code{4}lib JOURNAL

Mission Editorial Committee Process and Structure Code4Lib

Search

Issue 52, 2021-09-22

Conspectus: A Syllabi Analysis Platform for Leganto Data Sources

In recent years, higher education institutions have implemented electronic solutions for the management of syllabi, resulting in new and exciting opportunities within the area of large-scale syllabi analysis. This article details an information pipeline that can be used to harvest, enrich and use such information.

by David Massey, Thomas Sødring

Introduction

In recent years, higher education institutions have implemented electronic solutions for the management of syllabi [1]. This opens up for the development of new student experiences and workflows, as well as possibilities for integration with learning management systems as well as other systems. From a research perspective, new and exciting opportunities within the area of large-scale syllabi analysis will materialize once a significant portion of reading lists are digital and publicly available. One such opportunity will be the possibility to analyze and compare syllabi from both intra- and inter-institutional perspectives, as well as intra- and inter-domain perspectives.

The Conspectus project has a goal to develop an open source ETL-style information pipeline for syllabi analysis, and has three overarching functionalities: harvesting, enrichment, and dissemination. There are a number of systems for managing syllabi available, including Leganto [2] from ExLibris, Aspire [3] from Talis, KeyLinks [4] from CLA and Kortext, and BLUEcloud Lists [5] from SirsiDynix. Many of these have a published API that may allow for the automated extraction of data. Conspectus is currently limited to using metadata extracted from a server hosting the ExLibris Leganto syllabus management system, but it should be possible to extract metadata from the other systems as well.

The article is organized into four main sections. In the first section, we introduce the high-level architecture of Conspectus. In the second section, we describe the Leganto API and our harvesting approach. In the third section, we describe an enrichment process using a selection of data sources. In the final section, we discuss issues regarding dissemination, transformation of the data to RDF, and the general use of the pipeline.

Conspectus architecture

Conspectus is an information pipeline to harvest, enrich, use and disseminate digital syllabi information. Figure 1 details an architectural overview of Conspectus. The pipeline consists of three stages. The harvesting stage downloads data from any number of Leganto servers. Currently, we only have access to the local OsloMet (Oslo Metropolitan University, https://www.oslomet.no/en) Leganto server, however once data processing agreements can be reached with other institutions we expect the collection of data to increase significantly. In the context of Norwegian higher level institutions, each institution has its own Leganto server. The enrichment stage makes use of additional sources, that include open data sources available on the Internet, as well as information procured from local administrative systems at the University. Finally, the dissemination and use stage makes the processed data available to data consumers.

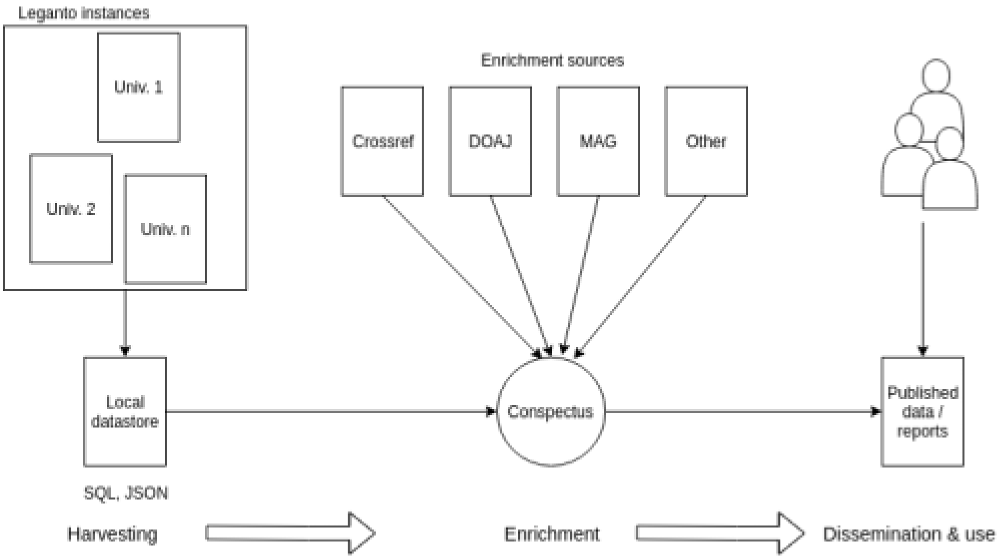

Figure 1. Conspectus Architecture.

The Leganto system provides access to course syllabi, proposed reading lists, instructors, and syllabus comments. Figure 2 depicts the relationship between these resources. The Leganto API exposes endpoints for multiple entities, however, only the course, reading list and citation entities are deemed to be relevant.

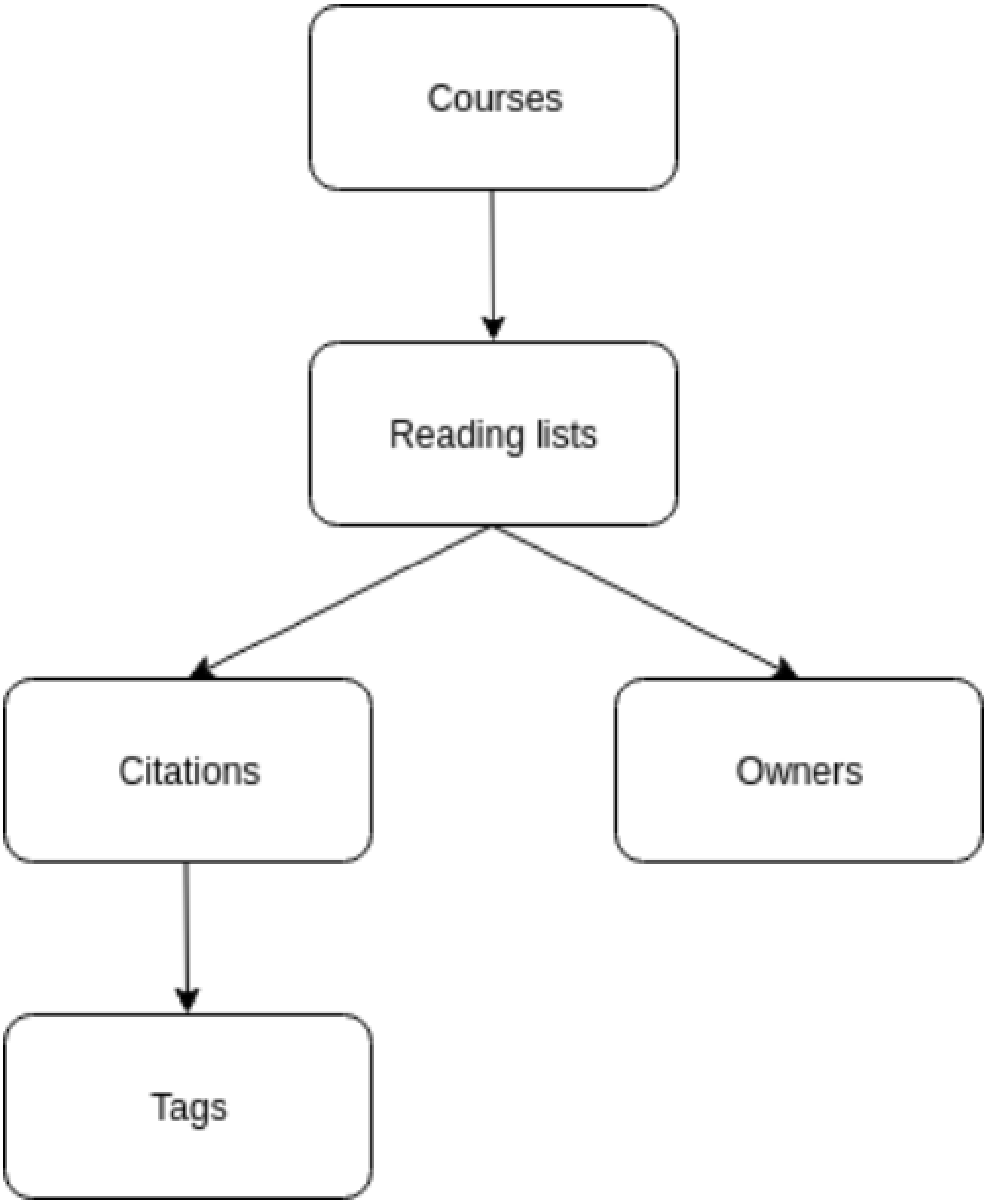

Figure 2. Resources available from the Leganto API.

