DESIDOC Journal of Library & Information Technology, Vol. 43, No. 1, January 2023, pp. 45-54, DOI : 10.14429/djlit.43.1.18619

Automated Knowledge Organisation: AI/ML-based Subject Indexing System for Libraries

Mustak Ahmed*, Mondrita Mukhopadhyay, and Parthasarathi Mukhopadhyay

Department of Library and Information Science, Kalyani University, West Bengal- 741 235, India
**E-mail: mustak.masu@gmail.com*

ABSTRACT

The research study as reported here is an attempt to explore the possibilities of an AI/ML-based semi-automated indexing system in a library setup to handle large volumes of documents. It uses the Python virtual environment to install and configure an open source AI environment (named Annif) to feed the LOD (Linked Open Data) dataset of Library of Congress Subject Headings (LCSH) as a standard KOS (Knowledge Organisation System). The framework deployed the Turtle format of LCSH after cleaning the file with Skosify, applied an array of backend algorithms (namely TF-IDF, Omikuji, and NN-Ensemble) to measure relative performance, and selected Snowball as an analyser. The training of Annif was conducted with a large set of bibliographic records populated with subject descriptors (MARC tag 650$a) and indexed by trained LIS professionals. The training dataset is first treated with MarcEdit to export it in a format suitable for OpenRefine, and then in OpenRefine it undergoes many steps to produce a bibliographic record set suitable to train Annif. The framework, after training, has been tested with a bibliographic dataset to measure indexing efficiencies, and finally, the automated indexing framework is integrated with data wrangling software (OpenRefine) to produce suggested headings on a mass scale. The entire framework is based on open-source software, open datasets, and open standards.

Keywords: Semi-automated subject indexing; LCSH; Annif; NDCG; OpenRefine; NN-Ensemble

1. INTRODUCTION

The magnitude of collections and the corresponding bibliographic records are accelerating in value, volume, and variety in libraries of all types and sizes all over the world. Libraries in India are no exception to this trend. As a result, workloads related to technical processing activities involving classification and subject indexing are increasing manifold. An AI/ML-based semi-automated indexing system may help professionals manage skill-oriented, labor-intensive, and time-consuming activities related to the processing of documents. The term "semi-automated" means that the system will predict and offer a set of suggestive subject descriptors on the basis of a given vocabulary system (like LCSH, MeSH, Agrovoc, UDC, and so on), but the final decision of selecting the appropriate descriptor(s) will be the privilege of a LIS professional.

So far, the AI/ML-based applications in the LIS domain are initiatives either of the libraries associated with large organisations (like the medical text indexer programme of NLM, US, and NASA's computer-supported indexing known as the MAI System) or commercial initiatives like PoolParty, TopBraid, Leiki, etc. For example, PoolParty's thesaurus management system can suggest descriptors from existing vocabularies against a text corpus. This closed-access era of AI and ML-based applications is coming to an end lately with the availability of different open-source analysers (like spaCy; and Simplemma analyser), NLP toolkits (like NLTK and Gensim), and different machine learning backends (like TF-IDF, Omikuji, fastText, Ensemble and so on).

The Annif open source framework for automated indexing, developed by the National Library of Finland, is an umbrella package that includes and combines different open source toolkits to predict subject descriptors or class numbers on the basis of KOSs (like LCSH, UDC, MeSH, Agrovoc, and so on) that are in wide use by libraries. In the Annif framework, it is quite feasible to use textual data in MARC 21 bibliographic format, like the content of the title (tag 245), summary notes (tag 520), etc., as input to automatically generate subject headings or class numbers with the help of linked open data (LOD)-based KOSs. In the case of journal datasets, a combination of title and abstract fields (short text corpus) or full-text papers (long text corpus) may act as input for the autogeneration of subject descriptors on the basis of a LOD-based KOS like MeSH, Agrovoc, the UNESCO thesaurus, and the like.

In view of the foregoing, this research study is an attempt to apply Annif as a framework to automatically generate subject descriptors for a set of MARC records (short text corpus) by using LOD-based LCSH as the backend KOS (https://id.loc.gov/authorities/). It also aims to integrate open-source data wrangling software with

Received : 31 October 2022, Revised : 23 December 2022
Accepted : 05 January 2023, Online published : 31 March 2023

the Annif framework for generating suggested subject descriptors for large MARC-formatted bibliographic datasets.

2. REVIEW OF LITERATURE AND OVERVIEW OF SYSTEMS

The TF-IDF language model, which is used by many AI/ML-based text predicting systems, is actually an information retrieval theory that dates back to the mid 1970s[1,2,3]. Subject indexing is the process of assigning relevant terms from a given standard vocabulary to express the major themes of the document under processing. Generally, multiple terms or descriptors are assigned by a trained professional to ensure retrieval of the document from many different perspectives. The very basic purpose of subject indexing is to support the retrieval of relevant documents for a given query. The ISO standard[4] says that there are three basic steps for subject indexing: 1) determine the subject content of the document; 2) decide the aspects of the content that should be represented; and 3) represent the subject content and its aspects by using the terms/descriptors from a controlled vocabulary[4].

In 2016, Golub, *et al*. stated that it is too soon to expect an automated subject indexing system to replace the complex subject indexing process defined in the ISO standard.[5] The major reason is that, generally, such automated systems are developed in laboratory conditions, which do not take complex real-life scenarios into consideration.

Exactly 5 years later[6], the same author reported the success of AI/ML-based indexing systems in libraries such as the OCLC Scorpion project to automatically generate DDC-based class numbers for books[7-8], automatic classification of web resources based on UDC[9], prediction of class numbers based on LCC[10], and formation of DDC-based class number and FAST subject headings for a set of MARC records from the Worldcat database[11], and so on.

