Annals of Library and Information Studies
Vol. 72, December 2025, pp. 434-445
DOI: 10.56042/alis.v72i4.25677

Indigenous Knowledge Systems and Machine Learning: Evaluating the Suitability of Colon Classification

Parthasarathi Mukhopadhyay
Department of Library and Information Science, Kalyani University, West Bengal, India,
Email: psm@klyuniv.ac.in

Received: 21 April 2025; Accepted: 24 September 2025

This study explores the application of machine learning (ML) techniques to automate subject classification using S.R. Ranganathan's Colon Classification (CC - 6th edition), a faceted system rooted in traditional Indian knowledge frameworks. The research integrates the Colon Classification scheme with Annif, an open-source AI/ML-based subject indexing tool developed by the National Library of Finland to predict main class of a text corpus based on Colon Classification 6th edition. A curated dataset of nearly 100,000 English-language bibliographic records (with CC 6th notation) from the Indian National Bibliography (INB) was used for model training. The study evaluates the performance of several machine learning backends fastText, Omikuji (Bonsai), and Support Vector Classification (SVC) - as well as two ensemble models, including a neural network-based ensemble created through hyperparameter optimization. Key retrieval metrics such as F1@5 and NDCG were used to assess models efficacy. Among the tested models, the neural network ensemble achieved the highest scores, with F1@5 = 0.5873 and NDCG = 0.9473, showing strong accuracy in both prediction and ranking. The study demonstrates that machine learning can effectively support traditional classification systems when backed by well-structured data. Finally, through REST/API integration, the framework enables scalable classification, allowing real-time, automated processing of large bibliographic corpora. This work bridges indigenous classification logic with modern AI, contributing to more inclusive knowledge organization systems.

Keywords: Annif, Colon Classification, Machine learning, Neural Network model, Retrieval metrics

Introduction

The organization of knowledge has been a fundamental challenge throughout human intellectual history, with library classification systems serving as critical frameworks for accessing and retrieving information. Among these systems, the Colon Classification, developed by S.R. Ranganathan in 1933, stands as a landmark achievement in library classification. Its analytical-synthetic approach, which breaks down complex subjects into their constituent elements and then recombines them according to prescribed patterns or formula, offers remarkable flexibility and expressiveness for representing the multidimensional nature of knowledge.

Colon Classification (CC) system, influenced by Indian philosophical traditions, allows for more nuanced organization of knowledge[1]. However, the facet-analytic paradigm has been criticized for its lack of empirical basis and speculative ordering of knowledge[2]. The evolution of classification systems reflects ongoing efforts to improve knowledge organization and retrieval, balancing logical frameworks with the need for empirical grounding and adaptability to diverse knowledge domains[3]. This research study aims to explore the possibilities in application of machine learning tools and techniques (by using open source software and open datasets) for automated class number generation (presently only main class) based on Colon Classification 6th edition. It adopts an open source framework called Annif, developed by the National Library of Finland[4] and curated training datasets (bibliographic records with Colon Classification numbers) from Indian National Bibliography (INB), developed and maintained by the Central Reference Library, Ministry of Culture, Govt. of India.

Indian Knowledge Systems and Colon Classification

S.R. Ranganathan's **Colon Classification** reflects strong conceptual ties to traditional Indian knowledge systems[5]. While these influences may not always be direct, they clearly align with Indian methods of organizing knowledge[6]. Scholars like Singh and Satija have studied a range of historical Indian classification approaches from ancient Hindu, Jain, and Buddhist

systems to Islamic and colonial frameworks—highlighting how early library practices in India influenced later systems, including Ranganathan's[5]. Indian scriptures have long used structured methods to categorize vast bodies of knowledge, which likely inspired Ranganathan's faceted approach[7].

Among these traditional systems, the **Vedanga** divided knowledge into six areas: Shiksha (phonetics), Kalpa (rituals), Vyakarana (grammar), Nirukta (etymology), Chhanda (metrics), and Jyotisha (astronomy)[8]. The **Darshana** system classified Indian philosophy into six schools: Nyaya (logic), Vaisheshika (natural philosophy), Samkhya (metaphysics), Yoga (spiritual practice), Mimamsa (rituals), and Vedanta (spirituality)[9]. Additional frameworks include **Trivarga**—grouping knowledge into Dharma (duty), Artha (wealth), and Kama (pleasure)—and **Chaturvarga**, which adds Moksha (liberation)[10]. These systems appear in key scriptures such as the *Vedas*, *Upanishads*, *Mahabharata*, and *Manusmriti*, offering a holistic view of human life and learning. The concept of **Purusharthas**—Dharma, Artha, Kama, and Moksha—offers a balanced framework for worldly and spiritual life[11]. Dharma refers to moral duties, Artha to material well-being, Kama to sensory enjoyment, and Moksha to spiritual liberation[12]. This framework emphasizes harmony between practical living and ultimate realization, and has parallels with modern classification logics.

Ranganathan, widely regarded as the father of Library Science in India, adapted these foundational ideas to create the **Colon Classification** system[13]. His approach was based on **facets**, breaking down each subject into key components. He proposed five fundamental facets:

- **PM (Personality)**: the main entity or subject,
- **MN (Matter)**: the material involved,
- **EN (Energy)**: the process or action,
- **SN (Space)**: the location,
- **TN (Time)**: the period or chronology.

These facets are combined using connecting symbols to form flexible and precise classification codes. This method allows complex subjects to be accurately represented, a feature that traditional decimal-based systems struggle with.

Interestingly, Ranganathan's five facets show conceptual parallels with the ancient Indian Panchavidha system, which classifies knowledge by five types of relationships: Sambandha (relation), Vishesha (distinction), Samavaya (inherence), Samanya (generality), and Visheshana (particularity)[14]. For instance:

- **PM** relates to **Sambandha**, as both focus on core relationships,
- **MN** aligns with **Vishesha**, highlighting characteristics,
- **EN** connects with **Samavaya**, referring to inherent actions,
- **SN** reflects **Samanya**, addressing general context like space,
- **TN** links with **Visheshana**, dealing with specifics like time.

