselves and train their own models. Training a well-performing Annif model requires possessing adequate amounts of suitable training data, and can be computationally heavy. Searching for good hyperparameters for a model takes a lot of computation time. In contrast, when a model has been trained, and it is used by an Annif instance to offer subject indexing functionality via API, much less CPU resources are needed. For these reasons it can be worthwhile to have separate computing environments for training Annif models and for serving them.

Here we present some institutions that have set up their own Annif installations.

The **Leibniz Information Centre for Economics ZBW** has a long history of developing automated subject indexing solutions. Currently they are working on the AutoSE project with the aim of transferring their existing automation solutions into productive use (Kasprzik 2020). They use Annif as a part of their framework, and also activelycontribute to the development of Annif.

The **Finnish Broadcasting Company Yle** is setting up Annif for semi-automatic subject indexing of online news articles. They use their own vocabulary and training corpus, and as their custom vocabulary evolves rapidly, they retrain their Annif models every week (Nikkarinen 2021).

The Finnish **National Audiovisual Institute (KAVI)** offers various services, such as film digitization, and maintains archives. They are also responsible for content rating and screenings of audiovisual material in Finland. KAVI first tested Annif as a standalone installation for indexing radio and TV programs using a speech-to-text transcript of the audio content. Based on the test results, KAVI decided to adopt Annif for this use in their future archive management system (Lehtonen and Piukkula 2020).

The **National Library of the Netherlands** has explored the possibilities of automatic indexing (Kleppe et al. 2019). Annif is now used as a part of their larger tool that is being developed for library catalogers (Haighton and Veldhoen 2020). The training data for their current models have been gathered from a collaborative cataloguing system for Dutch libraries. The data consists of titles, subtitles and summaries of Dutch e-books. The Brinkman thesaurus[19] has been used as the controlled vocabulary. Annif has also been applied in a Dutch research project called Entangled Histories. The project focused on early modern ordinances, i.e. law texts, and Annif was used in their classification (Romein, Veldhoen, and de Gruijter 2020).

Dissemin[20] is an online service for researchers to find open publishing repositories for their publications. Dissemin uses Annif to categorize academic pre- and postprints uploaded to open repositories.

Community building

We aim to foster a community around Annif and to make it easy for people to learn about it. Annif has a website that serves as an introduction and an interactive demo. In the Annif GitHub project, we offer a thorough technical description and tips for Annif use. Users can also report

[19] https://www.kb.nl/sites/default/files/docs/brinkmanonderwerpen-2018.pdf
[20] https://dissem.in/

bugs or contribute ideas and solutions using GitHub issues and pull requests. There is also a user forum called annif-users[21] where people can ask for help, discuss and share their experiences. The forum is also a platform for Annif-related announcements and news.

Together with ZBW, we have created a hands-on tutorial[22] to help people get started with Annif. The first tutorial session was held at the SWIB19 conference[23]. When the Covid-19 pandemic hit in 2020, we turned the material into an online tutorial suitable for self study, with videos on YouTube and exercises on GitHub. We have organized several interactive workshops based on the tutorial materials at suitable online conferences.

We also took part in the EU-funded High-Performance Digitisation project, which was a joint effort with CSC – IT Center for Science and the National Archives of Finland. The project sought to find intelligent solutions for automatic indexing workflow in LAM organizations. We were really pleased with this collaboration, which resulted in e.g. the discovery and thorough evaluation of the highly efficient Omikuji algorithms that were later integrated into Annif. The project is described on its web page[24] and in Lehtinen & Kallio (2020). The project also produced a whitepaper (in Finnish) describing the uses and challenges of automatic subject indexing in a cultural heritage organization, with Annif as an example (Hulkkonen et al. 2021).

Conclusion and Lessons Learned

Manually indexing documents for subject-based access is a labour-intensive process, and with the growing mass of digital material it becomes more and more difficult to keep up. There is a need for automation. Although it has taken several years and a lot of development effort, we have successfully created an open source solution for multilingual, vocabulary independent automated subject indexing that has become a production service used in many Finnish libraries, especially through the Finto AI service.

Annif is a unique framework into which different text classification algorithms can be integrated. The algorithms may be used alone, or in combinations called ensembles. We have found that the ensembles nearly always perform better than the individual algorithms.

Subject indexing is not an easy process, either for human indexers or for algorithms. Some parts of it are inherently subjective. When humans do subject indexing, they can have very different perspectives, or sometimes simply make mistakes. These types of mistakes or differences of opinion, however, are usually still relatable or understandable. When algorithms do subject indexing, their mistakes often do not necessarily make any sense from a human perspective.