In the domain of LIS, there are two schools of thought as far as the success of automated subject indexing is concerned. A group of experimental researchers think that such systems have huge potential and can help LIS professionals process large volumes of metadata and full-text objects effectively[12-16]. The other group of researchers is of the opinion that a computer-assisted indexing system or semi-automated indexing system will be much more useful in considering the complexities of subject indexing[17-19]. This research study is in agreement with the moderate group that a computer-assisted human indexing system is possibly the logical solution considering the state-of-the-art of AI and ML-based indexing tools and the associated complexities of subject indexing processes. Another major issue with automated indexing systems is the performance comparison between manual indexing and automated indexing. Lancaster opined in 2003 that the framework for measuring the efficiency of an automated indexing system is seriously flawed[20]. The reason for such comments is possibly due to the fact that the very concept of "relevance" is subjective in nature[21]. Several researchers proposed a new conceptual framework for relevance based on identifying two related factors:relevance "as is" and relevance "as determined"[18,22]. The normalised discounted cumulative gain (NDCG) retrieval metric may provide a solution for ranking subject descriptors in an automated subject indexing system[23].

The large projects in the direction of automated classification and subject descriptors are quite well known in the domain of LIS, like the initiatives of NASA[24], the National Library of Medicine, US[25], the National Agricultural Library of the US[26], and the German National Library[27]. However, all of these systems are based on in-house expertise, and the software and tools that are in use are never made available for applications outside of these library premises. The first ever fully functional open source tool, named Annif, was developed and made available under the Apache 2.0 license by the National Library of Finland (https://github.com/NatLibFi/Annif). Osma Suominen, the project leader of Annif, along with his team, have published a few research studies describing methodologies of automated indexing by using the tool[28,29], and a few researchers have reported the use of Annif in different other projects.[30-32]

3. OBJECTIVES AND RESEARCH QUESTIONS

The broad objectives and the corresponding research questions (RQs) of this study are enumerated in Table-0.

The tasks to accomplish these specified goals can be divided into four categories: a) designing the framework by selecting appropriate analyser and backend algorithm; b) preparing the backend KOS, here LCSH in SKOS format, to feed into the framework; c) developing a sizable training dataset; and d) testing and measuring the model framework's indexing efficacy using a test dataset.

4. METHODOLOGY

The foregoing section ends with a panoramic view of the tasks required to fulfill the stated goals of this research study, but a close analysis of the objectives and RQs reveals that the activities related to accomplishing the objectives may broadly be grouped under the following steps: (1) obtain LCSH as an LOD dataset and fine-tune the obtained file's SKOS structure as required by the Annif framework; (2) collect as many MARC formatted bibliographic records as possible, preferably with subject descriptors (tag 650 $a) and summary notes (tag 520 $a); (3) merge MARC files to generate a single consolidated file and then export the file in a format suitable for OpenRefine data wrangling software by using the MarcEdit tool; (4) reconcile the subject descriptors (present in tag 650$a) as available in the file by using the linked data service of LCSH to fetch and extract the subject URIs of the descriptors as Annif needs the file in the form – Text corpus (may be a combined field of title and summary note) and the URIs of the assigned subject descriptors (in the case of more than one URI for a given text corpus, the URIs must be separated by

Table 0: Objectives and corresponding RQs

	Objectives	Associated Research Questions (RQs)
A	To load LCSH as LOD dataset inside the AI/ML framework.	How to setup the Annif framework with associated tools? What RDF serialisation format of LCSH is suitable for loading the vocabulary?
B	To prepare a large dataset of bibliographic records covering many disciplines, preferably with summary note, in a format suitable for Annif.	How to obtain MARC records from different sources and to join them in a single MARC file? How to convert the consolidated MARC file into a form suitable for Annif framework that is - Title-Note | <URI of LCSH descriptor>?
C	To prepare a test dataset to check accuracy of subject descriptors as suggested by Annif, and to design a mechanism for large-scale use of the framework.	How to measure indexing efficiency of Annif in terms of different retrieval metrics against a test dataset? How to integrate OpenRefine with the Annif framework to capture suggested descriptors for large number of documents on the basis of text corpora generated by combining titles and summary notes?

a space); (5) load the SKOS-compliant vocabulary (here LCSH) generated in step 1 into the Annif framework; (6) train the framework with the curated MARC file generated in step 4; (7) measure the system's indexing efficiency using a set of appropriate retrieval metrics; and 8) test the system for large-scale subject descriptor production using a suitable script. The details of these tasks are discussed in depth under four headings in this section:

4.1 Building the Framework

Like most of the AI/ML systems, Annif (whose present stable release is version 0.59) also works in the Python virtual environment (version 3.8+ of Python). The basic Annif package includes backends like TF-IDF and components like TensorFlow and Gensim. Additionally, the framework may be powered by NLTK punctuation rules (punkt) and advanced backend algorithms like fastText Omikuji, Neural Network of Ensemble (NN-Ensemble), etc. The details of the components in the framework are given in Table 1.

In Annif, a project must include the following statements: 1) project id (e.g., [lcsh-tfidf-en]); 2) project name (e.g., name=LCSH TFIDF project); 3) language of the text corpora (e.g., language=en); 4) backend algorithm (e.g., backend=tfidf); 5) vocabulary id (vocab=lcsh-en); and 6) name of analyser (e.g., analyser=snowball(english)). The framework can accommodate any number of projects. The project details in Annif may be displayed through the command - *annif show-project <project id>*. The backend algorithms in Annif may be TF-IDF, MLLM, Omikuji (Parabel and Bonsai), Ensemble (Simple, PAV, and Neural Network). The selection of an appropriate backend algorithm and a suitable analyser on the basis of the nature of bibliographic data are crucial decisions for the efficient and expected performance of the framework. However, the TF-IDF backend is an easy start as algorithm-specific configurations are not required here.

A broad-range comparative study of different algorithms on the basis of their performances in this research framework has been done in section 7. The selected analyser in the framework performs the tasks of tokenisation and normalisation of a text corpus along with stemming and lemmatization.[33,34] A detailed discussion of the backend algorithms and analysers of Annif is available in the software wiki (https://github.com/NatLibFi/Annif/wiki – Section Backends/Algorithms supported by Annif).