Although Ranganathan did not explicitly cite Panchavidha, the similarities suggest that Indian philosophical traditions may have influenced his thinking[14,15]. His Colon Classification system is therefore not only a modern, scientific tool for library organization but also a reflection of India's deep-rooted intellectual heritage[6]. Further research may clarify the extent of this influence, but the alignment between traditional and modern frameworks remains both compelling and meaningful.

Future Possibilities

In recent years, the emergence of artificial intelligence (AI) and machine learning (ML) has transformed numerous domains, from natural language processing to computer vision. These technologies have demonstrated unprecedented capabilities in pattern recognition, categorization & classification, and prediction tasks across diverse applications. However, the integration of these powerful computational approaches with indigenous knowledge organization systems like the Colon Classification remains an under-explored frontier with significant potential.

Previous research works had examined the Colon Classification's applicability to knowledge-driven AI implementations, particularly discussing use of Ranganathan's faceted schema to develop conceptual frameworks for digital libraries, focusing on knowledge abstraction and management[17] and using logic programming languages such as PROLOG to create inference engines that could replicate the classification process[18]. These approaches, while valuable, relied primarily on explicitly encoded rules and symbolic representations of knowledge. The

contemporary landscape of AI, however, has witnessed a paradigm shift toward data-driven methodologies[4,19,20], exemplified by large language models like GPT-4, BERT, and DeepSeek, which learn patterns and relationships from vast corpora of text rather than from explicitly programmed rules[21].

This shift presents both opportunities and challenges for the application of the Colon Classification system[22]. For example, a researcher analyzes the representation of FRBR entities in Colon Classification call numbers, finding correspondences between FRBR entities and Ranganathan's categories, and demonstrating the potential for FRBRized bibliographic arrangement using Colon Classification[23]. On one hand, modern AI/ML approaches can potentially identify complex patterns and relationships within bibliographic data that might elude rule-based systems. On the other hand, these methodologies typically require substantial labeled datasets for training, which have been notably scarce in the context of the Colon Classification. Furthermore, the lack of a standardized, machine-readable representation of the Colon Classification scheme in formats such as the Simple Knowledge Organization System (SKOS) has impeded its integration with contemporary information systems[24].

Two related research works are worth mentioning here that discuss the application of Colon Classification in Wikidata environment. The first paper presents CCLitBox, a Wikidata gadget that uses faceted classification and Linked Open Data (LOD) to automatically classify literary authors and works. CCLitBox reproduces the classification algorithm of Class O Literature of the Colon Classification and generates Colon Classification class numbers using freely available data in Wikidata[24]. The second paper, authored by a team of students from DLSc, Kalyani University explores the application of CCLitBox[25] in generating Colon Classification class numbers for Indian literary works (with interface in 22 Indian languages)[26]. The authors discuss how the faceted linked data elements of literary works in Wikidata can be joined together based on the rule base of the Colon Classification to synthesize Colon class numbers automatically (Figure-1).

This study addresses the next level of challenges through a comprehensive approach to automating book classification using the Colon Classification (6th edition). By applying data carpentry techniques, we have extracted a substantial dataset of 395,479 book records from the Indian National Bibliography (INB), each classified according to both the Dewey Decimal Classification (DDC) and the Colon Classification systems (6th edition). This dataset provides the foundation for training and evaluating data-driven AI/ML models capable of generating Colon Classification numbers for previously unclassified works.

The significance of this research extends beyond mere technical innovation. By bridging traditional knowledge organization principles with cutting-edge AI/ML methodologies, we aim to preserve and extend the intellectual legacy of the Colon Classification system while enhancing its practical utility in contemporary information environments. The development of an automated classification framework

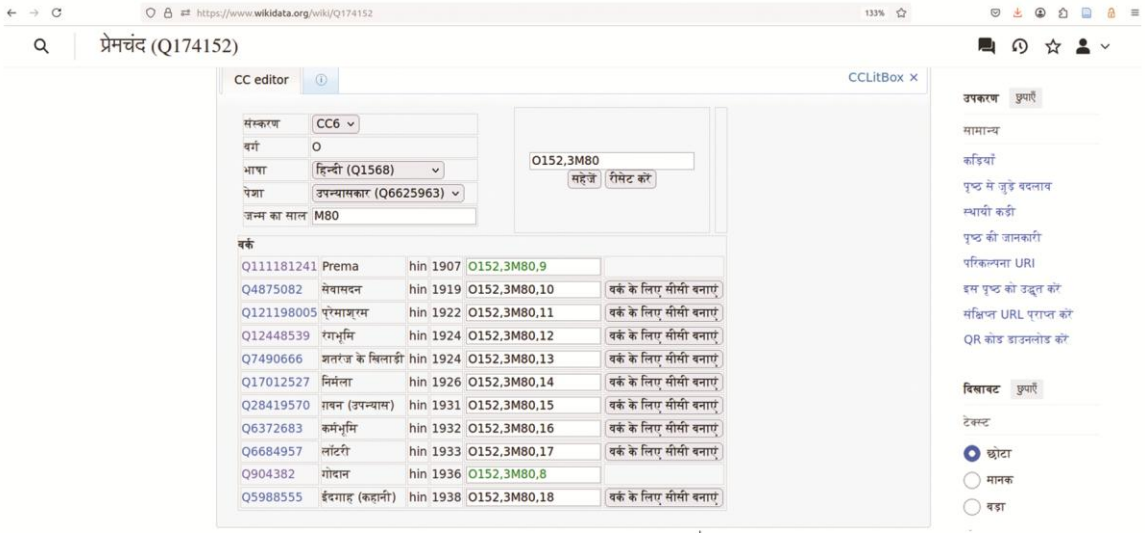

Fig. 1 — CC number In Wikidata based on colon classification 6th ed (Hindi interface)

that can generate Colon Classification numbers at scale (processing 10 records per second in our prototype system) represents a substantial advancement in bibliographic classification efficiency and accessibility.

In the following sections, this study reports methodology for dataset creation, model deployment, and evaluation, with particular attention to order-aware retrieval metrics such as Normalized Discounted Cumulative Gain (NDCG). It is then present preliminary results demonstrating the framework's effectiveness and discuss implications for library and information science practice, as well as directions for future research in this emerging interdisciplinary domain.