Harvesting

Conspectus currently only has support for the Leganto API and can download syllabus and related information from a Leganto server that complies with version 1 of the API [6]. A specially purposed harvester retrieves syllabi metadata, that is subsequently stored both as JSON text files as well as in a relational database. Citation data is also aggregated in a single large CSV file. Conspectus employs this approach in order to support various tools and expectations of researchers.

Course data

The highest level of the API provides a list of courses. Listing 1.1 shows an example of a curl command to retrieve information on courses. The command consists of a straight forward HTTP request with a hostname, context-path (almaws/v1) and entity type. The only additional requirement is the identification of an API key to interact with the API. The API is well-documented and relatively easy to use. The API supports result filtering using query parameters and allows users to retrieve results in both JSON and XML formats. Note, all URLs corresponding to internal resources have been replaced with the hostname https://example.com.

```
curl "https://example.com/almaws/v1/courses?apikey=[REDACTED]&format=json"
```

Listing 1.1. Curl command to get all courses from the API in JSON format.

Listing 1.2 shows a subset of the metadata belonging to a course. The information that is stored here includes details about the instructors, the department the course belongs to, the year the course occurs, a Leganto identifier for the course, an identifier about the previous instance of the course, the official name of the course and identifiers used for search. The link field is a URL that points to the current course.

```
{
"searchable_id ": [
     "MBIB4600",
     "UA_215_MBIB4600_1_2021_VÅR_1"
 ],
 "name": "MBIB4600 - Interactive Information Retrieval (2021 -V)",
 "academic_department ": {
     "desc": "Institutt for arkiv -, bibliotek - og informasjonsfag",
     "value": "215 _15_2"
 },
 "year": "2021" ,
 "id": "5798071290002212" ,
 "link": "https :// example.com/almaws/v1/courses/5797973330002212" ,
 "instructor ": [ {
     "last_name ": "Doe",
     "first_name ": "John"
 }],
 "rolled_from ": "4729892920002212"
}
```

Listing 1.2. Abridged version of a payload describing a particular course from the OsloMet Leganto instance.

Some values included here are imported from other systems. These include information about the academic department, searchable identifiers and the official name of the course. There is also a field called *primary_id* (not shown) that is associated with an instructor.

For brevity, some information has been excluded. This includes fields that appear to have relevance for internal system processing, fields indicating processing history, as well as fields that appear to not be in use. An example of a field that appears to not be in use is a field called *campus*. It is always empty.

Reading lists

Leganto supports the management of reading lists on a per-course basis, where multiple reading lists can be associated with a course. A reading list is a list of required, recommended or additional reading resources defined for a course. Syllabus citations are associated with a reading list, so a course must have at least one reading list if it has a syllabus. Listing 1.3 shows an example of a curl command to get the sole reading list for the spring 2021 MBIB4600 course. The command shows how a reading list identifier (5797973440002212) is explicitly associated with a course (5798071290002212), and a request for a reading list must include the course identifier as well as the reading list identifier. An HTTP endpoint is available that identifies all reading lists associated with a course.

```
curl "https://example.com/almaws/v1/courses/5797973330002212/reading-lists/5797973440002212?apikey=[REDACTED]"
```

Listing 1.3. Curl command to get the only reading list for the course MBIB4600 (spring 2021).

Listing 1.4 shows the returned payload for a request for a reading list. From a bibliographic analysis perspective, there is little value in the data here. The contents of the description field is mostly a description of the study-program and the year the class (cohort) started, and does not provide any insight into the contents of the reading list. Very few reading lists provide any information about the purpose of the reading list, while a few reading lists provide an overview of the number of pages that the reading list consists of.

```
  {
    "id": "5797973440002212" ,
    "name": "MBIB4600 - Interactive Information Retrieval (2021 -V)",
    "due_back_date": "2021-07-31Z",
    "description": "Mandatory and optional literature for MBIB4600 (IIR)",
    "link": "https://example.com/almaws/v1/courses/5797973330002212/reading-lists/5797973440002212",
    "status": {
        "desc": "Complete",
        "value": "Complete"
    },
    "visibility": {
        "desc": "Anyone Restricted",
        "value": "OPEN_TO_WORLD"
    },
  }
```

Listing 1.4. Abridged version of a payload describing an identified reading list from the OsloMet Leganto instance. (Contents translated to English, for clarity)

There are 5,759 courses registered in the OsloMet Leganto server (as of June 2021). These courses cover seven semesters of teaching (autumn and spring). New information about a course is generated each time a course is scheduled to be run. The course *MBIB4600* is, for example, registered three times. A course may change name, be retired and replaced by a similar course, retired without replacement, and new courses will be introduced. As such, the details presented here are aggregate data. 121 courses use more than one reading list. 109 courses have more than two reading lists. The fact that most courses have a single reading list, while only a small percentage (approx 2%) use more than one reading list, tells that the functionality is not widely used. It is unclear why so few courses have multiple reading lists. It may be an intentional approach by the course instructors, or it could be that they are unaware that the functionality is available. Another reason could be that the practices employed in previous approaches to managing syllabus citations, e.g., citations in office document formats, led to a similar practice being continued in Leganto.

812 courses (approx. 14%) have no reading list associated with them. A reading list is required for syllabus citations, so these courses have no citations associated with them in the Leganto server.

Citations

Citations are references to a source of information used as part of a reading list. Typically, citations make up part of the information requirements that provide students with the necessary skills and knowledge in order to successfully pass the assessment requirements when studying a course.

From a bibliographic analysis point of view, the citations' endpoint offers a trove of interesting information. Listing 1.5 shows the curl command to get a particular citation used in the syllabus of the spring 2021 MBIB4600 course. The command shows that each citation has a unique identifier (5933190900002212), that is explicitly associated with both the course (5798071290002212) and reading list identifier (5797973440002212). There is an HTTP endpoint available that identifies all citations associated with a reading list.

```
curl "https://example.com/almaws/v1/courses/5798071290002212/reading-lists/5797973440002212/citations/59331909
```

Listing 1.5. Curl command to get a particular citation that is part of the reading for the course MBIB4600 (spring 2021).

