ROC data

ROC Dev Team

Sep 23, 2021

CONTENTS

The Repository of Organized Curriculums (ROC) server is an implementation of the ROC data model for digitally publishing curriculum standards documents used in the educational systems of different countries. To learn more about the project visit the website https://rocdata.global or read the report Digitizing Curriculum Standards to Unlock the Potential of Open Educational Resources in a Global Context.

CHAPTER

ONE

ROC DATA MODEL

The Repository Organized Curriculums (ROC) data model consists can be used to curriculum alignment data including curriculum standards, content correlations (content-standard links), and standards crosswalks (standard-standard links).

The summary section provides a condensed overview of all objects, with more details provided in following sections.

1.1 Summary

The following list contains all the different types of objects in the ROC data model. For each model, we provide links to the model definition (Django model class), auto-generated "rocdocs", schema notes, and an example.

- Jurisdiction: a country or an organization that publishes curriculum data. model, rocdocs, example.
- Vocabularies and terms: are used to represent constants within each a jurisdiction (e.g. academic subjects, and grade levels)
  - ControlledVocabulary: model, rocdocs, example.
  - Term: model, rocdocs, example.
- Standards:
  - StandardsDocument: model, schema, rocdocs, example.
  - StandardNode: model, schema, rocdocs, example.
- Content:
  - ContentCollection: schema, model, rocdocs, example.
  - ContentNode: schema, model, rocdocs, example.

The ROC data model can also represent the following types of relations:

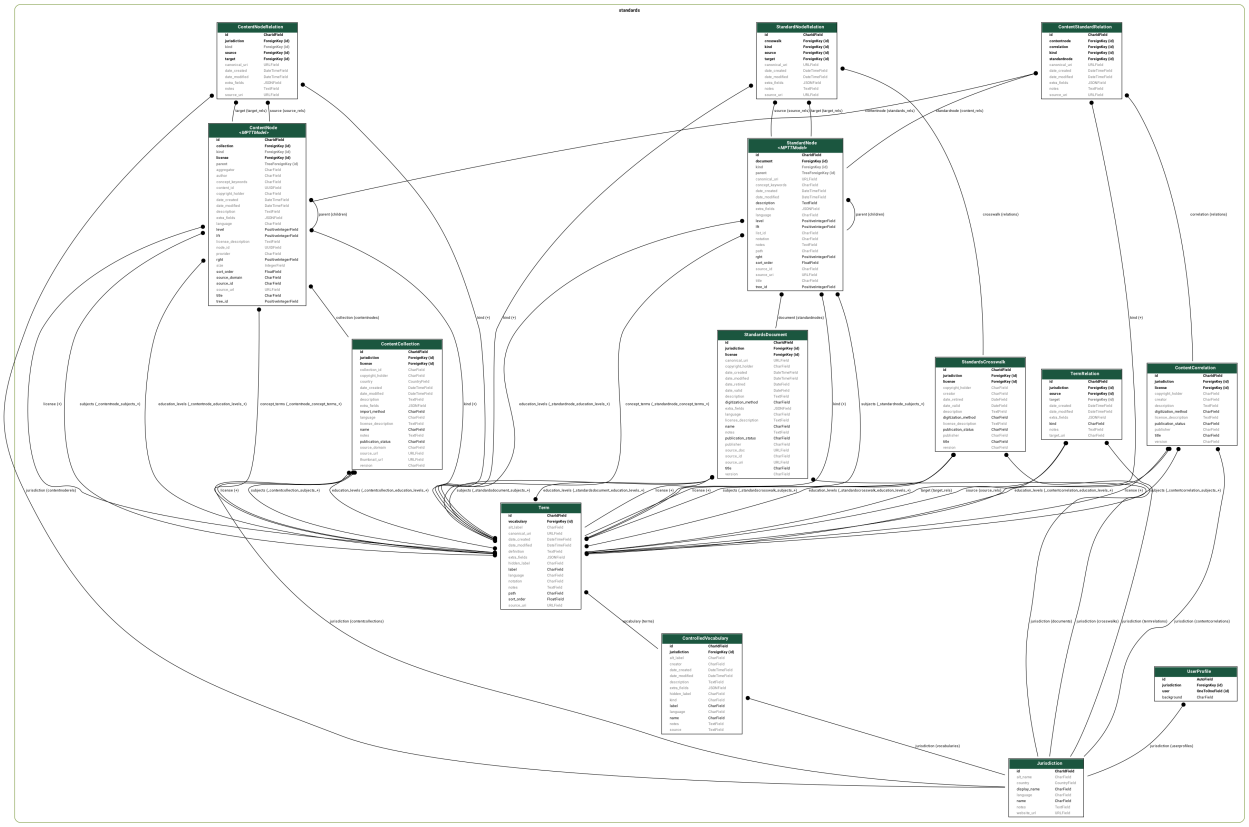
- TermRelation: model, rocdocs.
- ContentNodeRelation: model, rodocs.
- Content correlations: model, rodocs, example.
  - ContentStandardNodeRelation: schema, model, rocdocs, example.
- Standards crosswalks: model, rocdocs, example.
  - StandardNodeRelation: schema, model, rocdocs, example.