Objectives

The key objectives of this research study are as follows:

- To implement and set up an AI/ML framework, specifically Annif, for uploading the Colon Classification (CC) dataset using the BARTOC identifier scheme (http://bartoc.org/en/node/862).
- To create a structured bibliographic dataset for Annif, which includes MARC Bibliographic records containing titles (tag 245$a and 245$b), summary notes or descriptions (tag 520$a preferably or any 5xx tags/fields), and Colon Classification (CC) notation (tag 084$a).
- To analyze and compare the performance and effectiveness of various machine learning backends in Annif, particularly associative models (SVC, FastText, Omikuji) and Ensemble models (Simple and Neural Network) in determining main class for text corpus (title and summary note of a book) based on CC 6th edition.

Methodology

The methodology of this study involves using bibliographic data that has been labeled with CC notations and feeding it into an AI/ML framework called Annif. Annif is the only open-source tool in this field and was developed by the National Library of Finland. The goal is to find out how well indigenous classification systems like CC work with modern algorithm-based models, and can predict main class of an unknown document on the basis of Colon Classification 6th edition.. The research process includes the following steps, which are shown in the flow diagram (Figure 2).

Research Design

This study follows a step-by-step experimental framework to evaluate how effective and adaptable the Colon Classification (CC), especially its 6th edition vocabulary, is when used in machine learning (ML)-based knowledge organization systems (see Figure-2).

System Initialization and Vocabulary Deployment

The project begins with the installation and configuration of Annif, an automated subject indexing tool. The 6th edition of the Colon Classification system is deployed within Annif to serve as the primary controlled vocabulary for classification. The CC vocabulary, in this study, is using the following tabular format (Table-1) by using BARTOC (https://bartoc.org/) entity URI for CC (https://bartoc.org/en/node/862). Although this is not the ideal method, it is currently necessary because no linked open dataset (LOD) for Colon Classification is available till date.

BARTOC is a comprehensive, multilingual database of Knowledge Organization Systems (KOS) from all subjects and formats. It aims to make KOS more visible, searchable, and comparable. Founded in Switzerland and now hosted in Germany, BARTOC is open source and supported by ISKO (https://en.wikipedia.org/wiki/BARTOC).

AI/ML Framework Configuration

The machine learning framework is set up to support associative learning backends that are

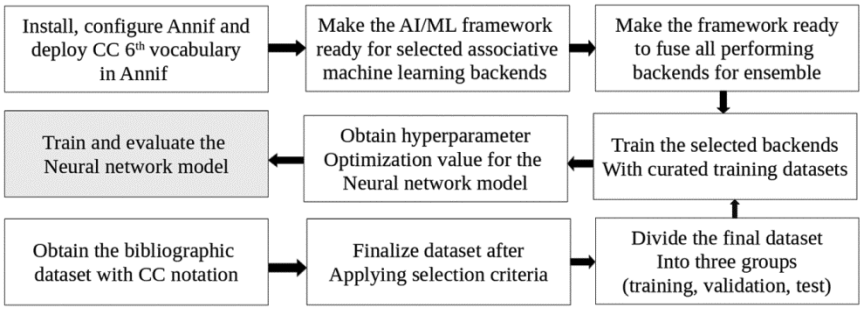

Fig. 2 — Research design

Table 1 — CC 6th vocabulary structure for Annif

uri	notation	label_en
http://bartoc.org/en/node/862/6/O	O1591,1M62:g	Main class: Literature
http://bartoc.org/en/node/862/6/R	R4	Main class: Religion
http://bartoc.org/en/node/862/6/R	R(Q84443)	Main class: Religion
http://bartoc.org/en/node/862/6/R	R635	Main class: Religion

compatible with Annif. This involves configuring environments for multiple machine learning, designed for text classification tasks. The development of Annif by the National Library of Finland marked a major step in making automated indexing and classification more accessible. As the first fully functional open-source tool, released under the Apache 2.0 license (https://annif.org/), Annif made AI and ML tools, along with various ML backends, widely available to library and information science (LIS) professionals[4,27,28]. Since its release, Annif has inspired numerous research projects and practical applications in libraries[18,29–31].

ML Backends: Regular and Fusion

Annif supports various machine learning backends for automated subject classification. These include MLLM, STWFSA, TF-IDF, fastText, Omikuji, SVC and deep neural networks via TensorFlow. Backends are categorized as *regular* (single algorithm) or *fusion* (ensemble). Regular backends are either *lexical* (e.g., string matching with thesauri) or *associative* (e.g., using word embeddings, contextual analysis, or deep learning). Associative backends are more flexible and can use transfer learning and pre-trained models for better accuracy[26,32]. The framework for this study is configured to enable ensemble learning by fusing the predictions of multiple performing backends, aiming to improve classification accuracy through model diversity.

Data Acquisition and Curation

A bibliographic dataset annotated with Colon Classification (CC) notations (6th edition) was obtained from the Central Reference Library, Kolkata through OAI/PMH-based harvesting. This dataset is part of the Indian National Bibliography (INB) and is available through the National Virtual Library of India project as an OAI/PMH-compatible dataset (https://inb.nvli.in/). As of April 30th, 2025, the full dataset contains 395,479 records. For this study, only English-language records with CC notations are selected for the initial round of experiments. This decision was made because the Annif framework and its associated natural language processing (NLP) tools (such as spaCy) do not yet support all Indian languages (INB covers 14 Indian languages including English). From the harvested INB dataset, a total of 115,916 English-language bibliographic records with CC notation (6th edition) were extracted for further processing. OpenRefine, an open-source data wrangling tool, is used to clean and standardize the dataset before it is used in the Annif framework. This step ensures the data is well-structured and suitable for machine learning, improving the overall quality and performance of the training process. A major drawback of the bibliographic records from the Central Reference Library, Kolkata is the lack of summary notes or any descriptive note fields. To address this gap, this study used OpenRefine to enhance the data by matching ISBNs and retrieving summary information through REST/API calls to the Google Books and Open Library projects. As a result, the final dataset - containing titles, summary notes, and CC notations - includes 98,776 bibliographic records formatted for use in the Annif framework (Figure-3).