4.2 Developing the KOS Backend

The framework, thus prepared, needs a structured standard vocabulary to start with. In Annif, a standard vocabulary may be added in two ways: 1) feeding a SKOS-compliant vocabulary in any common RDF serialisation format (like RDF/XML (.xml), N-Triple (.nt), Turtle (.ttl), etc.); or 2) using a vocabulary file in a UTF-8 encoded TSV file, where the first column contains a subject URI and the second column includes

Table 1. Components of the framework

Target	Tools	Purpose
Framework for automated subject indexing	Python Virtual Environment (Python 3.8.13 version & PIP)	Requires to install and configure Python virtual environment with Python (3.8+) and PIP (22.0+) for Annif and its associated components.
	Annif (version 0.60) (with NLP and ML tools) https://github.com/NatLibFi/Annif/	The main component of the framework available as open source tool including components like TensorFlow and Gensim.
	Backends and Tools (Annif virtual environment will select appropriate versions)	NLTK model for punctuation rules (punkt); and backend models like fastText; Omikuji; and NN-Ensemble.

the corresponding label (subject descriptor). As most of the standard vocabularies that are in use in the LIS domain, like LCSH, MeSH, Agrovoc, and the UNESCO thesaurus, are available as SKOS-compliant KOS, the additional workloads related to preparing a TSV file in the given format may be avoided. This research study has deployed the TTL format of LCSH available from the Library of Congress (https://id.loc.gov/). The TTL format is preferred over the other SKOS/RDF formats because of its comparatively smaller download size, and the TTL format is easier to read by Annif. The LCSH LOD dataset as obtained requires cleaning to eliminate redundancy and other limitations. Therefore, it is cleaned by using a tool developed by the National Library of Finland named Skosify, and then the cleaned file is validated through the utility RDF Validator as developed by the W3C (Table 2). The command to feed the ready vocabulary inside the Annif framework is - *annif load-vocab <path/to/TTL file>*.

Table 2. Dataset and tools for preparing the KOS

Target	Dataset & Tools	Process & Purpose
Vocabulary dataset preparation	Linked Open Dataset for LCSH (in TTL format)	The SKOS-compliant LCSH in TTL format is deployed to develop the backend KOS for the framework.
	Skosify (github.com/NatLibFi/Skosify)	It converts the TTL file of LCSH into a clean SKOS file by eliminating redundancy, removing duplicates and other inconsistencies automatically.
	RDF Validator (w3.org/RDF/Validator/)	It performs the role of a strict validator of the file generated through Skosify before further use.

4.3 Preparing the Training Dataset

After the vocabulary feeds inside the framework, it requires training to ensure efficient prediction of subject descriptors against a text corpus (usually a combination of the title of a document and its abstract or summary note, separated by space or any other character – here the double pipe || sign). The framework requires a training dataset as a TSV file with the first column containing a text corpus and the second column containing the URIs of the subject descriptors from LCSH (to be enclosed with angular brackets <>). The descriptors are assigned by trained LIS professionals from LCSH. In the case of more than one descriptor, the URIs of the descriptors must be separated by one space (Table 3).

This research study applies two large MARC-formatted bibliographic datasets with subject descriptors (tag 650$a) assigned by LIS professionals on the basis of LCSH. These are from: 1) the Springer-Nature metadata download facility for librarians (http://metadata.springernature.com/), which offers free MARC record downloads under 22 broad subject groups for the period from 2005 to 2023; and 2) the MARC download service of the University of Michigan library (http://www.lib.umich.edu/files/umich_bib.marc.gz). These two sources (datasets are available under ODbL licensing) are used to collect approximately 10 lakhs (1 million) of high-quality MARC records with titles, summary notes, and subject descriptors (tags 245, 520, and 650, respectively). The MARC files as obtained are then split into smaller MARC files (.mrc), each containing around 1.25 lakh records (0.125 million), for the sake of easy handling, by utilising the MARCSplit option in the tool MARCEdit. Each of these MARC files is then exported into an OpenRefine-compatible format (.tsv) from MARCEdit for further processing. OpenRefine, an open source data wrangling software, allows us to select only the rows having certain tags, for example, in this case, tag 650 data. This sort of selective display is needed to reconcile human-indexed subject descriptors (the content of tag 650, after converting it into raw strings without subfield code) for matching with the

Table 3. Structure the training dataset required for the framework

Text corpora	LCSH subject descriptors (URI)
ESR Spectroscopy for Life Science Applications: An Introduction || This book introduces the audience with basic theoretical and experimental aspects of Electron Spin Resonance (ESR) Spectroscopy. It further talks about ESR spectroscopy applications in Healthcare & Pharmaceutical Science, Paleontology & Geochronology and Food Science.	<http://id.loc.gov/authorities/subjects/sh85083097><http://id.loc.gov/authorities/subjects/sh89006496><http://id.loc.gov/authorities/subjects/sh85110774><http://id.loc.gov/authorities/subjects/sh85126423><http://id.loc.gov/authorities/subjects/sh85023026><http://id.loc.gov/authorities/subjects/sh85093451><http://id.loc.gov/authorities/subjects/sh85110774><http://id.loc.gov/authorities/subjects/sh85126423><http://id.loc.gov/authorities/subjects/sh85093451>

corresponding subject descriptor of LCSH (see Fig. 1).

After the reconciliation process is over, it is an easy task to extract URIs of the subject descriptors that are matched with the LCSH dataset through a GREL (General Refine Expression Language) *cell.recon.match.id to extract descriptor URIs f*or the purpose of developing the training dataset in the required format.