There are many approaches for evaluating the quality of automated subject indexing systems. We have found that a combination of approaches works well for our purposes. Quantitative comparisons to a human indexed gold standard are the easiest to produce, and we perform them frequently both for the purpose of algorithm development and for evaluating the models that we deploy into production services. User oriented evaluation methods, such as assessment by evaluators, are more laborious, but they produce important insights about how algorithmically produced subject indexing differs from manually created indexing. Organizing workshops around automated subject indexing has provided a way of crowdsourcing the human evaluation effort,

[21] https://groups.google.com/g/annif-users
[22] https://github.com/NatLibFi/Annif-tutorial/
[23] https://swib.org/swib19/
[24] https://www.csc.fi/en/-/high-performance-digitisation

while simultaneously spreading awareness about automated indexing among librarians. We have also started to track how our tool is being used in the indexing workflow of systems that are using our API services. In the future, it would also be possible to investigate how the use of automated indexing affects users of retrieval systems.

The Annif tool is increasingly being deployed in Finnish library systems by integration with the API services provided by Finto AI. The Finto AI web user interface is also being used directly by librarians in cases where direct integration between systems is not feasible or has not yet been implemented. So far, users have been very positive towards the subject suggestions given by the service, as it provides an initial suggestion of potential subjects instead of an empty field to fill in. This is especially important for university library repositories where students, who are usually not experts in subject indexing, upload their own thesis documents.

The API services available through Finto AI are currently limited in the terms of indexing vocabularies and languages we can offer. We are working with Finnish organizations that have more diverse needs, for example custom domain-specific vocabularies, so that we can expand the service in the future.

Annif has been community oriented open source software from the start. We have created a web site and a wiki with technical documentation, set up a user forum, presented the tool at conferences and webinars, and together with ZBW, produced a tutorial for learning the basics of the tool. The effort put into community building is starting to pay off, as we are seeing an increasing number of test installations of Annif and some organisations are investing seriously in the adoption of Annif, for example by making extensive tests and comparisons.

One of the challenges in adopting Annif is collecting suitable training data and converting it to the corpus formats that Annif understands. This process usually requires programming skills. Even with a corpus in the correct format, achieving and maintaining good quality can be a challenge. We have gathered advice for setting up and refining projects into a wiki page[25].

There are upsides and downsides of the open source model for library systems. It allows for freedom and flexibility, but requires more technical expertise and resources than similar systems and services provided by commercial vendors. Organizations adopting an open source solution must be prepared to build the in-house expertise required to set up and maintain the systems. Some of the development effort can be shared and pooled through co-operating using code sharing platforms such as GitHub.

In the future we continue to actively develop Annif and Finto AI. We hope to keep the community involved and welcome any contributions and feedback. Our aim is to support more vocabularies and languages in the Finto AI service while following the development of new text classification algorithms and utilizing them.

Acknowledgements

We thank the institutions and people who provided us with the corpora that have been used to train and evaluate the automated subject indexing methods, and Ari Häyrinen for providing the data used for the evaluation of Annif in the contextof the JYX repository indexing workflow.

[25] https://github.com/NatLibFi/Annif/wiki/Achieving-good-results

References

Golub, Koraljka, Dagobert Soergel, George Buchanan, Douglas Tudhope, Marianne Lykke, and Debra Hiom. 2016. 'A Framework for Evaluating Automatic Indexing or Classification in the Context of Retrieval'. *Journal of the Associationfor Information Science and Technology* 67 (1): 3–16. https://doi.org/10.1002/asi.23600.

Haighton, Thomas, and Sara Veldhoen. 2020. 'Assisted Keyword Assignment Using Annif. KB Lab: The Hague.' 2020. http://kbresearch.nl/annif/.

Hulkkonen, Juha, Juho Inkinen, Aleksi Kallio, Markus Koskela, Mikko Lappalainen, Mona Lehtinen, Mats Sjöberg, Osma Suominen, and Laxmana Yetukuri. 2021. 'Sisällönkuvailun automatisoinnin haasteita ja ratkaisuja kulttuuriperintöorganisaatiossa'. Kansalliskirjaston raportteja ja selvityksiä. http://urn.fi/URN:ISBN:978-951-51-7233-4.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. 'Bag of Tricks for Efficient Text Classification'. *ArXiv:1607.01759 [Cs]*, August. http://arxiv.org/abs/1607.01759.