The payload response to the request described in Listing 1.5 is presented separately in Listing 1.6 and Listing 1.7. Listing 1.6 describes the administrative context of the citation, while 1.7 details bibliographic information about the citation. Administrative information includes details about whether the citation describes an article or book and that it is either physical or electronic. The *secondary_type* object details the type of object the citation is. This can, for example, be an audio recording, video, or manuscript. A peculiarity about the payload (not shown for brevity) is that there are ten fields named *source1* to *source10*. From a data modelling point of view, it appears that such a source field is a multi-valued field and should be modelled as a separate entity in a 0:m relationship to the citation.

```
{
 "id": "5933190900002212" ,
 "link": "https://example.com/almaws/v1/courses/5797973330002212/reading-lists/5797973440002212/citations/593
 "type": {
     "desc": "Physical Article",
     "value": "CR"
 },
 "status": {
     "desc": "Complete",
     "value": "Complete"
 },
 "secondary_type": {
     "desc": "Article",
     "value": "CR"
 },
 "section_info": {
     "visibility": false ,
     "description": "Supplementary",
     "section_tags": {
         "link": "",
         "total_record_count": 0
     },
     "section_locked": false ,
     "id": "5797973690002212" ,
     "name": "Optional literature"
 },
 "note": [],
 "copyrights_status": {
     "desc": "Not Determined",
     "value": "NOTDETERMINED"
 }
}
```

Listing 1.6. An abridged example of a citation belonging to the reading list presented earlier.

Each citation has a *metadata* object that contains detailed information about the citation. Listing 1.7 shows an example of a metadata object.

```
"metadata": {
 "author": "Taylor , Robert S",
 "issn": "00100870" ,
 "pmid": null ,
 "start_page": "251" ,
 "publication_date": "2015-03-01" ,
 "end_page": "267" ,
 "doi": "10.5860/crl.76.3.251" ,
 "issue": "3" ,
 "title": null ,
 "isbn": null ,
 "journal_title": "College & research libraries",
 "chapter": null,
 "article_title": "Question - Negotiation and Information Seeking in Libraries",
 "note": null ,
 "part": null ,
 "volume": "76" ,
 "publisher": null ,
 "year": "2015-03-01" ,
 "edition": null ,
 "pages": "251-267" ,
 "source": "https://example.com/10.5860/crl.76.3.251"
}
```

Listing 1.7. An example of a metadata object belonging to a particular citation.

Most of the fields in the metadata object are self-explanatory. There are, however, some data quality issues present. The *publication_date* field does not follow a standardized date format. The variations here include “YYYYMMDD”, “YYYY-MM-DD”, “YYYYMM”, “YYYY-MM”, “YYYY”, “[YYYY]”, “cop. YYYY”, “cYYYY” and a text value indicating that the year is unknown. There are other variations that make the field difficult to search. The year field also does not appear to adhere to a particular standardized description, with a number of variations. The year field can be a date field with similar variations of “YYYYMMDD” (found in *publication_date*), as well stating the month in textual form. There are two title fields, *title* and *article_title*. If a citation is identified as an article, then *article_title* is used, otherwise *title* is used.

There are 98,549 citations registered in the OsloMet Leganto server (as of June 2021). The maximum number of citations registered for a reading list is 142. 31 reading lists have more than 100 citations registered. 295 reading lists have only one citation. In some cases, the citation is a single note stating that there is no mandatory syllabus. In other cases, the syllabus consists of a single book. The average number of citations per reading list is 21.

There are 72 notes associated with citations. That equates to approximately 0.07% of citations having a note attached to them. The notes appear to be comments to students, but some are messages to the library staff.

The harvester

The harvester is a Python (version 3.6) script and is distributed as free and open source software. It is available from Gitlab [7]. The harvester requires the use of a MySQL relational database to store information for processing. The domain model implemented in the database reflects the JSON payloads returned from the API, where array results are implemented as 0:m relationships. The code repository also includes the DDL to create the database structure in MySQL.

The Leganto API makes limited use of HTTP-linking and, as such, it can be seen as an implementation of a level 2 hateoas API [8]. Any client interacting with the API must themselves build HTTP links to the various objects based on provided identifiers. A level 3 hateoas API would allow a user to traverse the API endpoints using built-in links in the returned payloads.

The harvester works by first downloading all courses available from the API. Any request to retrieve courses will have a count field detailing the number of courses available via the API. This allows for an upper bound in relation to the number of courses to retrieve. Listing 1.8 shows an example of a command using the harvester to download all course data from the Leganto API. The option of setting the institution allows the script to download from multiple Leganto servers belonging to various institutions.

```
python main.py --institution=OsloMet --what=course --start=1 --end=5760
```

Listing 1.8. Command to run harvester to retrieve courses.

The ordering of results is deterministic based on the values in *code* and *section*. As such, they are not related to time and new courses will not appear at the end of the list of courses when querying the API. It is possible, however, to add a query parameter called *order_by* to assign a particular ordering to the result set. This has not been implemented in the current version of the API. The harvester retrieves courses, ten at a time, until all courses are downloaded. This is a courtesy approach, to reduce server side processing.

The harvester downloads course information and persists it to a local MySQL database. It also saves the retrieved JSON data as a JSON files using the course identifier as the file name. Leganto can also provide the data as XML, but the harvester only requests JSON. Once the required course information has been downloaded, the harvester can be re-run to download reading lists. Listing 1.9 shows the command to collect reading list data.

```
python main.py --institution=OsloMet --what=reading-list --start=1 --end=5760
```

Listing 1.9. Command to run harvester to retrieve reading lists.

Reading lists are downloaded in the same order courses are persisted in the courses table in the database. The retrieved reading lists are also stored as the harvester can be re-run to download related citations. Listing 1.10 shows the command to download citations. Recall that the citations are only retrievable using a combination of the course and reading list identifiers.

```
python main.py --institution=OsloMet --what=citation --start=1 --end=5760
```

Listing 1.10. Command to run harvester to retrieve citations.

The harvester avoids a download-everything approach, as potential problems (for example, server unavailability) will likely require the script to download everything repeatedly. An incremental approach is more robust and makes it easier to download subsequent data when the Leganto server is updated with new course information at the start of a semester. With more experience from working on multiple Leganto APIs, it may be worth considering alternative approaches.

In Norway, higher education institutions have a unique code available in the Leganto data. This can be seen in Listing 1.2 where the value in the *academic_department* field is *215_15_2*. Such unique identifiers may not be in use in other countries, so the script allows additional tagging of institutional information.

Enrichment

The syllabus metadata are enriched using both internal (university administrative system) and external data sources using various protocols, APIs and file downloads. The information pipeline addresses some general interoperability problems faced when using heterogeneous data sources. The external data sources currently used are Crossref, Microsoft Academic Graph (MAG), Directory of Open Access Journals (DOAJ), the Norwegian Register for Scientific Journals, Series and Publishers (NSD), and the Bibsys union catalog. The purpose of the enrichment data is twofold. Firstly, to improve existing data by replacing semi-structured data (e.g., author strings) with structured, machinereadable data. Secondly, to add additional properties, such as language and subject descriptors, and thus enabling enhanced analysis.

The results of the enrichment stage, presented here, is based on a subset of the downloaded data and limited to 25,333 citations from the 2020-21 academic year.

Identifiers

A successful enrichment process depends on the reliable interlinking of metadata descriptions across data sources, and requires the use of shared identifiers. Table 1 lists the four relevant identifiers available in the Leganto citation metadata that can be used in an enrichment process.

The Metadata Management System (MMS) identifier differs from the other three identifiers, in that it is an internal identifier, linking the syllabus citation metadata to the corresponding metadata description in the university library management system. Note that the *mms_id* field is only found in the metadata object of the citation when a link to the library system is made. The syllabus metadata schema also allows for a Pubmed (https://pubmed.ncbi.nlm.nih.gov/) identifier, but no values are found in the retrieved metadata.

Table 1. Identifiers found in the Leganto metadata.