Finally, the dataset in the format suitable for the automated indexing framework is made ready through a series of other operations in the OpenRefine software. The final dataset contains 6,00,000 (0.6 million) bibliographic records (out of 1 million of gathered data) with titles (tag 245), notes (block 5xx, especially tag 520), and assigned descriptors (by trained professionals) in tag 650 that are exactly matched from the LCSH reconciliation service. The success of an automated indexing framework largely depends on the quality of the training dataset. Ideally, a training dataset should touch on all the descriptors available in a given vocabulary system. This research study has sincerely attempted to cover almost every broad subject through 6,00,000 (0.6 million) MARC-formatted bibliographic records with titles and notes as text corpora and URIs of subject descriptors from LCSH assigned by trained librarians. A sample dataset in the final format ready for use in the training process is illustrated in (Fig. 2).

4.4 Measuring Prediction Efficiencies

The complex process of preparing a training dataset is valuable only if the prediction accuracy of an automated subject indexing system is as per expectations. There is an array of retrieval metrics, each with its own advantages and disadvantages, to measure the efficiencies of an automated subject indexing system with scores. Annif supports many retrieval metrics like precision, recall, F1 score, F1@5, and normalised discounted cumulative gain (NDCG) for measuring the accuracy of subject prediction. Some of these are order-unaware metrics (like recall, precision, F1 score, etc.) and do not take into consideration the order of the retrieved results set[35,36]. The order-aware retrieval metrics (graded relevance) are

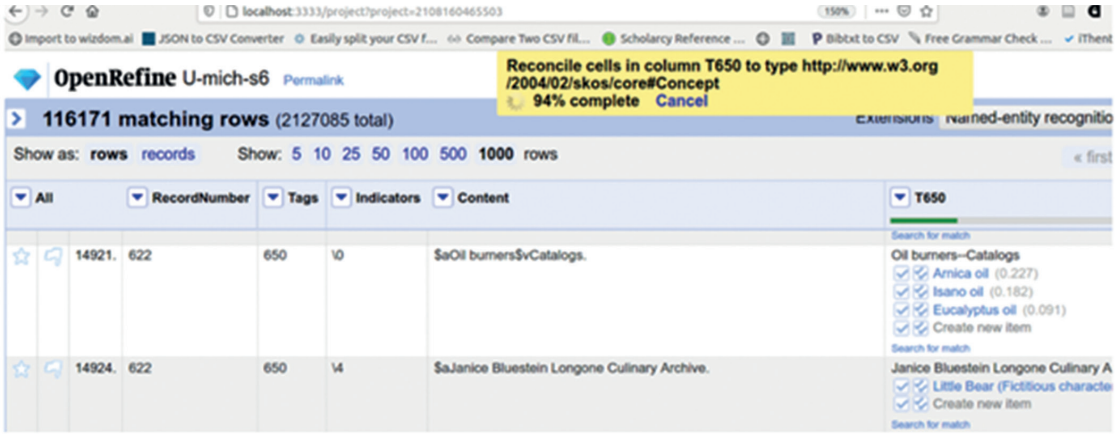

Figure 1. Reconciliation of human-indexed descriptors with LCSH.

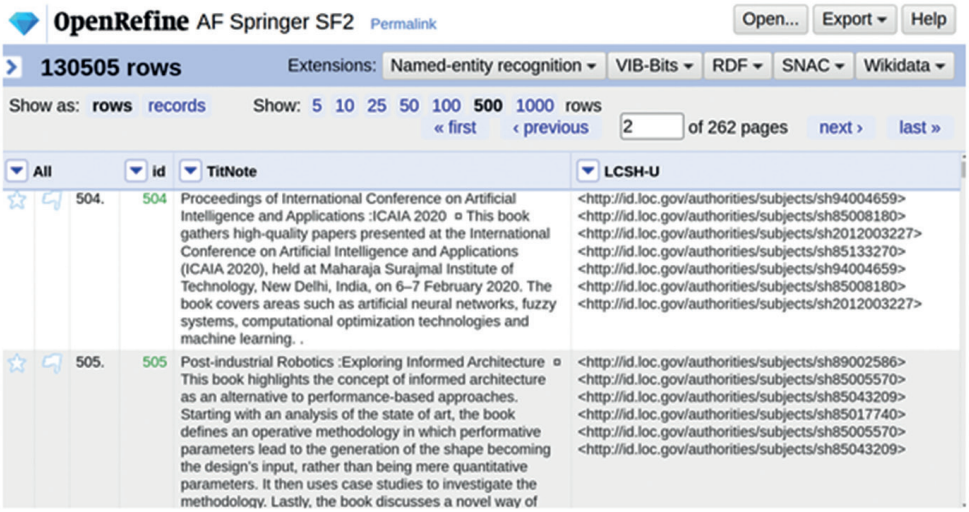

Figure 2. Final structure of the training dataset in OpenRefine.

cumulative gain (CG), discounted cumulative gain (DCG), and normalised discounted cumulative gain (NDCG). Annif ranks predicted subject descriptors against a given text corpus by accuracy scores (specific to the backend algorithm is in use). Accuracy scores vary with a range between 0 and 1 (see Fig. 3). The Figure 3 shows the subject prediction results from a command line call -annif suggest lcsh-tfidf-en-*threshold 0.3* for a given text corpus (lcsh-tfidf-en is the project name here), where algorithm-specific accuracy scores are equal to or greater than 0.3.