Kasprzik, Anna. 2020. 'Putting Research-Based Machine Learning Solutions for Subject Indexing into Practice'. In *Proceedings of the Conference onDigital Curation Technologies (Qurator 2020)*. Berlin, Germany. http://ceur-ws.org/Vol-2535/paper_1.pdf.

Khandagale, Sujay, Han Xiao, and Rohit Babbar. 2020. 'Bonsai: Diverse and Shallow Trees for Extreme Multi-Label Classification'. *Machine Learning* 109 (11): 2099–2119. https://doi.org/10.1007/s10994-020-05888-2.

Kleppe, Martijn, Sara Veldhoen, Meta van der Waal-Gentenaar, Brigitte den Oudsten, and Dorien Haagsma. 2019. 'Exploration possibilities Automated Generation of Metadata'. Zenodo. https://doi.org/10.5281/zenodo.3375192.

Lehtinen, Mona, Juho Inkinen, and Osma Suominen. 2019. 'Aaveita koneessa: Automaattisen sisällönkuvailun arviointia Kirjastoverkkopäivillä 2019'. *Tietolinja* (blog). 2019. http://urn.fi/URN:NBN:fi-fe2019120445612.

Lehtonen, Tommi, and Juha Piukkula. 2020. 'Automaattinen asiasanoitus Radio- ja televisio-ohjelmatietokanta Ritvassa'. *Informaatiotutkimus* 39 (1): 27–45–27–45. https://doi.org/10.23978/inf.88107.

Medelyan, Olena. 2009. 'Human-Competitive Automatic Topic Indexing'. Thesis, The University of Waikato. https://researchcommons.waikato.ac.nz/handle/10289/3513.

Niininen, Satu, Susanna Nykyri, and Osma Suominen. 2017. 'The Future of Metadata: Open, Linked, and Multilingual – the YSO Case'. *Journal of Documentation*73 (3): 451–65. https://doi.org/10.1108/JD-06-2016-0084.

Nikkarinen, Irene. 2021. 'Annif <3 Yle 2.0: Annifin osittainen käyttöönotto artikkeleiden koneavusteisessa asiasanoituksessa'. Presented at the Meeting of the Finnish Automatic Indexing Interest Group, March 15. https://www.kiwi.fi/display/tekoalykumppanuus/Automaattisen+kuvailun+verkoston+tapaamiset?preview=/147358597/211911484/Automaattisen%20kuvailun%20verkoston%20tapaaminen%2015.3.2021%20Annif.pdf.

Prabhu, Yashoteja, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. 'Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising'. In *Proceedings of the 2018 WorldWide Web Conference*, 993–1002. WWW '18. Lyon, France. https://doi.org/10.1145/3178876.3185998.

Romein, C. Annemieke, Sara Veldhoen, and Michel de Gruijter. 2020. 'The Datafication of Early Modern Ordinances'. *DH Benelux Journal* 2.

https://journal.dhbenelux.org/journal/issues/002/article-23-romein/article-23-romein.html.

Stevens, Mary Elizabeth. 1965. *Automatic Indexing: A State-of-the-Art Report.* NBS Monograph 91. Washington, D.C: United States. Government Printing Office.

Suominen, Osma. 2019. 'Annif: DIY Automated Subject Indexing Using Multiple Algorithms'. *LIBER Quarterly* 29 (1): 1. https://doi.org/10.18352/lq.10285.

Suominen, Osma, and Pia Virtanen. 2020. 'Yle Meets ANNIF – an Open Source Tool for Automated Subject Indexing'. Presented at the EBU MDN Workshop 2020, June 10. https://tech.ebu.ch/contents/publications/events/presentations/mdn2020/yle-meets-annif--an-open-source-tool-for-automated-subject-indexing.

Toepfer, Martin, and Christin Seifert. 2020. 'Fusion Architectures for Automatic Subject Indexing under Concept Drift: Analysis and Empirical Results on Short Texts'. *International Journal on Digital Libraries* 21 (2): 169–89. https://doi.org/10.1007/s00799-018-0240-3.

Uhlmann, Sandro. 2020. 'Automatische Vergabe von GND-Schlagwörtern Mit Annif - Ergebnisse Einer Evaluation Im DNB - Projekt EMa'. Presented at the Erfahrungen und Perspektiven mit dem Toolkit Annif, December 3. https://wiki.dnb.de/display/FNMVE/Erfahrungen+und+Perspektiven+mit+dem+Toolkit+Annif?preview=/181751388/190121925/2-3_Automatische-Vergabe-von-GND-Schlagw%C3%B6rtern_Uhlmann_2020-12-03_final.pdf.