1.2 Prior work

The Repository of Organized Curriculums (ROC) data model is heavily inspired by the following prior work:

- ASN schema for curriculum standards and content correlations
- CASE schemas for curriculum standards
- LRMI specifications for content correlations
- Studio and Kolibri data model for content collections (channels) and content nodes

1.3 Overview of all models

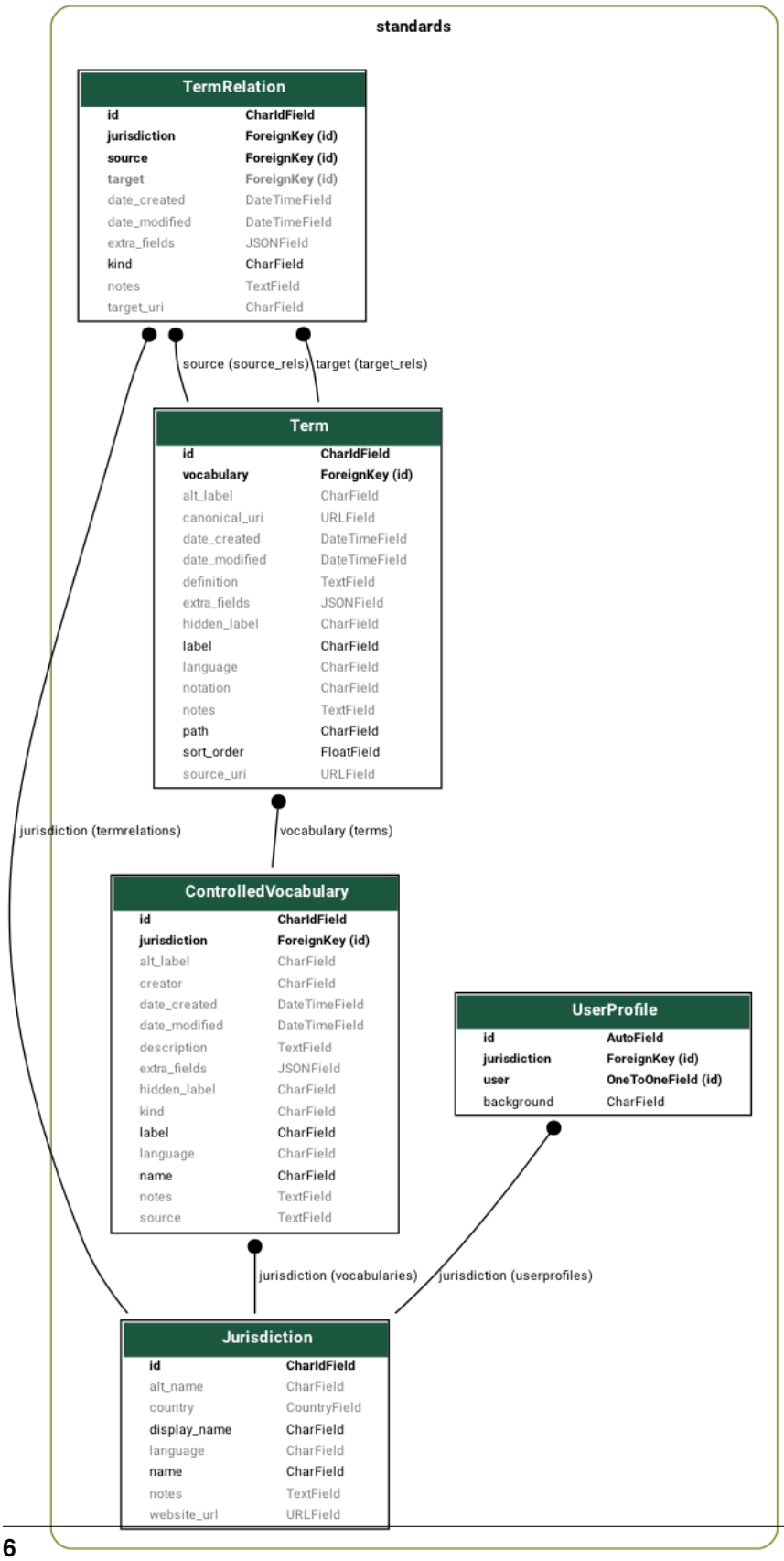


The combined data model of all

1.4 Jurisdictions

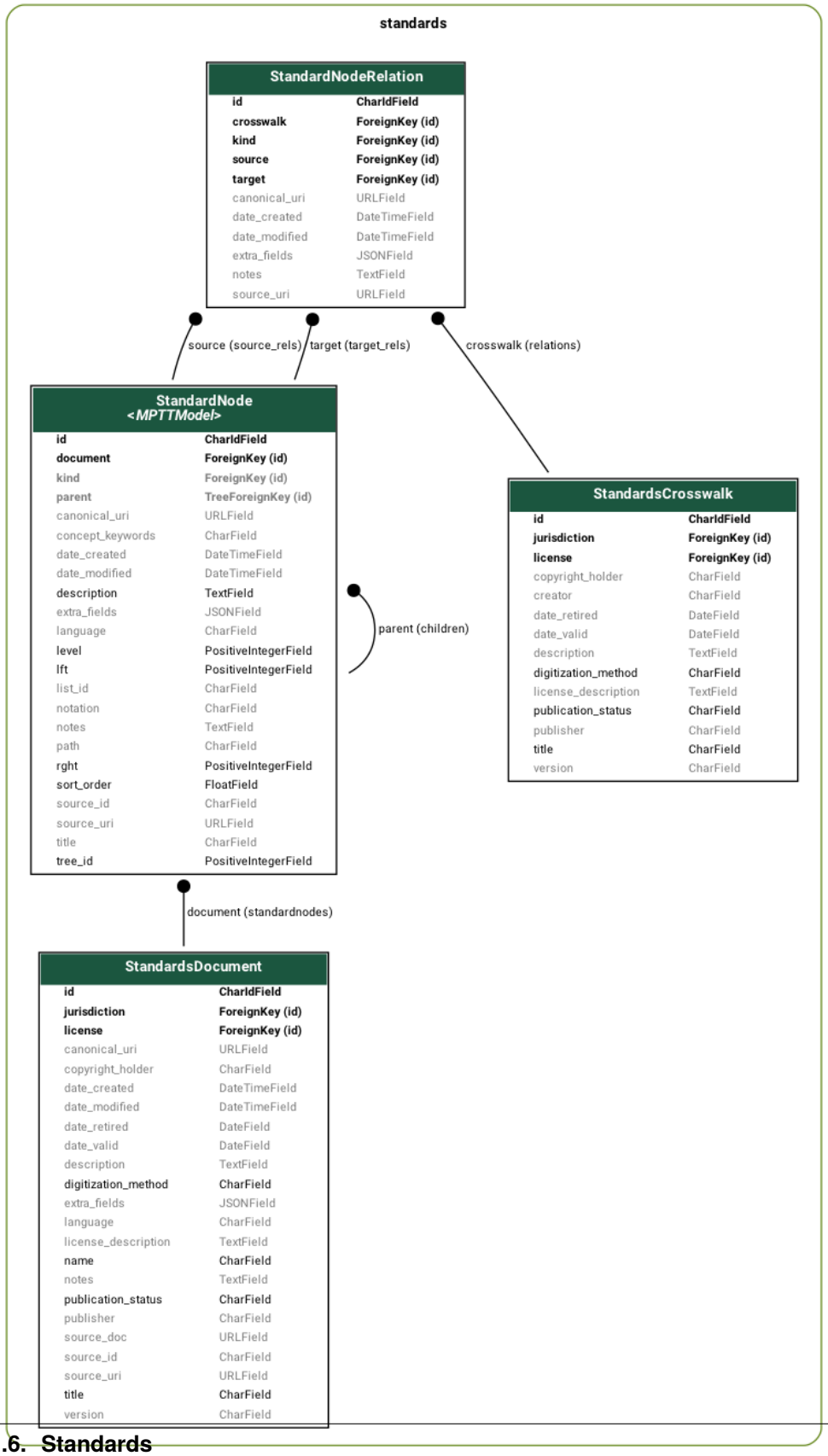
- Jurisdiction: a country or an organization that publishes curriculum data. model, rocdocs, example.