5. ACCESSING THE FRAMEWORK

The automated indexing framework can be utilised, on the basis of a given purpose, in three different ways: 1) from the command prompt (see Fig. 3); 2)

curl -X POST --header 'Content-Type: application/ x-www-form-urlencoded' --header 'Accept: application/json' -d '**text=Agriculture Value Chain - Challenges and Trends in Academia and Industry&limit=1&threshold=0.3**' 'http://127.0.0.1:5000/v1/projects/ lcsh-tfidf-en/suggest'	{ "results": [{ **"label": "Agriculture--Environmental aspects--Congresses",** "notation": null, **"score": 0.3890317678451538,** "uri": "http://id.lo c.gov/ authorities/subjects/ sh2009114224" }] }

REST/API call mechanism.

```
(annif-venv) roshni@roshni-HP-Pavilion-Laptop-14-dv0xxx:~/annif$ echo "Chemical Profiles of Selected
Jordanian Foods  ¤ This book provides an overview of various traditional Middle Eastern food products
 and beverages and investigates their chemical, microbial and physical profiles. The authors selected
 5 traditional Jordanian products that represent the region's historical foods and beverages and exam
ined them with a focus on their chemical composition and preparation." | annif suggest lcsh-tfidf-en
--threshold 0.3
<http://id.loc.gov/authorities/subjects/sh85050202>     Food--Microbiology      0.4896392226219177
<http://id.loc.gov/authorities/subjects/sh2018000786>   Food science    0.48154234886169434
<http://id.loc.gov/authorities/subjects/sh85050185>     Food--Analysis  0.4595916271209717
<http://id.loc.gov/authorities/subjects/sh2009007706>   Food security   0.44948524236679077
<http://id.loc.gov/authorities/subjects/sh85050184>     Food    0.4428102970123291
<http://id.loc.gov/authorities/subjects/sh2001007789>   Food--Safety measures   0.4409063160419464
<http://id.loc.gov/authorities/subjects/sh85022913>     Chemical industry       0.4367563724517822
<http://id.loc.gov/authorities/subjects/sh85093451>     Nutrition       0.42807021737098694
<http://id.loc.gov/authorities/subjects/sh85065899>     Industrial microbiology 0.412789911031723
<http://id.loc.gov/authorities/subjects/sh85022986>     Chemistry       0.4123398959636688
```

Figure 3. Subject descriptor prediction in command line along with accuracy scores.

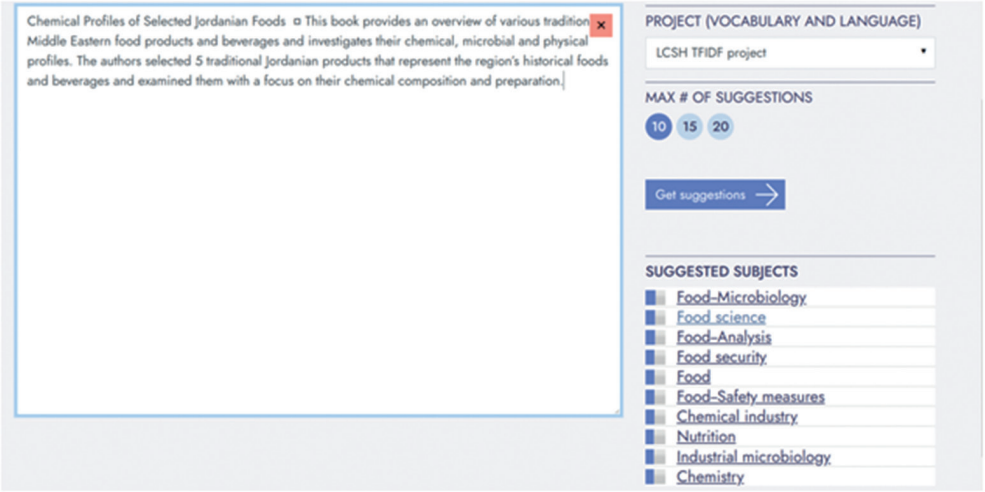

Figure 4. The framework in Web UI.

through a Web UI micro-service running at port 5000 (see Fig.4 and 3) over a REST/API call (see Textbox 1). The command prompt-based access is actually meant for testing and not for large-scale use, as it loads the model every time a query is triggered. The most appropriate way to get suggestions for descriptors for a large text corpus is through REST/API call-based access. The most important REST/API endpoint available presently is - /projects/project_id/suggest, for suggesting subject descriptors from the KOS in use (here, LCSH) against a given text corpus in JSON format.

The framework includes a Web UI as a micro-service to test the model. It allows the end user to select a project from the drop-down list and to add a text corpus in a text-box (here, Fig. 4 shows the LCSH TFIDF project, though this study has also explored other backends like Omikuji and NN-Ensemble; see Section 7). The "Get suggestions" button (Fig. 4) will predict a list of subject descriptors from the vocabulary in use. Each predicted subject heading is hyperlinked through the URI with the vocabulary (here LCSH).

6. LARGE-SCALE PREDICTION

This research study has also achieved the goal stated in objective 3, i.e., to develop a mechanism for large-scale use of the framework. It integrates OpenRefine, where text corpora are stored, with the Annif framework through a Python script[37,38].

The script is required as Annif does not yet support GET requests but only the POST method to respond to a REST/API call. The Python script in OpenRefine (see Fig. 5) can fetch suggested subject descriptors for a text corpus (column 1 in Fig. 5) from the framework on the basis of REST/API calls (POST request) in real-quick time. It has been observed that this mechanism can fetch subject descriptors for 500+ bibliographic records in less than a minute, more precisely at the rate of 10 records per second (tested in an i7 processor based laptop with 16GB RAM in the Ubuntu 22.04 OS platform).

7. HUMAN VS MACHINE: PERFORMANCE ANALYSIS

The process of subject indexing is a complex one, even for the same subject content, two LIS professionals may not assign the same descriptors. Studies show that when two LIS professionals index the same document by using the same vocabulary control device, only around one-third descriptors are identical.[29] Though the indexing process and results are highly subjective in nature, a group of researchers (scikit-learn.org) proposed many quantification methods and formulae to measure retrieval efficiencies.[39] This research study measures indexing efficiencies of human-assigned descriptors and machine-generated descriptors by using the '*eval*' command of Annif. The method involves the creation of a test bibliographic dataset of 926 book records with title and summary notes as text corpora and subject headings assigned by trained professionals (around 1 % of the training dataset). This test dataset kept separate from the training dataset, and therefore,