Wilbur, W. John, and Won Kim. 2014. 'Stochastic Gradient Descent and the Prediction of MeSH for PubMed Records'. *AMIA Annual Symposium Proceedings* 2014 (November): 1198–1207.

=== HASIL INDEXING OTOMATIS ===

1. Metode TF-IDF:
- annif (Halaman: 1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15, 16)
- indexing (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- subject (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- ev (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16)
- alg (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 16)
- ser (Halaman: 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14)
- ha (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- 2020 (Halaman: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 13, 15, 16)
- er (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- orithms (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 16)
- sug (Halaman: 1, 3, 4, 5, 6, 7, 8, 9, 10, 14)
- automated (Halaman: 1, 2, 5, 6, 8, 9, 10, 12, 13, 14, 16)
- es (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- vice (Halaman: 1, 2, 5, 6, 7, 8, 9, 10, 12, 13, 14)
- used (Halaman: 1, 2, 5, 6, 7, 8, 10, 11, 12, 13, 14)
- documents (Halaman: 1, 4, 5, 6, 7, 8, 13, 14)
- api (Halaman: 1, 2, 5, 6, 7, 9, 10, 11, 12, 14)
- finnish (Halaman: 1, 2, 5, 6, 8, 10, 11, 12, 13, 14, 15)
- national (Halaman: 1, 2, 4, 6, 8, 10, 11, 12, 13, 16)
- ai (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
- also (Halaman: 1, 2, 3, 4, 7, 8, 9, 10, 12, 13, 14)
- librar (Halaman: 1, 2, 6, 7, 8, 10, 11, 12, 13, 14, 16)
- finto (Halaman: 1, 2, 5, 9, 10, 11, 12, 13, 14)
- based (Halaman: 1, 2, 3, 4, 5, 9, 10, 12, 13, 15)
- org (Halaman: 1, 7, 9, 13, 14, 15, 16)
- dev (Halaman: 1, 2, 3, 5, 6, 8, 9, 12, 13, 14)
- subjects (Halaman: 1, 3, 4, 5, 6, 7, 9, 10, 14)
- estions (Halaman: 1, 3, 5, 6, 7, 8, 9, 10, 14)
- text (Halaman: 1, 2, 3, 4, 5, 8, 9, 11, 12, 13, 14, 15)
- metadata (Halaman: 1, 2, 4, 5, 8, 9, 10, 11, 15)
- ed (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)
- many (Halaman: 1, 2, 3, 4, 5, 6, 9, 10, 13, 15)
- integ (Halaman: 1, 2, 4, 9, 10, 11, 13, 14)
- document (Halaman: 1, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15)
- quality (Halaman: 1, 2, 5, 6, 7, 8, 9, 12, 13, 14)
- ocabular (Halaman: 1, 2, 3, 4, 5, 9, 12, 13, 14)
- classification (Halaman: 1, 2, 3, 4, 8, 12, 13, 14, 15)
- whic (Halaman: 1, 3, 4, 5, 6, 7, 9, 10, 11, 13)
- finland (Halaman: 1, 2, 6, 10, 11, 12, 13)
- man (Halaman: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16)
- use (Halaman: 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)
- using (Halaman: 1, 3, 4, 5, 8, 10, 11, 12, 13, 14, 15, 16)
- ork (Halaman: 1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14, 15, 16)
- languag (Halaman: 2, 3, 5, 7, 8, 9, 12, 14)
- systems (Halaman: 2, 5, 9, 11, 13, 14)
- figure (Halaman: 2, 3, 7, 8, 9, 10, 11)
- bac (Halaman: 3, 4, 5, 11, 14)
- ends (Halaman: 3, 5)
- aluation (Halaman: 3, 5, 6, 7, 8, 13, 14, 16)
- ensemble (Halaman: 5, 6, 7, 8, 13)