Dataset Partitioning

The refined primary dataset (98,776 records) is split into three subsets—training, validation, and testing - to facilitate robust model training and evaluation. The division structure is as follows – training dataset (95,872 records ≈ 97%); validation dataset (1,899 ≈ 2%); and test dataset (1,005 ≈ 1%). The division of datasets is completely random by using GoKB plugin of OpenRefine and utmost care has been taken that there is no overlap in these three datasets to avoid data leakage as per the requirements of machine learning. This division is based on earlier research reports in machine learning literature, which generally suggests: Training sets typically range from 70-90% of data; Validation and test sets are often more balanced (5-15% each); but for large datasets (around 100K), smaller percentages for validation/testing can be reasonable[33–35].

Model Training

Selected ML backends are trained using the curated training dataset. This includes associative ML backends (namely fastText, Omikuji and SVC on the basis of their performance in a pilot study) as well as neural network-based ensemble model (combining fastText, Omikuji and SVC based on a weightage

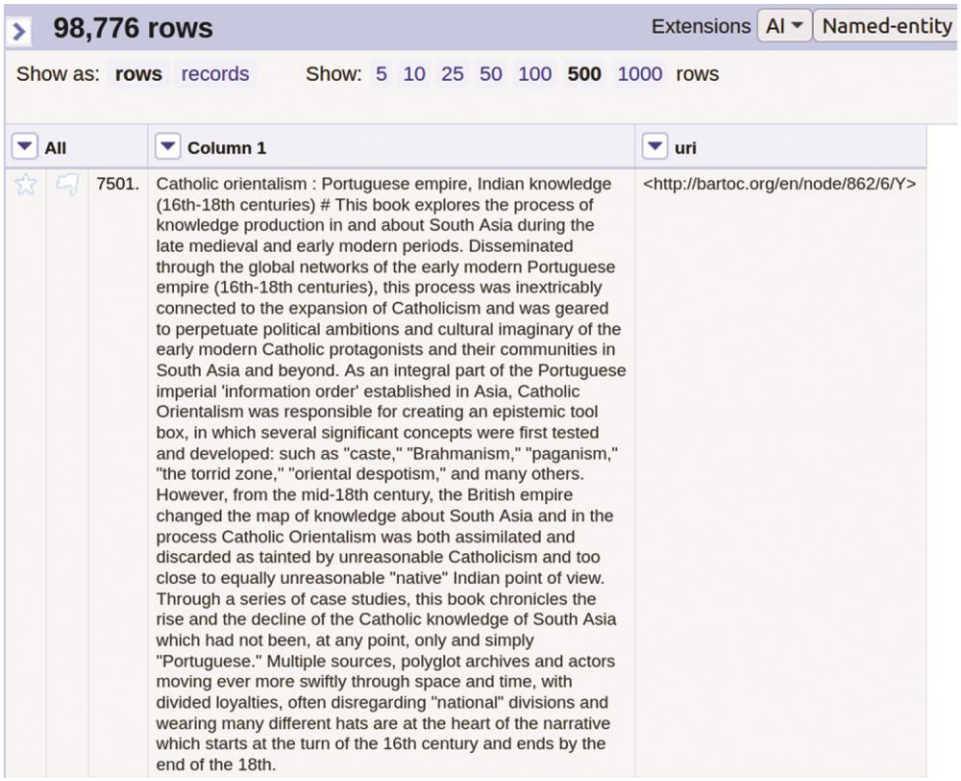

Fig. 3 — Structure of the primary dataset

formula obtained through hyperoptimization). This study curated nearly 100,000 records (98,776 to be exact) to build the final training dataset. Each selected machine learning backend in Annif - like fastText, Omikuji (Bonsai), SVC, and Neural Network has different data requirements to perform optimally. Finding the right amount of training data for each model typically involves manual trial and error, which is time-consuming. To solve this, the study introduces an automation script that tests different training data sizes and evaluates model performance using F1@5 and NDCG metrics on a fixed test set. The script takes inputs such as initial, minimum, and maximum training sizes, along with increment steps and the test dataset. It then runs multiple training and testing cycles to measure performance. The results are used to create learning curves that show how accuracy changes with more data, helping identify the point where adding more data no longer improves results significantly. This helps determine the optimal training size for each ML backend in a systematic, data-driven way.

For example, the learning curve analysis of the SVC backend shows a steady and consistent improvement in performance as the training data increases. At 2,500 records, the model starts with an F1@5 of 0.2471 and NDCG of 0.6578. Both metrics continue to rise smoothly as more data is used, reaching 0.3259 (F1@5) and 0.9324 (NDCG) at 95,000 records. Unlike some other backends, the SVC model does not show a sharp plateau early in the curve. Instead, it exhibits gradual performance gains across the full range of data sizes. However, the rate of improvement slows slightly after 60,000–70,000 records, indicating the beginning of a performance saturation point. NDCG consistently remains higher than F1@5, suggesting strong ranking capability in subject prediction. As evident from the given data in Table-2 and a chart based on that data (Figure-4), the SVC backend benefits from more data, with optimal efficiency likely achieved around 85,000–95,000 records, where the learning curve begins to flatten.

Similar learning curve experiments were conducted for the other selected machine learning backends in Annif - namely fastText, Omikuji, and Neural Network - to identify the optimal amount of training data required to achieve the best performance. For each backend, model performance was evaluated using the same test dataset and measured using the F1@5 and NDCG metrics. These experiments aimed to observe how each backend responds to increasing volumes of training data and to determine the data

threshold beyond which performance gains become minimal. The resulting curves help visualize efficiency trends and guide informed decisions on dataset size for effective training, ensuring each model operates at peak performance with minimal computational overhead.