Identifier	Acronym	Identifies
International Standard Book Number	ISBN	Books
International Standard Serial Number	ISSN	Journals
Metadata Management System	MMS id	Books and Journals
Digital Object Identifier	DOI	Articles and Books

We limit the current pipeline to citations that describe either books or articles. Together, they account for almost 90% of the citations. 57.6% of the citations are books and 30% are articles. The remaining 13% include web pages and other document types.

The ISSN field identifies journals rather than individual articles. The pipeline has an assumption that journal metadata may be able to provide additional relevant information about an article. For example, the language accepted by the journal can be used to assign a language code to an article that is published in the journal.

Approximately 90% of the article citations contain an ISSN, whereas only approximately 60% contain a DOI. The DOIs and ISSNs complement each other, enabling the use of a greater number of enrichment sources. Seven percent of the articles contain neither identifier, and therefore cannot be linked to enrichment sources. A similar identifier coverage is found for book citations, where approximately 90% of the citations contain one or more ISBN.

The next question the pipeline explored is which sources should be used to enrich the Leganto metadata? This leads to a series of additional questions: Which sources can be queried using the identifiers (ISSNs, DOIs and ISBNs)? Which sources offer high levels of coverage for the target articles and books? Which sources contain relevant metadata? Our current selection of sources are listed in Table 2. This selection is only an example of possible sources, and shows which identifier is used to query the source and the access method used. The table also shows the level of coverage, that is, what proportion of the syllabus identifiers that are found in the source. The low level of coverage for DOAJ suggests that only a limited number of the articles are published in open access journals. The final *overall coverage* column shows how many of the books or articles that can be linked using the source. The ceiling for this value is the identifier coverage discussed earlier. The Crossref and MAG coverage for DOIs is high, but the overall coverage is lower because fewer article citations contain DOIs than ISSNs.

Table 2. Current external enrichment sources.

Source	Publication Type	Identifier	Method	Coverage	Overall Coverage
Bibsys	Book	ISBN	SRU	94.7	91.2
Bibsys	Journal	ISSN	SRU	96.8	81.9
DOAJ	Journal	ISSN	Download	11.3	14.8
Crossref	Journal	ISSN	API	85.3	71.2
NSD	Journal	ISSN	Download	92.9	80.4
Crossref	Article	DOI	API	94.4	63.8
MAG	Article	DOI	API	94.5	63.8

We continue with a closer examination of the enrichment sources. How they are accessed, and what data do they provide?

Bibsys Union Catalogue

OsloMet is a part of the Bibsys union catalog. The catalog contains bibliographic metadata describing the holdings of 80 Norwegian higher education and special libraries. The source can be queried using the SRU (Search/Retrieval via URL) protocol (http://www.loc.gov/standards/sru/). ISBN, ISSN and MMS identifiers can be queried using separate search indexes [9]. For example, an SRU query for the example article's ISSN is shown in Listing 1.11.

```
curl "https://bibsys.alma.exlibrisgroup.com/view/sru/47BIBSYS_NETWORK?version=1.2&operation=searchRetrieve&rec
```

Listing 1.11. URL for ISSN lookup in the Bibsys SRU service.

Results can be returned in a variety of standard formats including MARC21 XML [10], Dublin Core [11] and Metadata Object Description Schema (MODS) [12]. We use the MARC21 format as this provides the most detailed data. An abridged version of the returned data is shown in Listing 1.12.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
 <recordData>
 <record xmlns="http://www.loc.gov/MARC21/slim">
 <leader>01443 cas a2200385 c 4500</leader>
 <controlfield tag="001">999101706734702201</controlfield>
 <controlfield tag="005">20180613120321.0</controlfield>
 <controlfield tag="007">ta</controlfield>
 <controlfield tag="008">150417d19392013xx#b||p|||||||||||0eng|d</controlfield>
 <datafield ind1=" " ind2=" " tag="022" >
   <subfield code="a">0010-0870</subfield>
 </datafield>
 <datafield ind1=" " ind2=" " tag="080" >
   <subfield code="a">027.7(05)</subfield>
 </datafield>
 <datafield ind1="0" ind2=" " tag="130" >
   <subfield code="a">College &amp; research libraries (trykt utg.)</subfield>
 </datafield>
 <datafield ind1="1" ind2="0" tag="210" >
   <subfield code="a">Coll.res.libr.</subfield>
 </datafield>
 <datafield ind1="1" ind2="0" tag="245" >
   <subfield code="a">College &amp; research libraries :</subfield>
   <subfield code="b">C& amp;RL</subfield>
 </datafield>
 <datafield ind1=" " ind2="7" tag="650" >
   <subfield code="a">Bibliotekvesen</subfield>
   <subfield code="v">Tidsskrifter</subfield>
   <subfield code="2">tekord</subfield>
 </datafield>
</record>
</recordData>
```

Listing 1.12. An abridged MARC21 XML record returned by a BIBSYS SRU query.

Directory of Open Access Journals

The Directory of Open Access Journals (DOAJ, https://doaj.org/) is a database with metadata descriptions of over 15,000 open access journals. The database is available for download (https://doaj.org/public-data-dump/journal) in both CSV and JSON formats. DOAJ is a rich source of information about open access journals. It also provides subject descriptors, as is shown in Listing 1.13 of metadata on our example article.

```
{
 "bibjson":{
 "keywords":[
     "library science",
     "information literacy",
     "higher education"
 ],
 "subject":[
 {
     "code":"Z",
     "scheme":"LCC",
     "term":" Bibliography . Library science . Information resources"
 }
 ],
 "eissn":"2150-6701" ,
 "language":[
     "EN"
 ],
 "title":"College and Research Libraries",
 "pissn":"0010-0870",
 "license":[
 {
     "NC":true,
     "ND":false,
     "BY":true,
     "type":"CC BY-NC",
     "SA":false
 }
 ],
 }
}
```

Listing 1.13. Abridged version of data from DOAJ.

Crossref

Crossref (https://www.crossref.org/) is a DOI registration agency that was launched in early 2000 as a cooperative effort among publishers to enable persistent cross-publisher citation linking of online academic journals. The Crossref REST API allows DOI lookup [13]. A lookup of the example article's DOI is shown in Listing 1.14.

```
curl "https://api.crossref.org/works/10.5860/crl.76.3.251"
```

Listing 1.14. DOI lookup in Crossref API.

The returned metadata is primarily a source of structured reference and citation data. However, as Listing 1.15 shows, the returned JSON data also offers structured information on article authors.

```
{ "message" : {
 "type": "journal-article",
 "created": {
 "date - time": "2015-03-13T14:06:42Z"
 },
 "page": "251-267",
 "title": [
     "Question - Negotiation and Information Seeking in Libraries"
 ],
 "volume":"76",
 "issue":"3",
 "author": [
 {
     "given":"Robert S.",
     "family":"Taylor",
     "sequence":"first",
     "affiliation ":[]
 }
 ]
 "container-title": [
     "College & Research Libraries"
 ]
 }
}
```

Listing 1.15. Abridged version of data from Crossref.

The Norwegian Register for Scientific Journals, Series and Publishers

The Norwegian Register for Scientific Journals, Series and Publishers (NSD, https://dbh.nsd.uib.no/publiseringskanaler/Forside.action?request_locale=en) is available for download as a CSV file. The register contains metadata on approximately 35,000 journals. The CSV file contains 36 columns. To improve readability, Listing 1.16 uses a column name colon value format. For clarity, the column headings have been translated from Norwegian. Relevant columns are *Language, NPI Subject area* (https://npi.nsd.no/fagfeltoversikt), *NPI Subject field and Level 2021*. The level is a classification of journals which is used in publishing metrics. Level 2 journals are deemed the most prestigious.