2. Metode RAKE:
- sev eral reasons (Halaman: 1)
- automated subject indexing (Halaman: 1, 2, 5, 16)
- j une f (Halaman: 1)
- semi automated methods (Halaman: 1)
- ar ticle accepted (Halaman: 1)
- indexing w ork (Halaman: 1)
- perspecti v e (Halaman: 1)
- gi v e (Halaman: 1)
- subject indexing processes (Halaman: 1, 5)
- raditionally subject indexing (Halaman: 1)
- expand subject indexing (Halaman: 1)
- data subject indexing (Halaman: 1)
- future w e (Halaman: 1)
- e assigning documents (Halaman: 1)
- indexing v ocabular (Halaman: 1, 2, 3, 9)
- alg orithms (Halaman: 1)
- open source softw (Halaman: 2)
- wn v ocabular (Halaman: 2)
- ha v e (Halaman: 2, 5, 6, 8, 12, 14)
- command line interface (Halaman: 2)
- commercial ser vices (Halaman: 2)
- transparency g oals (Halaman: 2)
- v ailable metadata (Halaman: 2)
- g eneral openness (Halaman: 2)
- v ailable solutions (Halaman: 2)
- new v ersion (Halaman: 2)
- dev eloping annif (Halaman: 2)
- bac k ends (Halaman: 3)
- modular arc hitecture (Halaman: 3)
- many different w (Halaman: 3)
- stac k ed (Halaman: 3)
- sug g estion (Halaman: 3)
- v er time (Halaman: 3)
- j e c (Halaman: 3)
- sug g estions (Halaman: 3, 8, 9)
- alg orithms implemented (Halaman: 3)
- r e p (Halaman: 3, 10)
- also made (Halaman: 3)
- either sk os (Halaman: 3)
- annif cur rently (Halaman: 3)
- appr oaches (Halaman: 3)
- f score annif (Halaman: 3)
- v ailable (Halaman: 3)
- handled b (Halaman: 3)
- f h (Halaman: 3)
- simple text (Halaman: 3)
- document text (Halaman: 3)
- ys h (Halaman: 3)
- q u (Halaman: 3)
- configuration module (Halaman: 3)
- b f (Halaman: 3)
- lexical appr oaches (Halaman: 4)

- lexical alg orithm (Halaman: 4)
- maui alg orithm (Halaman: 4)
- text classification created (Halaman: 4)
- documents index ed (Halaman: 4)
- gi v en (Halaman: 4)
- perspecti v es (Halaman: 5)
- v er documents (Halaman: 6)
- different alg orithms (Halaman: 6)
- g old standard (Halaman: 6)
- many exper ts (Halaman: 6)
- w e extracted (Halaman: 6)
- w orks w (Halaman: 6)
- uni v ersity (Halaman: 6, 9, 10, 15)
- answ er pairs (Halaman: 6)
- index ed (Halaman: 6)
- document n (Halaman: 6)
- many kinds (Halaman: 6)
- f ennica ask (Halaman: 6)
- ev aluation (Halaman: 6)
- finnish sw edish (Halaman: 6)
- documents w (Halaman: 6)
- subject indexing (Halaman: 6, 11)
- recall betw een (Halaman: 7)
- v erag e (Halaman: 7)
- figure f scores (Halaman: 7)
- best f scores (Halaman: 7)
- collections w e (Halaman: 7)
- w e split (Halaman: 7)
- figure w e (Halaman: 7)
- train v alidate (Halaman: 7)
- ar ticle w (Halaman: 7)
- par ticipants used (Halaman: 7)
- test collections b (Halaman: 7)
- eac h document (Halaman: 7)
- three view points (Halaman: 7)
- small n umber (Halaman: 7)
- f metric (Halaman: 7)
- created either b (Halaman: 7)
- g eneral (Halaman: 7)
- configuration w (Halaman: 7)
- also ev aluated (Halaman: 8)
- ev aluated annif (Halaman: 8)
- cur rent solution (Halaman: 8)
- differences betw een (Halaman: 8)
- automated indexing haighton (Halaman: 8)
- results annif w (Halaman: 8)
- perfor med b (Halaman: 8)
- qualitati v ely (Halaman: 8)
- professionals perfor ming (Halaman: 8)
- uc h better (Halaman: 8)
- sev eral years (Halaman: 8)
- languag es suominen (Halaman: 8)
- vir tanen nikkarinen (Halaman: 8)