Table 2 — Learning curve data for SVC backend

Data limit	F1@5	NDCG
2500	0.2471	0.6578
5000	0.2696	0.7279
10000	0.2874	0.7912
15000	0.2894	0.8051
20000	0.2991	0.8234
25000	0.3021	0.8344
30000	0.3052	0.8461
35000	0.3058	0.8571
40000	0.3095	0.8629
45000	0.3112	0.8755
50000	0.3129	0.8875
55000	0.3159	0.8923
60000	0.3176	0.899
65000	0.3199	0.9067
70000	0.3196	0.9139
75000	0.3209	0.9188
80000	0.3223	0.9273
85000	0.3256	0.9323
90000	0.3258	0.9324
95000	0.3259	0.9324

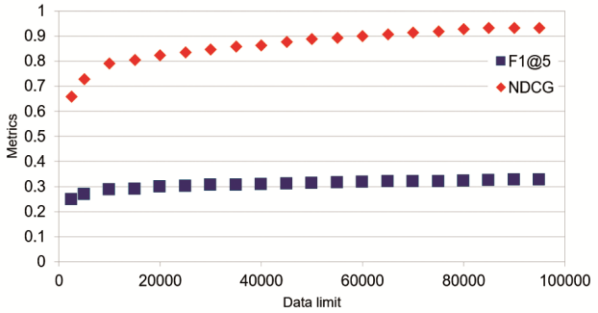

Fig. 4 — Learning curve illustration for SVC backend

Hyperparameter Optimization

A dedicated phase is conducted to determine optimal hyperparameter values for the neural network model to enhance its performance. Now that the selected machine learning backends - fastText, Omikuji (Bonsai), and SVC - have been trained with their optimal data limits, the next step is to combine them into a neural network-based ensemble model. This approach is motivated by two main goals: deploying the complementary strengths of each backend and enhancing overall performance. Each backend contributes differently, and their combined output can outperform individual models by compensating for one another's limitations. To effectively combine these models, it is important to determine how much weight each one should carry within the ensemble. Annif facilitates this through hyperparameter optimization, which tests various configurations using a validation dataset to identify the most effective combination. The first step is to create a basic ensemble by combining the selected backends without assigning any weights. Then, hyperparameter optimization is done using a validation dataset of 1,899 records, ideally with at least 200 trials[27]. The final weight values from this process are used to set up the neural network model (Table-3).

These results indicate that SVC and Omikuji contribute significantly to the ensemble's performance, while fastText plays a minimal role (Table-3). This weighted configuration forms the foundation of the final neural network ensemble,

Table 3 — Hyperparameter optimization in simple ensemble

Configuration – Simple Ensemble
[cc6-ens]
name= CC6 (Simple ensemble)
language=en
backend=ensemble
vocab=cc6
sources=cc6-fasttext,cc6-omikujiB,cc6-svc
limit=100
Command for hyperopt: *annif hyperopt --trials 250 cc6-ens /path/to/validation/data/val-1899.tsv*
Trail at 250th epoch:
Trial 249 finished with value: 0.9542264342308044 and parameters: {'cc6-fasttext': 0.002116532791281641, 'cc6-omikujiB': 0.46314442941269374, 'cc6-svc': 0.6248514623727459}. Best is trial 106 with value: 0.954701840877533.
Result:
Got best NDCG score 0.9547 with:
---
sources=cc6-fasttext:0.0010,cc6-omikujiB:0.4379,cc6-svc:0.5612

Table 4 — Deployed machine learning backends

Regular Backends	Lexical approach based backends	
	None	
	Associative approach based backends	
	fastText	A text classification algorithm derived from the fastText library, developed by Facebook AI Research (FAIR).
	Omikuji	It has two variants - Parabel and Bonsai, and both are advanced algorithms for multilabel classification based on tree-based machine learning algorithms. This study has selected Omikuji (Bonsai).
	SVC	The SVC is a realization of Linear Support Vector Classification.
Fusion Backends	Simple Ensemble	A technique for aggregating outcomes from various backends, devoid of algorithm-specific setup.
	Neural Network Ensemble	Implementation of neural network models like Keras and TensorFlow, and allows training and successive learning.

designed to achieve the best results by strategically integrating the strengths of the individual models. Next, the neural network model goes through the learning curve process to find out how many training records are needed to reach the best performance in terms of NDCG.

Model Evaluation

It has already been stated that a total of 5 machine learning backends have been selected based on a pilot study (Table-4). These models belong to two major groups – Regular (Associative) and Fusion (Ensemble). Fusion backends can combine more than one machine learning backends for obtaining optimum performance. The final neural network model is trained and evaluated against the validation and test datasets. The results are compared with other, and the ensemble output to gauge the overall performance.

The performance of a given backend in Annif is measured through a series of retrieval metrics. Retrieval metrics are essential for evaluating how well different machine learning backends perform in the Annif framework. Two key metrics used are F1@5 and NDCG, as they offer complementary insights into backend performance. A detail report on efficacy evaluation of the deployed machine learning backends in predicting main class for documents based CC 6th edition is discussed in the next section (Section 6).

Efficacy Evaluation

Retrieval metrics are used to measure how accurately machine learning backends perform automated subject indexing in Annif. This study focuses on two key metrics - F1@5 and NDCG - to evaluate and compare backend performance. F1@5 is an order-unaware metric that balances precision and recall, calculated using the top 5 predicted labels. It helps assess how accurate the backend is in suggesting relevant subjects. On the other hand, **NDCG** (Normalized Discounted Cumulative Gain) is an order-aware metric that measures how well the suggested labels are ranked, giving more importance to correct labels that appear higher in the list. Both metrics range from 0 (worst) to 1 (best), and together they provide a well-rounded view of the system's effectiveness. In automated classification/indexing, achieving scores above 0.5 for F1@5 and NDCG is considered a significant result due to the complexity of the task. These metrics are used extensively in this study to compare and improve the performance of different machine learning backends in Annif. Table-5 presents the performance of different backends based on a test dataset and evaluated using various retrieval metrics.