```
NSD journal_id : 439235
Original title : College & Research Libraries
International title : College & Research Libraries
Print ISSN : 0010 -0870
Online ISSN : 2150 -6701
Open Access : DOAJ
NPI Subject area : Social sciences
NPI Subject field : Library and information science
Level 2021: 2
itar_id : 7656
NSD publisher_id :
Publishing house :
Publisher : American Library Association
```

```
Publication country : USA
Language :
Conference report : 0
Established : 1939
```

Listing 1.16. Abridged version of data from the NSD register for scientific journals (Contents translated to English, for clarity).

Microsoft Academic Graph

The Microsoft Academic Graph (MAG) “is a heterogeneous graph containing scientific publication records, citation relationships between those publications, as well as authors, institutions, journals, conferences, and fields of study” [14]. Microsoft announced in May 2021 that the project will be discontinued from the end of 2021 [15]. OurResearch have recently announced [16] that they will reuse the MAG data set in the OpenAlex service. Conspectus uses the Microsoft Academic Knowledge REST API [17] to query the graph for DOIs. The evaluate method [18] is used to query the graph. The entity attributes to return are listed in a commaseparated string of codes [19]. For example, ’F’ contains data on the field of study. Listing 1.17 shows a query for the DOI of our example article.

```
curl "https://api.labs.cognitive.microsoft.com/academic/v1.0/evaluate?subscription-key=[REDACTED]&attributes=I
```

Listing 1.17. DOI lookup in teh MAG API.

The list of field of study names, shown in Listing 1.18, is a possible source of subject descriptors for the article.

```
{"expr":"DOI==’10.5860/CRL.76.3.251’",
 "entities":[
     "Ti":"question negotiation and information seeking in libraries",
     "Y":1967,
     "CC":676,
     "AA":[{
         "AuN":"robert s taylor",
         "AuId":2140052427
     }],
     "F":[
         {"DFN":"Reference interview"},
         {"DFN":"Information seeking"},
         {"DFN":"Negotiation"},
         {"DFN":"Computer science"},
         {"DFN":"Search theory"},
         {"DFN":"Social communication"}],
     "J":{
         "JN":"college & research libraries"
     }
 }
]}
```

Listing 1.18. Abridged version of data from Microsoft Academic Graph.

Case: Analysing the languages of syllabi citations

The Leganto citation metadata does not include information about the language of the books or articles. An analysis of languages used in the university’s syllabi is therefore dependent on enrichment from external data sources. Figure 3 shows language codes gathered from the five external enrichment sources for our example article. Successfully assigning a language code to a citation depends on four factors. Firstly, that the citation includes one or more identifiers. Secondly, that sources can be queried using the identifier. Thirdly, that the sources contain metadata on the citation. Finally, that the sources contain information on the language of the citation. This illustrates how different forms of completeness – schema, property, and population – can negatively affect interoperability across sources. See [20, p. 103] for definitions of the three forms of completeness. Population completeness is also related to the concept of coverage, that was discussed earlier.

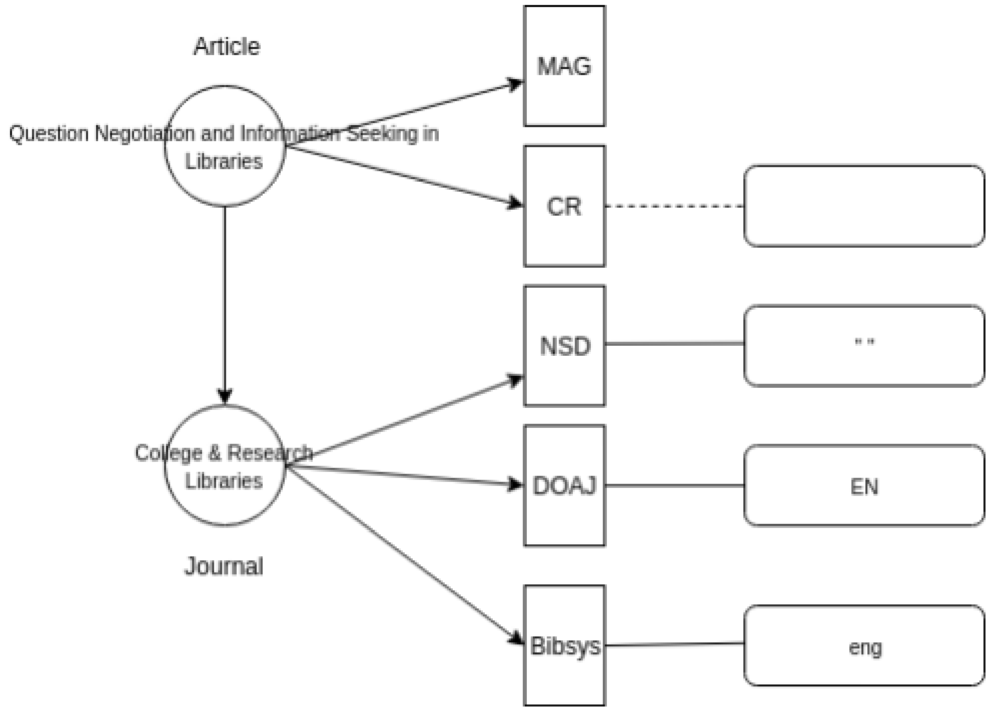
Article
Question Negotiation and Information Seeking in Libraries
MAG
CR
NSD
" "
College & Research Libraries
DOAJ
EN
Journal
Bibsys
eng

Figure 3. Article, journal and language field in external enrichment sources.

The example article is identified with a DOI, and both MAG and Crossref contain metadata on the article. The MAG schema, however, does not include a language field. Thus, enrichment is hampered by inadequate schema completeness. The Crossref schema does include language information, but 18.8% of the matched Crossref records, including our example article, have no language field. This is an example of an issue concerning property completeness. At the journal level, the journal is also identified with an ISSN. The NSD CSV file includes a language column, but 46.1% of the journals have no language assigned. All the DOAJ records contain one or more language codes. Finally, almost two percent of the Bibsys records have *und* (undefined) as a language code. It is clear that the property completeness varies across the sources. The example article and journal from Listing 1.7 are present in all five of the sample data sources, but as we saw earlier (see Table 2), this is not the case for all articles and journals. Population completeness is therefore a data quality issue that can impede effective enrichment.

Interoperability issues also arise at the data value level. Our sample sources express language in a variety of ways. BIBSYS uses MARC21 to describe journals and books. MARC21 stores the document's language in position 35-37 of the 008 control field (https://www.loc.gov/marc/bibliographic/bd008.html). The Library of Congress maintains a registry of the three letter language codes (https://www.loc.gov/marc/languages/language_code.html). Documents that are written in multiple languages are given *mul*. The value for our example article, as shown in line 6 of Listing 1.12, is *eng*. Languages in the NSD data file are specified with Norwegian names rather than codes, for example, *Engelsk* (eng. English). The documentation of the Crossref metadata API JSON format [21] does not include a language field. The returned data for DOI lookup, however, includes a language field for many of the documents. The value is a two-letter code which appears to correspond with the ISO 639-1 standard, e.g., *en*. The DOAJ JSON file contains a language field with a list of language codes. The value is a two-letter code which appears to correspond with the ISO 639-1 standard. Several language codes can be assigned to a journal, for example:

```
"language": ["NB", "DA", "EN", "NN"],
```

We map the language codes and names to a common coding system. Finally, we select one of the codes to represent the language of the citation in our analysis. Factors affecting the choice of code include: codes for articles (DOIs) override codes for journals (ISSNs); agreement between sources; and single codes overriding multiple codes.

Figure 4 uses the language codes to display the proportion of articles, chapters and books written in Norwegian at different academic levels. Not surprisingly, we see a fall in the proportion of Norwegian language citations at the master and PhD levels. A similar pattern is shared across the university's four faculties [22]. The academic level of the course is not contained within the Leganto data. Internal enrichment data from the university administrative systems is therefore combined with the language codes to generate this analysis.

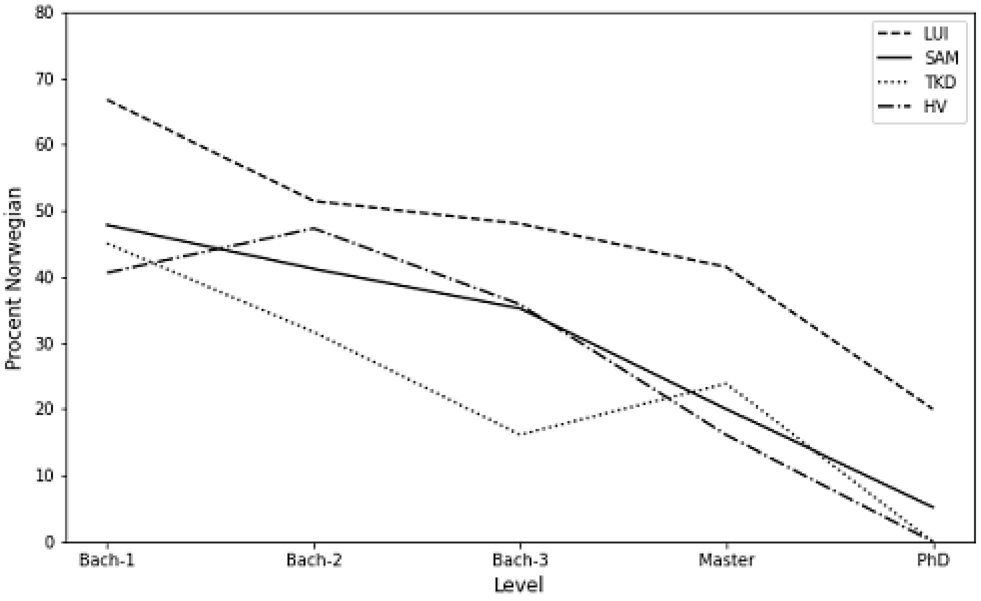

Figure 4. Proportion of Norwegian language citations for five academic levels across the university's faculties.

Dissemination and use

A project that is very relevant to our pipeline is *Open Syllabus* (https://opensyllabus.org/). The Open Syllabus project has collected syllabi from numerous institutions across the world and provides interesting insights. In many ways, the Open Syllabus project is a great example showing how such data can be disseminated and used. Conspectus differs here in that our approach is more fine-grained and tries to deal with data quality and other enrichment issues to provide a platform for syllabus analysis. We expect, however, that the Conspectus pipeline will retrieve relevant citation information from Leganto servers and submit them to the Open Syllabus project.

The Conspectus approach takes existing syllabi metadata that are currently semi-available and aims to make them more openly available in machine-readable sources for others. This builds upon the development of a generic data model to allow for the publication of data using standardized approaches. Currently, the project has explored RDF (https://www.w3.org/RDF/) to make data available to other users. An example use-case is to facilitate libraries undertaking bibliometric research of syllabi.

Semantic representation of the data

We have considered two ontologies found in the literature. Firstly, the Curriculum Course Syllabus Ontology (CCSO, https://vkreations.github.io/CCSO/#d4e3502) discussed in [23]. Secondly, OntoSyllabus (https://jachicaiza.github.io/ontologyDoc/) described in [24]. [25] and [26] also discuss CURONTO: An Ontological Model for Curriculum Representation, but we have been unable to locate and further assess this ontology.

We decided to combine CCSO with Schema.org (https://schema.org/). CCSO is used to model the non-bibliographic entities, such as courses, departments and syllabi. Schema.org is used to model the bibliographic data, such as articles, chapters and books. Table 3 shows the mapping between the merged Leganto and enrichment data and the classes and properties from the chosen ontologies. The initial codes, for example *cit.m* in *cit.m.author*, in the Leganto Field column refer to the resource (citation) and the object (metadata) where the field is located. The codes used in the source column are L for Leganto, I for enrichment data from the internal administrative system, and E for external enrichment data. The prefixes used in the ontology columns are c for CCSO,s for Schema.org, and x for local example properties.

Figure 5 shows the classes we used in transforming the data to RDF. The Schema.org model uses *PublicationVolume* and *PublicationIssue* to link articles to journals. Currently, we simplify this by linking the article directly to the journal. This results in the omission of the Leganto *volume* and *issue* fields. We also omit the CCSO *ProgramofStudy* class, and use a locally minted predicate (ex:academicLevel) to express the academic level the course is taught at (bachelor, master or PhD level).

The relationship between the article and the journal, or between the chapter and the book, is not explicitly expressed in the Leganto data. This is identified by the value of the *secondary_type* sub-fields (see lines 12-15 of Listing 1.6) and the presence of additional Leganto fields in the *metadata* object, such as *journal_title* and *article_title*. In addition to facilitating secure interlinking between data sources, identifiers are crucial for transforming textual labels to RDF entities. For example, the lack of identifiers for authors means we use *schema:author* for a textual label, i.e., a data property, rather than its intended use, to link to an independent entity, i.e., an object property. We hope that enrichment sources can provide the author identifiers that will allow the automated creation of author entities. Alternatively, other reconciliation techniques could be employed to create the entities. Richard Wallis identified entity reconciliation as the key missing piece in the process of transforming legacy cultural heritage data to linked data in a recent blog post [27].

The code of the course is not available as a separate field in the Leganto data, but the course name follows a common pattern [28] that allows the code to be extracted. Such pattern-based extractions are marked with *Regex* in the Notes column of the mapping table. The course code plays an important role as it is a shared identifier between Leganto and the university administrative system.

The Leganto API returns the reading lists that are associated with a course. The links between course, reading list and citations can be added when retrieving data. Alternatively, the citation metadata (see line six of Listing 1.6) contains a *link* field that contains the Leganto identifiers for the linked course and reading list.

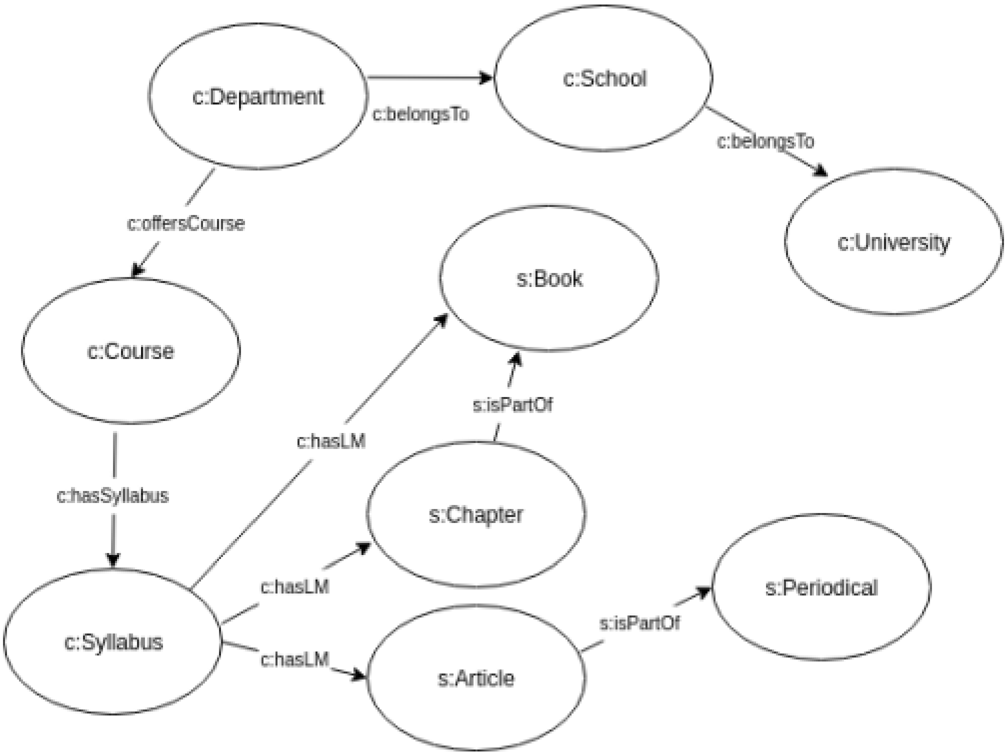

Figure 5. RDF classes and object properties.

The *about* property in Schema.org is used to assign a subject to a citation. It is an object property that links a Thing or the appropriate sub-class of Thing to a citation. Again, this depends on the creation of an entity to represent the concept. We show a simple example of subjects from MAG listed as strings. A successful mapping and merging of subjects from a variety of controlled and uncontrolled vocabularies across languages is a major challenge that goes beyond the simple mapping that is utilized to merge language codes.

The RDF turtle data for the example article, and related reading list, course and other entities are shown in Listing 1.19.