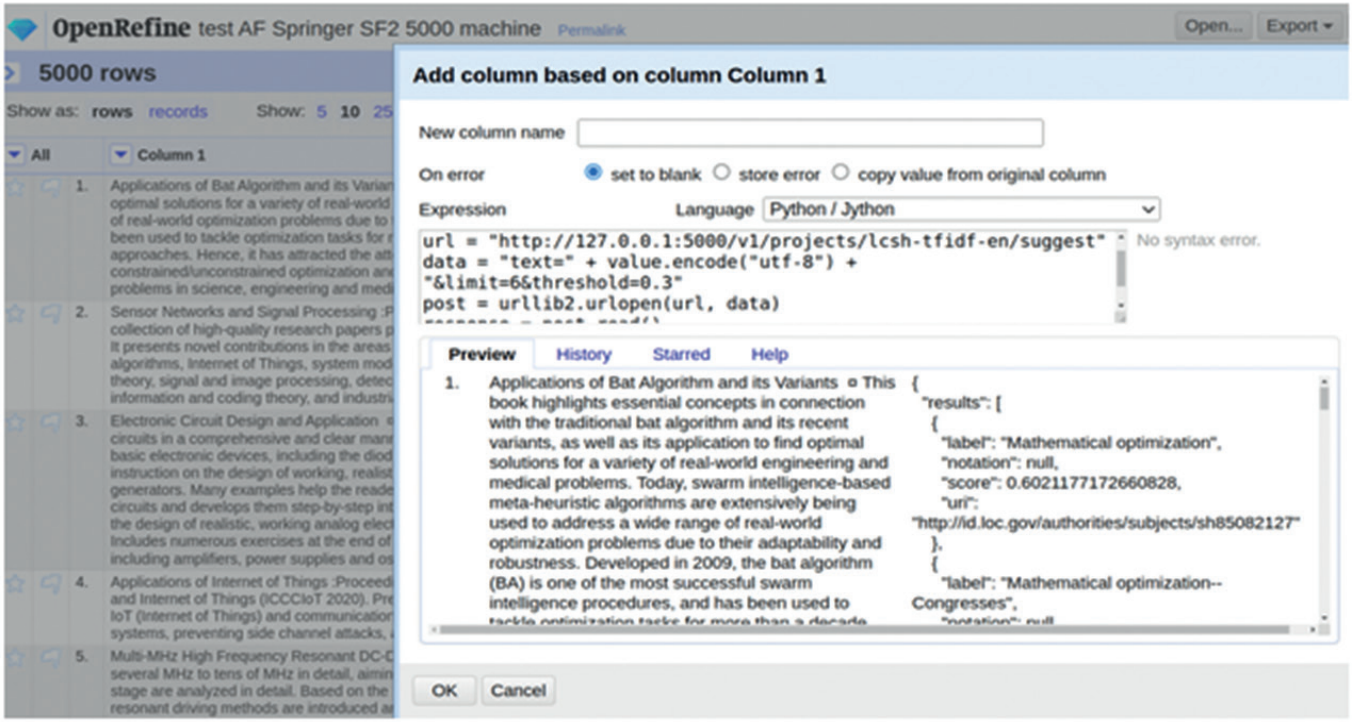

Figure 5. Large-scale suggestions from Annif in OpenRefine.

Table 4. Comparison of performances for different backends

Retrieval metrics	NN Ensemble	Omikuji-Bonsai	Omikuji-Parabel	TF-IDF
Precision (doc avg):	0.309631372347355	0.309179265658747	0.308747300215983	0.207019438444924
Recall (doc avg):	0.741756560574055	0.751288940840777	0.751240582072116	0.521869931724143
F1 score (doc avg):	0.420030408273821	0.420388112243803	0.420063371466736	0.284638884866966
Precision (subj avg):	0.0004590786637205	0.0004564161704149	0.00047478295913574	0.00040798721462671
Recall (subj avg):	0.0009228166100754	0.0009587395438364	0.00096739685770314	0.00077853950097507
F1 score (subj avg):	0.0005787470910970	0.0005897965061141	0.00060220552496157	0.00047667732083549
Precision (weighted avg):	0.35632779486119	0.361648804328963	0.366296986419734	0.379804098255177
Recall (weighted subj avg):	0.716708860759494	0.724810126582278	0.72379746835443	0.485316455696202
F1 score (weighted avg):	0.459910723814825	0.470426228610616	0.472921553934317	0.382491646668802
Precision (microavg):	0.309635786940829	0.309179265658747	0.308747300215983	0.207019438444924
Recall (microavg):	0.716708860759494	0.724810126582278	0.72379746835443	0.485316455696202
F1 score (microavg):	0.432444817841595	0.433459500378501	0.432853898561696	0.290234670704012
F1@5:	**0.49435852089581**	**0.506640417347443**	**0.506647255961193**	**0.30868245590009**
NDCG:	**0.699593259170681**	**0.711392219447817**	**0.71280011435324**	**0.4568922489481**
NDCG@5:	0.646700647008636	0.662096660163199	0.663606167279434	0.397237552565437
NDCG@10:	0.699795261420476	0.71161269988914	0.713003796056555	0.457071053777278
Precision@1:	0.748380129589633	0.769978401727862	0.777537796976242	0.461123110151188
Precision@3:	0.586393088552916	0.596832253419726	0.598632109431245	0.339812814974802
Precision@5:	0.467710583153348	0.477969762419006	0.477969762419006	0.285961123110151
LRAP:	0.58437931330346	0.598076936521985	0.599768720589043	0.328696841216877
True positives:	**2831**	**2863**	**2859**	**1917**
False positives:	**6312**	**6397**	**6401**	**7343**
False negatives:	**1119**	**1087**	**1091**	**2033**
Documents evaluated:	926	926	926	926

for this automated indexing framework the test dataset with 926 records is completely unknown. The test dataset (in TSV format) is evaluated through Annif by using four major backend algorithms, namely, TF-IDF, Omikuji-Parabel, Omikuji-Bonsai and NN-Ensemble (combining TF-IDF, Omikuji-Parabel and Omikuji-Bonsai) as separate projects. The test dataset with human-assigned index terms (as the Gold standard) is utilised in OpenRefine to generate suggested descriptors in Annif by using using different backend algorithms and by following the mechanism as discussed in section 6.