Among the various backends evaluated, Ensemble – Neural Network (**Ens-NN)** demonstrates superior performance, particularly in the context of F1@5 and NDCG**,** which are critical metrics for multi-label classification tasks. The F1@5 score of 0.5873 achieved by Ens-NN is significantly higher than those of the other models, indicating its strong ability to correctly predict the top 5 most relevant labels for each document. Additionally, Ens-NN maintains consistently high NDCG scores (0.9473 at @5 and @10)**,** reflecting its effectiveness in ranking relevant labels higher in the prediction list - an essential aspect for enhancing user-facing retrieval systems. Furthermore, since the classification task often involves assigning a single most appropriate class number to each document, Precision@1 becomes particularly important**.** Ens-NN scores 0.8924 in Precision@1**,** which is competitive with other top-performing models like Omikuji (Bonsai), suggesting that it reliably places the correct label at the top of its

Table 5 — Measuring efficacy of the deployed machine learning backends

Parameters	fastText	OmikujiB	SVC	Ens-Simple	Ens-NN
Precision (doc avg):	0.1112	0.0987	0.098	0.0996	0.4564
Recall (doc avg):	0.996	0.9869	0.9799	0.996	0.9859
F1 score (doc avg):	0.1972	0.1794	0.1782	0.1811	0.5852
Precision (subj avg):	0	0	0	0	0
Recall (subj avg):	0.0001	0.0001	0.0001	0.0001	0.0001
F1 score (subj avg):	0	0	0	0	0.0001
Precision (weighted subj avg):	0.1434	0.1693	0.1688	0.1616	0.3828
Recall (weighted subj avg):	0.996	0.9869	0.9799	0.996	0.9859
F1 score (weighted subj avg):	0.2449	0.28	0.2787	0.27	0.5401
Precision (microavg):	0.1043	0.0987	0.0987	0.0996	0.3448
Recall (microavg):	0.996	0.9869	0.9799	0.996	0.9859
F1 score (microavg):	0.1889	0.1794	0.1793	0.1811	0.5109
F1@5:	**0.3319**	**0.3273**	**0.3256**	**0.3307**	**0.5873**
NDCG:	**0.914**	**0.9502**	**0.9323**	**0.9463**	**0.9473**
NDCG@5:	0.9088	0.9485	0.9313	0.945	0.9473
NDCG@10:	0.914	0.9502	0.9323	0.9463	0.9473
Precision@1:	**0.8109**	**0.9014**	**0.8652**	**0.8763**	**0.8924**
Precision@3:	0.3208	0.3246	0.3223	0.3266	0.4854
Precision@5:	0.2006	0.1964	0.1954	0.1984	0.4578
True positives:	990	981	974	990	980
False positives:	8498	8959	8896	8950	1862
False negatives:	4	13	20	4	14
Documents evaluated:	994	994	994	994	994

predictions. This is crucial in practical scenarios where only the top prediction is used for classification, reinforcing the backend's applicability in real-world library classification systems.

Access

After completing the training and learning curve experiments for all selected machine learning backends - including both regular and ensemble models - Annif is ready to suggest CC (6th edition) based class numbers for new, unseen text (title and summary notes from books and other documents). The system offers three main methods to access these predictions:

Command-Line Interface (CLI)

This method is useful for development and testing. However, it reloads the model for each query, which slows down the process, especially for complex models like neural networks. Therefore, it is not recommended for large-scale or production use, but remains helpful for quick manual checks (Figure-5).

Web Interface

Annif also provides a user-friendly browser-based interface. Users can select a trained project, enter input text, and instantly receive subject suggestions. The system displays the results along with linked vocabulary terms (here CC 6th ed and BARTOC URI). This interface (Figure-6) can be deployed using a web server setup (e.g., Apache with WSGI) for persistent operation, ensuring fast response times by keeping the models loaded in memory.

REST/API Access

This is the most powerful and scalable option, ideal for real-time and large-volume processing. Through a RESTful interface, Annif allows external systems to send text and retrieve subject suggestions using simple HTTP requests. The core endpoint, /projects/{project_id}/suggest, receives input in JSON format and returns predicted descriptors or class numbers (Figure-7).

The RESTful interface offers a strong programmatic access point for automating subject indexing and integrating Annif with other systems. As shown in Figure-7, a simple Python or Jython script can be used to fetch suggestions from an Annif server through a REST/API call, even within a local OpenRefine setup. For this project, the REST/API method is especially important**,** as it enables automated classification of large bibliographic corpora using the Colon Classification (6th edition)**.**