```
@prefix schema : <https://schema.org/> .
@prefix ccso : <https://w3id.org/ccso/ccso/> .
@prefix ex: <https://example.org/> .

<http://example.org/institutions/4fc1875ac49b43fe5ab83121e27b8df5> a ccso:University ;
 schema:name "Oslo Metropolitan University" .

 <http://example.org/schools/a1c4253c96c559930e24f9f5349fd5a9> a ccso:School ;
 schema:name "Faculty of Social Sciences" ;
 ccso:belongsTo <http://example.org/institutions/4fc1875ac49b43fe5ab83121e27b8df5> .

 <http://example.org/departments/764639ea184bb49ed19de5b968a9ea12> a ccso:Department ;
 schema:name "Institutt for arkiv-, bibliotek- og informasjonsfag" ;
 schema:identifier "215 _15_2" ;
 ccso:belongsTo <http://example.org/schools/a1c4253c96c559930e24f9f5349fd5a9> ;
 ccso:offersCourse <http://example.org/courses/b4a9e82caa7f257b67f8834534fd2558> .

<http://example.org/courses/b4a9e82caa7f257b67f8834534fd2558> a ccso:Course ;
```

```
ccso:csName "Interactive Information Retrieval" ;
ccso:code " MBIB4600" ;
ex:academicLevel "Master" ;
ccso:hasSyllabus <http://example.org/syllabi/20b9736c7ff1df5e2001149abb201dc0> .

<http://example.org/syllabi/20b9736c7ff1df5e2001149abb201dc0> a ccso:Syllabus ;
ccso:academicYear "2021-SPRING" ;
ccso:hasLM <http://example.org/articles/d73b25120b396a3cfefa1aa8a88003f0> .

<http://example.org/articles/d73b25120b396a3cfefa1aa8a88003f0> a schema:Article;
schema:name "Question - Negotiation and Information Seeking in Libraries" ;
schema:author "Taylor, Robert S." ;
schema:datePublished "2015" ;
schema:identifier [ a schema:PropertyValue ;
    schema:propertyID "DOI" ;
    schema:value "10.5860/crl.76.3.251" ] ;
schema:pageStart "251" ;
schema:endPage "267" ;
schema:pagination "251 -267" ;
schema:about "Reference interview", "Information seeking", "Negotiation", "Computer science", "Search theory
schema:isPartOf <http://example.org/journals/44a073d72c91b4aa5030553ae18d66fb> .

<http://example.org/journals/44a073d72c91b4aa5030553ae18d66fb> a schema:Periodical;
schema:name "College & research libraries" ;
schema:issn "00100870" ;
schema:inLanguage "eng" ;
ex:level "2" .
```

Listing 1.19. RDF Turtle data for the example article.

Listing 1.20 shows a SPARQL query that returns the names of courses that have a syllabus that contains an article from the journal *College & research libraries*. The titles of the reading list articles are also returned. A similar query could be the basis of a simple network analysis where courses are nodes and shared journals are weighted edges.