- w ell comparing (Halaman: 8)
- similar comparison w (Halaman: 8)
- kf lo w (Halaman: 8)
- difference w asn (Halaman: 8)
- indexing w (Halaman: 8)
- v er (Halaman: 8, 10)
- nine annif (Halaman: 8)
- v eldhoen (Halaman: 8)
- used annif (Halaman: 8)
- national librar (Halaman: 8, 11)
- lehtinen inkinen (Halaman: 8)
- api ser vice (Halaman: 9)
- e r v (Halaman: 9)
- ser vice offers (Halaman: 9)
- student selected (Halaman: 9)
- sug gest method (Halaman: 9)
- ho w many (Halaman: 9)
- figure generally (Halaman: 9)
- languag e suppor (Halaman: 9)
- finto ai (Halaman: 9, 11)
- systems integ rated (Halaman: 9)
- mark ed increase (Halaman: 9)
- final subjects (Halaman: 9)
- c hosen based (Halaman: 9)
- finto ai w (Halaman: 10)
- w w w (Halaman: 10)
- positories maintained b (Halaman: 10)
- v arsta system (Halaman: 10)
- pipeline whic h (Halaman: 10)
- heseus used b (Halaman: 10)
- annif prototype system (Halaman: 10)
- uploading ser vice (Halaman: 10)
- uploading pipeline osuv (Halaman: 10)
- first institutional user (Halaman: 10)
- k e k (Halaman: 10)
- student selected subjects (Halaman: 10)
- also add subjects (Halaman: 10)
- jyx r epositor (Halaman: 10)
- r j (Halaman: 10, 11)
- dspace softw (Halaman: 10)
- used b (Halaman: 10)
- annif w (Halaman: 10)
- v arsta (Halaman: 10)
- v u (Halaman: 10)
- u v (Halaman: 10)
- e u (Halaman: 10)
- finland maintains (Halaman: 10)
- v alidating (Halaman: 10)
- v aasa (Halaman: 10)
- h e (Halaman: 10)
- finnish book distributor (Halaman: 11)
- finland see figure (Halaman: 11)
- v er view (Halaman: 11)

- melinda union catalogue (Halaman: 11)
- produce metadata used (Halaman: 11)
- e n h (Halaman: 11)
- v aria (Halaman: 11)
- finto ai figure (Halaman: 11)
- bro wsed using (Halaman: 11)
- whic h (Halaman: 11)
- non fiction books (Halaman: 11)
- publication files (Halaman: 11)
- hey use (Halaman: 11)
- upcoming titles (Halaman: 11)
- libraries booksellers (Halaman: 11)
- integ rated (Halaman: 11)
- computers within (Halaman: 11)
- publishers enhances (Halaman: 11)
- audio visual material (Halaman: 12)
- dutc h libraries (Halaman: 12)
- ser v es (Halaman: 12)
- indexing e g (Halaman: 12)
- audio content based (Halaman: 12)
- online ser vice (Halaman: 12)
- w e present (Halaman: 12)
- subject based access (Halaman: 13)
- automatic subject indexing (Halaman: 13)
- c halleng es (Halaman: 14)
- indexing v ocabularies (Halaman: 14)
- subject indexing upload (Halaman: 14)
- positi v e (Halaman: 14)
- w ould also (Halaman: 14)
- w e thank (Halaman: 14)
- offer w e (Halaman: 14)
- api ser vices (Halaman: 14)
- w eb site (Halaman: 14)
- w l e (Halaman: 14)
- n g g (Halaman: 14)
- manik v ar (Halaman: 15)
- j uha piukkula (Halaman: 15)
- conf er ence (Halaman: 15)
- yso case j (Halaman: 15)
- infor mation science (Halaman: 15)
- f ramew ork (Halaman: 15)
- v erse (Halaman: 15)
- shallo w (Halaman: 15)
- echnolo g (Halaman: 15)
- osma suominen (Halaman: 15)
- extreme classification (Halaman: 15)
- tek oalykumppan uus (Halaman: 15)
- empirical r esults (Halaman: 16)
- utomatic subject indexing (Halaman: 16)
- elizabeth automatic indexing (Halaman: 16)
- uhlmann sandro (Halaman: 16)
- fusion arc hitectures (Halaman: 16)
- v ember (Halaman: 16)

```
- open source tool (Halaman: 16)
- open source (Halaman: 16)
- christin seifer (Halaman: 16)

3. Metode Word2Vec:
- Kata kunci: finto
   Tidak ada kata mirip yang ditemukan di dokumen.
- Kata kunci: ai
   > j (0.53, Halaman: 1, 3, 4, 5, 6, 9, 10, 11, 15, 16)
   > u (0.52, Halaman: 1, 3, 4, 5, 9, 10, 12, 13, 14, 16)
   > gi (0.52, Halaman: 1, 4, 7, 8, 14)
- Kata kunci: developing
   Tidak ada kata mirip yang ditemukan di dokumen.
- Kata kunci: implementing
   > implemented (0.69, Halaman: 3, 4, 14)
   > implementation (0.68, Halaman: 4)
```