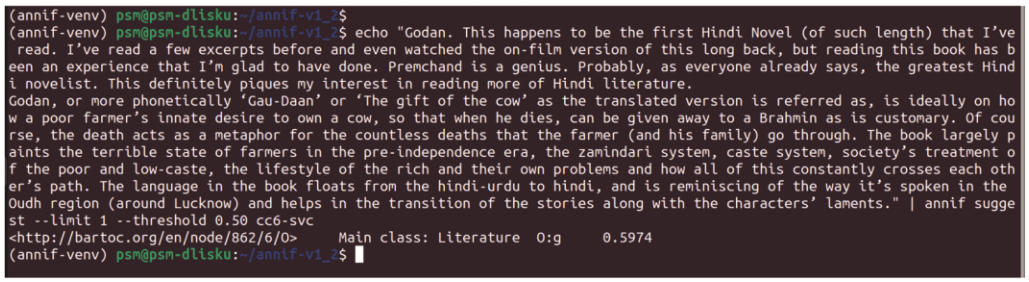

Fig. 5 — Prediction of CC (6th) main class from SVC backend in CLI

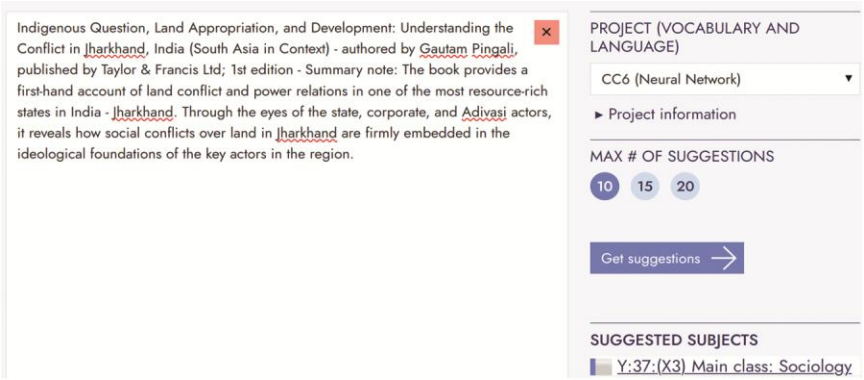

Fig. 6 — Prediction of CC (6th) main class from Neural Network backend in GUI

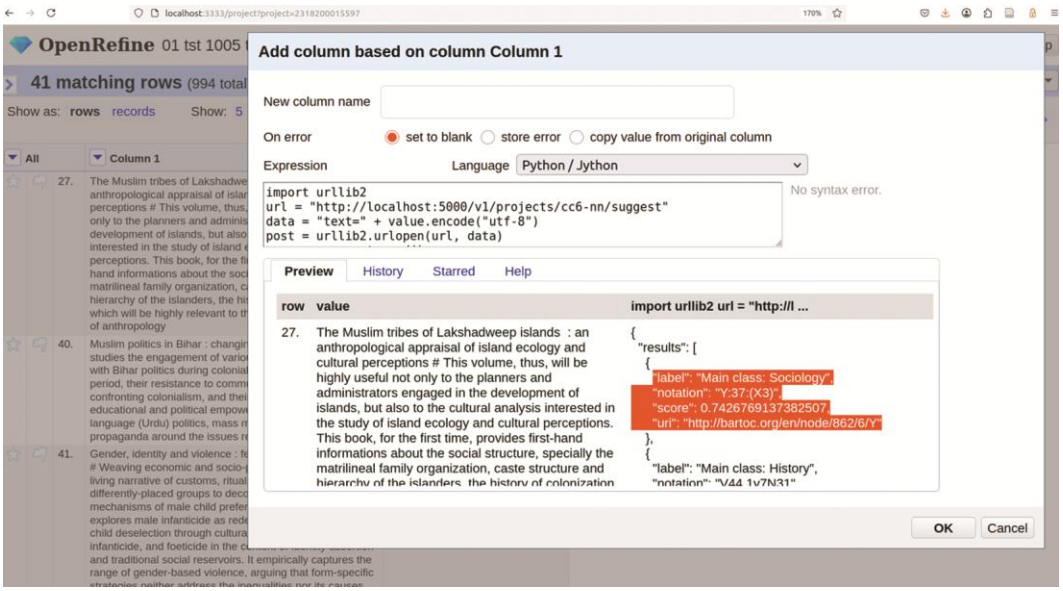

Fig. 7 — large-scale deployment in OpenRefine through REST/API

By integrating the API with tools like OpenRefine**,** it becomes possible to classify tens of thousands of records efficiently and consistently - something not feasible through manual interfaces. REST/API access ensures seamless integration with cataloguing systems, scalable deployment in institutional environments, and full automation of classification workflows, making it a key component of modern library infrastructure.

Conclusion

This study successfully meets its core objectives by implementing an AI/ML-based framework using the Annif platform to automate subject classification based on the Colon Classification (6th edition). A structured dataset of approximately 100,000 bibliographic records was curated, and several machine learning backends - including fastText, Omikuji (Bonsai), SVC, and neural network

ensembles - were trained and evaluated. The results demonstrate that machine learning can effectively replicate the logic of Colon Classification, with the neural network ensemble achieving notable performance, particularly an NDCG score of 0.9473, indicating strong prediction accuracy for unseen text.

However, the study faces certain limitations. The training dataset, though substantial, is limited to around 100K records, while an ideal training scenario would involve 500K or more[4,20]. Additionally, due to the unavailability of a complete Colon Classification 6th edition schedule in Linked Open Data (LOD) format, the system currently predicts only main class numbers, not deeper facets. The dataset is also constrained by its source - the Central Reference Library - which, despite its size (~400K records), includes multilingual entries not fully usable due to NLP (Natural Language processing) tool limitations. Most NLP tools like spaCy and Snowball offer limited support for Indian languages beyond Hindi and Bengali, restricting broader language integration.

Despite these challenges, the study reveals promising outcomes. The system demonstrates the ability to predict main class numbers for new texts with nearly 94% accuracy (NDCG), suggesting high potential for real-world application. REST/API-based integration enables large-scale deployment and seamless integration into resource description workflows. Moreover, the comparative performance of models like Omikuji (Bonsai), SVC, and the neural network ensemble across multiple retrieval metrics highlights their practical viability for classification tasks.

Looking ahead, the future of deploying Colon Classification[36] in machine learning environments appears highly promising. Ongoing efforts to publish the full Colon Classification schedule in LOD format via BARTOC will greatly enhance machine-readability and enable deeper class number predictions. With access to more comprehensive bibliographic datasets that include elements like summary notes, and improved support for Indian languages in NLP tools, the scalability and precision of such automated classification systems will only improve. This study serves as a foundational step toward that goal, bridging indigenous knowledge organization with modern AI capabilities.

References

1 Kwasnik B H, The role of classification in knowledge representation and discovery, *Libr Trends*, 48 (1) (1999) 22–47.

2 Hjørland B, Facet analysis: the logical approach to knowledge organization, *Inf Process Manag*, 49 (2) (2013) 545–557. https://doi.org/10.1016/j.ipm.2012.10.001

3 Bianchini C, Giusti L, and Gnoli C, The APUPA bell curve : Ranganathan's visual pattern for knowledge organization, *Cah Numér*, 13 (1) (2017) 49–68. https://doi.org/10.3166/lcn.13.1.49-68

4 Golub K, Suominen O, Mohammed A T, Aagaard H, and Osterman O, Automated dewey decimal classification of swedish library metadata using Annif software, *J Doc*, 80 (5) (2024) 1057–1079. https://doi.org/10.1108/JD-01-2022-0026

5 Satija M P and Singh J, Colon Classification: a requiem, *DESIDOC J Libr Inf Technol*, [Internet] 33 (2013) . Available from: https://api.semanticscholar.org/CorpusID:56069974

6 Gupta D K and Satija M P, Lights from the Ramayana in Ranganathan's philosophy, *Ann Libr Inf Stud*, 71 (1) (2024) 44–53. https://doi.org/10.56042/alis.v71i1.8961

7 Roe G, Challenging the control of knowledge in colonial India: political ideas in the work of S. R. Ranganathan, *Libr Inf Hist*, 26 (1) (2010) 18–32. https://doi.org/10.1179/175834909X12593371068342

8 Joglekar S, Window to ancient India: a tryst with ancient science & philosophy, First edition. (StoryMirror Infotech Pvt. Ltd.; Mumbai), 2023.