```
PREFIX schema : <https://schema.org/>
PREFIX ccso : <https://w3id.org/ccso/ccso/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?course_name ?article_title
WHERE {
 ?course rdf:type ccso:Course ;
         ccso:hasSyllabus ?syllabus ;
         ccso:csName ?course_name .
 ?syllabus ccso:hasLM ?citation .
 ?citation schema:isPartOf ?journal ;
           schema:name ?article_title .
 ?journal rdf:type schema:Periodical ;
          schema:name "College & research libraries" .
}
```

Listing 1.20. Example SPARQL query to find course and article information of all articles published by 'College & research libraries'.

Summary and next steps

Conspectus is an information pipeline that harvests data on courses, reading lists and citations from the OsloMet Leganto syllabus management system using a published API. Harvested citation metadata is enriched using data retrieved from both internal administrative systems from the university, and a sample of external data sources. The chosen external data sources provide an insight into some interoperability challenges faced when combining data from heterogeneous sources. Issues relating to completeness and to mapping are highlighted. We also describe how the enriched data can be transformed to RDF and identify issues that must be faced when undertaking such transformations. An example of how enriched data can expand the possible scope of analysis is also detailed.

Conspectus is an exploratory tool for syllabus research that will mature as the project progresses. The current approach has focused on understanding the process of creating a pipeline and identifying potential enrichment sources. The next steps will see an increased focus on dissemination and use, and may provide relevant feedback to data owners about the data they manage. Data quality issues and potential API improvements are obvious candidates to provide feedback on. Conspectus is built upon an open source approach, however, currently only the code for the harvester has been released (a copy is attached to this article). As the approach and coding matures, the other stages of the pipeline will also be published.

One of the future goals for Conspectus is to provide an alternative to the current data-silo approach where syllabus information is institutionalized in proprietary formats in order to make data available for independent analysis.

The next step for harvesting is to increase the volume of data by importing data from additional Norwegian universities that also use Leganto. After that, we aim to investigate how to integrate data from various syllabus management systems. That is, to go beyond Leganto. This will require the development of a common data model that encompasses data returned from the different APIs. The RDF data model presented under the section "Semantic representation of the data" will form the basis for such a model.

Regarding enrichment, further exploration is necessary to identify other relevant sources. Currently, enrichment is limited to bibliographic entities – journals, articles and books – that have global identifiers. A next step is to expand the enrichment stage to include additional entities (for example, authors and subjects) and use fuzzy string-based matching techniques to link to relevant data sources, such as Wikidata.

The next steps for dissemination require discussions with stakeholders within the university library, and academic departments on potential analysis projects. The Conspectus pipeline can form the basis of a more innovative approach to the way syllabi are viewed and managed by a university library and may provide new ways to undertake bibliometric analysis across courses, departments and institutions.

Table 3. Mapping between Leganto fields, enrichment sources and CCSO and Schema.org ontologies.

Leganto Field	Src.	Ontology Class	Ontology Property	Note
–	I	c:University	s:name	
–	I	c:School	s:name	

Leganto Field	Src.	Ontology Class	Ontology Property	Note
–	I	c:School	c:belongsTo	
cou.ad.desc	L	c:Department	s:name	
cou.ad.value	L	c:Deparment	s:identifier	
–	I	c:Deparment	c:belongsTo	
cou.academic_department	L	c:Deparment	c:offersCourse	
cou.name	L	c:Course	c:csName	Regex
cou.name	L	c:Course	c:code	Regex
–	I	c:Course	e:academicLevel	
rea.link	L	c:Course	c:hasSyllabus	Regex
rea.name	L	c:Syllabus	c:academicYear	Regex
cit.link	L	c:Syllabus	c:hasLM	Regex
cit.m.article_title	L	s:Article	s:name	
cit.m.author	L	s:Article	s:author	String, not Person or Org.
cit.m.year	L	s:Book	s:datePublished	Regex
cit.m.doi	L	s:Article	s:identifier	Blank node of type s:PropertyValue
cit.m.start_page	L	s:Article	s:pageStart	
cit.m.end_page	L	s:Article	s:pageEnd	
cit.m.pages	L	s:Article	s:pagination	
–	E	s:Article	s:inLanguage	
–	E	s:Article	s:about	String, not Thing
–	L	s:Article	s:isPartOf	No PublicationVolume or PublicationIssue
cit.m.journal_title	L	s:Periodical	s:name	
cit.m.issn	L	s:Periodical	s:issn	
–	E	s:Periodical	s:inLanguage	
–	E	s:Periodical	x:level	NSD Publication Level
cit.m.chapter_title	L	s:Chapter	s:name	
cit.m.chapter_author	L	s:Chapter	s:author	String, not Person or Org.
cit.m.chapter	L	s:Chapter	s:position	Check
cit.m.start_page	L	s:Chapter	s:pageStart	
cit.m.start_end	L	s:Chapter	s:pageEnd	
–	L	s:Chapter	s:isPartOf	
cit.m.title	L	s:Book	s:name	
cit.m.author	L	s:Book	s:author	String, not Person or Org.
cit.m.year	L	s:Book	s:datePublished	Regex
cit.m.isbn	L	s:Book	s:isbn	
cit.m.edition	L	s:Book	s:bookEdition	
cit.m.pages	L	s:Book	s:numberOfPages	String, not integer
cit.m.publisher	L	s:Book	s:publisher	String, not Person or Org.
–	E	s:Book	s:inLanguage	
–	E	s:Book	s:about	String, not Thing
cit.m.place_of_publication	L	s:PubilcationEvent	s:location	String, not Location

Notes

[1] Ken Chad. The rise of library centric reading list systems. HELibTech Briefing Paper No, 5:1–13, 2018.

[2] Leganto Course Resource List Management. [Internet]. Available from: https://exlibrisgroup.com/products/leganto-reading-list-management-system/

[3] Talis Aspire. [Internet]. Available from: https://talis.com/talis-aspire/

[4] Introducing KeyLinks: the New Reading List Platform. [Internet]. Available from: https://www.cla.co.uk/news-introducing-KeyLinks

[5] BLUEcloud course Lists. [Internet]. Available from: https://www.sirsidynix.com/bluecloud-course-lists/

[6] Courses – ExLibris Developer Network. [Internet]. Available from: https://developers.exlibrisgroup.com/alma/apis/courses/

[7] Syllabus Harvesting. [Internet]. Available from: https://gitlab.com/alvis-project/bibliometrics/development/syllabus-harvesting

[8] Roy Thomas Fielding. Architectural styles and the design of network-based software architectures. University of California, Irvine, 2000.

[9] BIBSYS SRU details. [Internet]. Available from: https://bibsys.alma.exlibrisgroup.com/view/sru/47BIBSYS_NETWORK?version=1.2&operation=explain

[10] MARC21 XML Schema. [Internet]. Available from: https://www.loc.gov/standards/marcxml/

[11] DC Schema for SRU. [Internet]. Available from: https://www.loc.gov/standards/sru/recordSchemas/dc-schema.html

[12] MODS: Metadata Object Description Schema. [Internet]. Available from: https://www.loc.gov/standards/mods/v3/mods-3-5.xsd

[13] Crossref REST API. [Internet]. Available from: https://github.com/CrossRef/rest-api-doc

[14] Microsoft Academic Graph. [Internet]. Available from: https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

[15] Microsoft Academic. May 4, 2021. Next Steps for Microsoft Academic – Expanding into New Horizons. [Internet]. Available from: https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-to-expand-horizons-with-community-driven-approach/

[16] OurResearch blog. June 13, 2021. MAG replacement update: meet OpenAlex!. [Internet]. Available from: https://blog.ourresearch.org/openalex-update-june/

[17] Microsoft. Project Academic Knowledge. [Internet]. Available from: https://www.microsoft.com/en-us/research/project/academic-knowledge/

[18] Microsoft. Evaluate method. [Internet]. Available from: https://docs.microsoft.com/en-us/academic-services/project-academic-knowledge/reference-evaluate-method

[19] Microsoft. Paper entity attributes. [Internet]. Available from: https://docs.microsoft.com/en-us/academic-services/project-academic-knowledge/reference-paper-entity-attributes

[20] Carlo Batini and Monica Scannapieco. Data and Information Quality: Dimensions, Principles and Techniques. Springer, 2016.

[21] Crossref Metadata API JSON Format. [Internet]. Available from: https://github.com/Crossref/rest-api-doc/blob/master/api_format.md

[22] LUI: Faculty of Education and International Studies; SAM: Faculty of Social Sciences; TKD: Faculty of Technology, Art and Design; HV: Faculty of Health Sciences.

[23] Evangelos Katis, Haridimos Kondylakis, Giannis Agathangelos, and Kostas Vassilakis. Developing an ontology for curriculum and syllabus. In European Semantic Web Conference, pages 55–59. Springer, 2018.

[24] Mariela Tapia-Leon, Carlos Aveiga, Janneth Chicaiza, and Mari Carmen Suárez-Figueroa. Ontological model for the semantic description of syllabuses. In Proceedings of the 9th International Conference on Information Communication and Management, pages 175–180, 2019.

[25] Ghadeer Ashour, Ahmad Al-Dubai, Imaed Romdhani, and Naif Aljohani. An ontological model for courses and academic profiles representation: A case study of king abdulaziz university. In 2020 International Conference Engineering Technologies and Computer Science (EnT), pages 123–126. IEEE, 2020.

[26] Maha Al-Yahya, Auhood Al-Faries, and Remya George. Curonto: An ontological model for curriculum representation. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education, pages 358–358, 2013.

[27] Richard Wallis. May 14, 2019. Library metadata evolution: The final mile. Data Liberate. [Internet]. Available from: https://www.dataliberate.com/2019/05/14/library-metadata-evolution-final-mile/

[28] For example, *MBIB4600 – Interactive Information Retrieval (2021-V)*. Pattern: Course code (MBIB4600) – Course name (Interactive Information Retrieval) (Year (2021) – Semester (V).

About the Authors

David Massey is an Assistant Professor at the Department of Archivistics, Library and Information Science at Oslo Metropolitan University. He is a member of the Information Systems based on Metadata (METAINFO) research group.

Thomas Sødring is an Associate Professor at the Department of Archivistics, Library and Information Science at Oslo Metropolitan University. He is a member of the Information Systems based on Metadata (METAINFO) research group.

Subscribe to comments: For this article | For all articles