A comparative study of the scores, generated on the basis of an array of retrieval metrics by the '*eval*' command for the major backend algorithms, is given in Table 4 to understand the relative performances.

The wiki of Annif says that the two most important values from the array of retrieval results are F1@5 and NDCG. The comparative scores for these retrieval metrics show that the Omikuji and NN-Ensemble backends of the AI/ML-based automated indexing framework have performed better than the TF-IDF backend, considering the evaluation parameters (given in bold text) against human-assigned indexing as the Gold standard.

8. FUTURE RESEARCH

The convergence of data carpentry and AI/ML-based knowledge processing will lead to many interesting breakthroughs in the domain of LIS. It will, in the near future, pave the path toward designing systems for the automatic generation of class numbers and suitable subject descriptors for large volumes of documents. Some of the possibilities include: auto-generation of UDC-based class numbers (as UDC summary is available as a LOD dataset), conversion of DDC as LOD datasets and

then automatic class number synthesis, using MeSH for developing automated indexing systems for bio-medical literature (MeSH is available as a LOD dataset), and so on. The Annif framework may also be utilised as the backend indexing system to generate suggestions for the subject access field (Tag 650) on the basis of title (Tag 245$a) and summary note (Tag 520$a) in an ILS like Koha or for populating the DC.Subject metadata element automatically on the basis of text input in the DC.Title and DC.Description elements in DSpace or EPrints digital archiving systems.

9. CONCLUSIONS

The AL/ML-based indexing system is still in its infancy but is already showing its potential for processing large volumes of bibliographic records. This is the perfect time for LIS professionals and LIS schools in the country to start learning data carpentry applications and AI/ML-based tools. So far, AI and ML applications have been either commercial endeavors or large-scale organisational initiatives. However, open source software solutions and open datasets have broadened horizons, allowing LIS professionals to experiment with these next-generation tools. This research study is a preliminary account of experimentation with an open source AI and ML tool that can offer a variety of sophisticated options that have not yet been fully explored, such as optimisation of the parameters in using Omikuji and NN-Ensemble as backend algorithms, the use of spaCy or Simplemma as a multilingual document analyser, and so on.

REFERENCES

1. Salton, G.; Wong, A. & Yang, C.S.A vector space model for automatic indexing. *CACM*, 1975. doi: 10.1145/361219.361220.
2. Salton, G. & McGill, M.J. Introduction to modern information retrieval. New York: McGraw-Hill, 1983.
3. Wu, H.C.; Luk, R.W.P.; Wong, K.-F. & Kwok, K.-L. Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. on Infor. System*, 2008, **26**, 13:1-13:37. doi: 10.1145/1361684.1361686.
4. ISO. IISO 5963:1985: Documentation - methods for examining documents, determining their subjects, and selecting indexing terms. 1985.
5. Golub, K.; Soergel, D.; Buchanan, G.; Tudhope, D.; Lykke, M. & Hiom, D. A framework for evaluating automatic indexing or classification in the context of retrieval. *J. Assoc. Infor. Sci. Technol.*, 2016, **67**(1), 3–16. doi: 10.1002/asi.23600.
6. Golub, K. Automated subject indexing: An overview. *Cataloging & Classification Q.*, 2021, **59**(8), 702–719. doi: 10.1080/01639374.2021.2012311.
7. Shafer, K.E. Automatic subject assignment via the Scorpion system. *J. Libr. Adm.*, 2001, **34**(1–2), 187–189. doi: 10.1300/J111v34n01_28.
8. OCLC. (2022, June 8). *Scorpion*. OCLC. https://www.oclc.org/research/activities/scorpion.html (Accessed on 02/08/2022)
9. Möller, G.; Carstensen, K.-U.; Diekmann, B. & Wätjen, H. Automatic classification of the World-Wide Web using the Universal Decimal Classification. 1999.
10. Frank, E. & Paynter, G.W. Predicting Library of Congress classifications from Library of Congress subject headings. *J. Am. Soc. Infor. Sci. Technol.,* 2004, **55**(3), 214–227.
11. Joorabchi, A. & Mahdi, A.E. Classification of scientific publications according to library controlled vocabularies: A new concept matching-based approach. *Libr. Hi Tech,* 2013, **31**(4). doi: 10.1108/LHT-03-2013-0030.
12. Handler, A.; Denny, M.; Wallach, H. & O'Connor, B. Bag of what? simple noun phrase extraction for text analysis. Proceedings of the First Workshop on NLP and Computational Social Science, 5 November 2016, Austin, Texas, USA. 2016. pp. 114–124. doi: 10.18653/v1/W16-5615.
13 Hillard, D.; Purpura, S. & Wilkerson, J. Computer-assisted topic classification for mixed-methods social science research. *J. Infor. Technol. Politics*, 2008, **4**(4), 31–46. doi: 10.1080/19331680801975367.
14. Purpura, S. & Hillard, D. Automated classification of congressional legislation. Proceedings of the 2006 International Conference on Digital Government Research, 21-24 May 2006, San Diego, California, USA. 2006. pp. 219–225. doi: 10.1145/1146598.1146660.
15. Roitblat, H.L.; Kershaw, A. & Oot, P. Document categorisation in legal electronic discovery: computer classification vs. manual review. *J. American Soc. Infor. Sci. Technol.*, 2010, **61**(1), 70–80. doi: 10.1002/asi.21233.
16. Young, L. & Soroka, S. Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 2012, **29**(2), 205–231. doi:10.1080/10584609.2012.671234.
17. Anderson, J.D. & Perez-Carballo, J. The nature of indexing: How humans and machines analyze messages and texts for retrieval: Part II: Machine indexing, and the allocation of human versus machine effort. *Infor. Processing Manage.*, 2001, **37**(2), 255–277. doi: 10.1016/S0306-4573(00)00046-7.
18. Saracevic, T. Relevance: A review of the literature and a framework for thinking on the notion in information science: Part II: nature and manifestations of relevance. *J. Am. Soc. Infor. Sci. Technol.*, 2007, **58**(13), 1915–1933. doi: 10.1002/asi.20682.
19. Svarre, T.J. & Lykke, M. Experiences with automated categorisation in e-government information retrieval. 2014, **41**(1), 76-84. doi: 10.5771/0943-7444-2014-1-76.
20. Lancaster, F.W. Indexing and abstracting in theory and practice (3rd ed). University of Illinois, 2003.
21. Borlund, P. The concept of relevance in IR. *J. Am. Soc. Infor. Sci. Technol.*, 2003, **54**(10), 913–925. doi: 10.1002/asi.10286.
22. Huang, X. & Soergel, D. Functional relevance and