9. Sirswal D R, Reconsidering classical Indian thoughts, (Centre for Positive Philosophy and Interdisciplinary Studies (CPPIS), Pehowa (Kurukshetra), 2011.

10 Dalal R, The religions of India: a concise guide to nine major faiths, (Penguin Global), 2011.

11 Singh S and Satija M P, Indian classification systems: an analysis, *Libr Sci Slant Doc Inf Stud*, 35 (3) (1998) 165–178.

12 Mehta J M, Four objectives of human life: dharma - the right conduct : artha - the right wealth : kama - the right desires : moksha - the liberation (the right exit), (Hindoology Books; New Delhi), 2010.

13 Sharma A K, S R Ranganathan: Combining Library Science With Indian Values, *Libr Her*, 53 (3) (2015) 301. https://doi.org/10.5958/0976-2469.2015.00032.9

14 Sharma P S K, Treatment of Indian philosophy and Indian religions in colon classification, *Int Libr Rev*, 10 (3) (1978) 283–300. https://doi.org/10.1016/0020-7837(78)90015-8

15 Mazzocchi, Fulvio and Gnoli, Claudio, S.R. Ranganathan's PMEST categories: analyzing their philosophical background and cognitive function, *Inf Stud*, 16 (3) (2010) 133-147.

16 Adhikary M and Nandi A, Ideas of Ranganathan's classification theory pervaded by oriental philosophy, *SRELS J Inf Manag*, 40 (3) (2003) 275–284.

17 Suman A, From knowledge abstraction to management: using Ranganathan's faceted schema to develop conceptual frameworks for digital libraries, (Chandos Publishing; Oxford, UK), 2014, p. 1. (Chandos information professional series).

18 Panigrahi P and Prasad A R D, Facet sequence in analytico synthetic scheme: a study for developing an AI based automatic classification system, *Ann Libr Inf Stud*, 54 (1) (2007) 37–43.

19 Mukhopadhyay P, Machine learning and bibliographic data universe: assessing efficacy of backend algorithms in Annif through retrieval metrics, *SRELS J Inf Manag*, (2023) 39–48. https://doi.org/10.17821/srels/2023/v60i1/170891

20 Kerketta S and Mukhopadhyay P, From text corpus to Dewey number: designing a prototype for automated classification,

J Inf Knowl, 61 (6) (2024) 295–302. https://doi.org/10.17821/srels/2024/v61i6/171643

21. Ho C W C, Weber T, Fritze T, and Risse T, Towards multilingual LLM-based approaches for automatic Dewey decimal classification, In: International Conference on Theory and Practice of Digital Libraries 2024. p. 23–33.
22 Zhang S, Li Z, Liu X, Yang L, and Wang P, Arcmmlu: a library and information science benchmark for large language models, *Arxiv Prepr* Arxiv231118658, (2023) .
23 Bianchini C, Arrangement of FRBR entities in Colon classification call numbers, *Cat Classif Q*, 50 (5–7) (2012) 473–493. https://doi.org/10.1080/01639374.2012.679877
24 Asundi A, Domain specific categories and relations and their potential applications: a case study of two arrays of agriculture schedule of Colon classification, In: Categories, Contexts and Relations in Knowledge Organization 2012. p. 171–175.
25 Bianchini C, CCLitBox. A wikidata gadget to classify world literature, *J Inf Knowl*, 60 (3) (2023) 133–141. https://doi.org/10.17821/srels/2023/v60i3/171024
26 Halder D and Biswas M, Machine-generated colon class numbers: automatic classification of indian literary works in the wikidata environment, *J Inf Knowl*, 60 (3) (2023) 143–149. https://doi.org/10.17821/srels/2023/v60i3/171025
27 Suominen O, Annif: DIY automated subject indexing using multiple algorithms, *Liber Q J*, 29 (1) (2019) 1–25. https://doi.org/10.18352/lq.10285
28 Suominen O, Lehtinen M, and Inkinen J, Annif and Finto AI : developing and implementing automated subject indexing, *JLIS.It* 13 (1) (2022) 265-82. https://doi.org/10.4403/jlis.it-12740.
29 Ahmed M, Mukhopadhyay M, and Mukhopadhyay P, Automated knowledge organization AI ML based subject indexing system for libraries, *DESIDOC J Libr Inf Technol*, 43 (1) (2023) 45–54. https://doi.org/10.14429/djlit.43.01.18619
30 Hahn J, Semi-automated methods for BIBFRAME work entity description, *Cat Classif Q*, 59 (8) (2021) 853–867. https://doi.org/10.1080/01639374.2021.2014011
31 Oliver C, Leveraging KOS to extend our reach with automated processes, *Cat Classif Q*, 59 (8) (2021) 868–874. https://doi.org/10.1080/01639374.2021.2023717
32 Toepfer M and Seifert C, Fusion architectures for automatic subject indexing under concept drift: analysis and empirical results on short texts, *Int J Digit Libr*, 21 (2) (2020) 169–189. https://doi.org/10.1007/s00799-018-0240-3
33 Dobbin K K and Simon R M, Optimally splitting cases for training and testing high dimensional classifiers, *BMC Med Genomics*, 4 (1) (2011) 31. https://doi.org/10.1186/1755-8794-4-31
34 Joseph V R, Optimal ratio for data splitting, *Stat Anal Data Min ASA Data Sci J*, 15 (4) (2022) 531–538. https://doi.org/10.1002/sam.11583
35 Rácz A, Bajusz D, and Héberger K, Effect of dataset size and train/test split ratios in QSAR/QSPR multiclass classification, *Molecules*, 26 (4) (2021) 1111. https://doi.org/10.3390/molecules26041111
36 Raghavan K S, The colon classification: a few considerations on its future, *Ann Libr Inf Stud*, 62 (4) (2015) 231–238.