inductive development of an e-retailing product information typology. *Infor. Res.*, 2013, **18**(2).
23. Lin, S.-C.; Yang, J.-H.; Nogueira, R.; Tsai, M.-F.; Wang, C.-J. & Lin, J. Multi-Stage Conversational Passage Retrieval: An Approach to fusing term importance estimation and neural query rewriting (arXiv:2005.02230), 2021. arXiv. http://arxiv.org/abs/2005.02230.
24. Silvester, J.P. Computer supported indexing: a history and evaluation of NASA's MAI system. 1997, Supplement 24, 61(No. NASA/CR-97-206517).
25. National Library of Medicine (NLM). NLM Medical Text Indexer (MTI). 2002. https://lhncbc.nlm.nih.gov/ii/tools/MTI.html (Accessed on 31/07/2022)
26. National Agricultural Library. NFAIS webinar: automated indexing: A case study from the National Agricultural Library, 2014. https://www.issn.org/newsletter_issn/nfais-webinar-automated-indexing-a-case-study-from-the-national-agricultural-library (Accessed on 30/07/2022).
27. Junger, U. Automation first-the subject cataloguing policy of the Deutsche Nationalbibliothek. 2017. http://library.ifla.org/id/eprint/2213/ (Accessed on 30/07/2022).
28. Suominen, O. Annif: DIY automated subject indexing using multiple algorithms. *LIBER Q.: J. Assoc. European Res. Libr.*, 2019, **29**(1), 1–25. doi: 10.18352/lq.10285.
29. Suominen, O.; Inkinen, J. & Lehtinen, M. Annif and Finto AI: developing and implementing automated subject indexing. *JLISIT*, 2022, **13**(1), 265–282. doi: 10.4403/jlis.it-12740.
30. Hahn, J. Semi-automated methods for Bibframe work entity description. *Cataloging & Classification Q.*, 2021, **59**(8), 853–867. doi: 10.1080/01639374.2021.2014011.
31. Hahn, J. Cataloger acceptance and use of semi-automated subject recommendations for web scale linked data systems. *IFLA WLIC,* 2022, 10. https://repository.ifla.org/bitstream/123456789/1955/1/062-hahn-en.pdf (Accessed on 28/07/2022).
32. Oliver, C. Leveraging KOS to extend our reach with automated processes. *Cataloging & Classification Q.*, 2021, **59**(8), 868–874. doi: 10.1080/01639374.2021.2023717.
33. Chung, Y.-M.; Pottenger, W.M. & Schatz, B.R. Automatic subject indexing using an associative neural network. *ACM Int. Conf. Digital Libr.*, 1998. doi: 10.1145/276675.276682.
34. Mishra, A. & Vishwakarma, S. Analysis of tf-idf model and its variant for document retrieval. *Int. Conf. Computational Intelligence and Commun. Networks*, 2015. doi: 10.1109/CICN.2015.157.
35. Aizawa, A. An information-theoretic perspective of tf–idf measures. *Infor. Processing Manage.*, 2003, **39**(1), 45–65. doi: 10.1016/S0306-4573(02)00021-3.
36. Thomas, R. & Uminsky, D. The problem with metrics is a fundamental problem for AI. 2020. (arXiv:2002.08512). ArXiv. doi: 10.48550/arXiv.2002.08512.
37. Mukhopadhyay, P.; Mitra, R. & Mukhopadhyay, M. Library carpentry: towards a new professional dimension (part i – concepts and case studies). *SRELS J. Infor. Manage.*, 2021, **58**(2), 67–80. doi: 10.17821/srels/2021/v58i2/159969.
38. Mukhopadhyay, P. How green is my Valley? measuring open access friendliness of Indian Institutes of Technology (IITs) through data carpentry. *In Panorama of open access: Progress, Practices & Prospects*, 2022, 67–89. EssEss. doi: 10.5281/zenodo.6511080.
39. Scikit-Learn. Metrics and scoring: quantifying the quality of predictions, 2007. Scikit-Learn. https://scikit-learn/stable/modules/model_evaluation.html (Accessed on 03/08/2022).

CONTRIBUTORS

Mr Mustak Ahmed, is Research Scholar (JRF) in Department of Library and Information Science, University of Kalyani, Kalyani, Nadia. His research interests include Virtual learning environment (VLE), Learning object repository (LOR), Learning tool interoperability (LTI) etc.
His role in this research study includes constructing the research problems, data curation, and data wrangling.

Ms Mondrita Mukhopadhyay, is Research Scholar (URS-Senior) in Department of Library and Information Science, University of Kalyani, Kalyani, Nadia. Her research interests include Geodetic search, Library discovery system, Integrated library management system (ILMS).
Her role in this research study includes literature review, data collection and testing the framework.

Mr Parthasarathi Mukhopadhay, is Professor in Department of Library and Information Science, University of Kalyani, Kalyani, Nadia. His research interests include Open source and open standards, Data carpentry, Library discovery system and AI/ML based applications.
His role in this research study includes designing the framework, constructing the methodologies, and data reconciliation.