1.5 Controlled vocabularies and terms



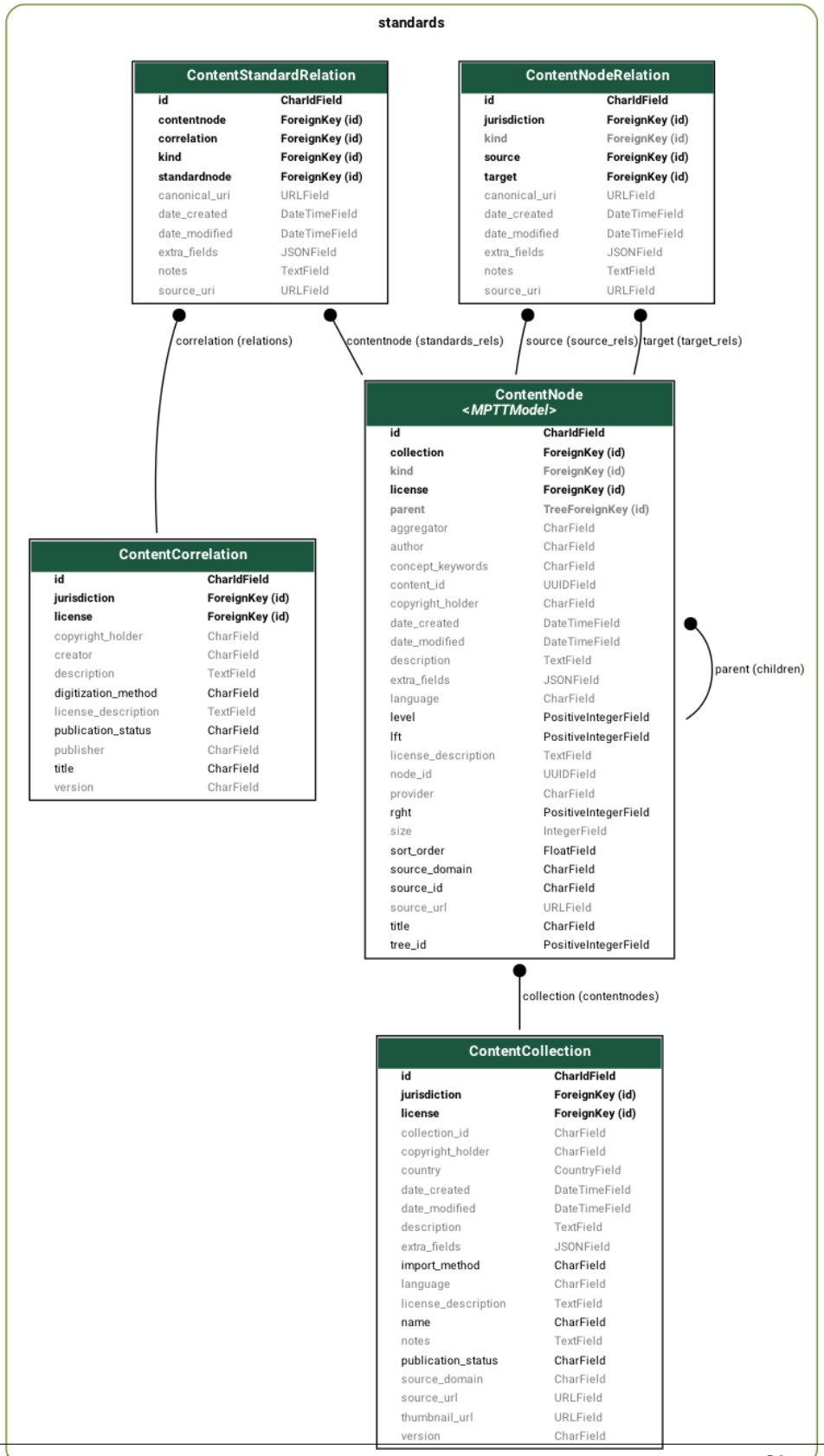
- Vocabularies and terms: are used to represent constants within each a jurisdiction (e.g. academic subjects, and grade levels)
  - ControlledVocabulary: model, rocdocs, example.
  - Term: model, rocdocs, example.
- TermRelation: model, rocdocs.

1.6 Standards



- StandardsDocument: model, schema, rocdocs, example.
  - StandardNode: model, schema, rocdocs, example.
- Standards crosswalks: model, rocdocs, example.
  - StandardNodeRelation: schema, model, rocdocs, example.

1.7 Content


standards
ContentStandardRelation
id CharIdField
contentnode ForeignKey (id)
correlation ForeignKey (id)
kind ForeignKey (id)
standardnode ForeignKey (id)
canonical_uri URLField
date_created DateTimeField
date_modified DateTimeField
extra_fields JSONField
notes TextField
source_uri URLField
ContentNodeRelation
id CharIdField
jurisdiction ForeignKey (id)
kind ForeignKey (id)
source ForeignKey (id)
target ForeignKey (id)
canonical_uri URLField
date_created DateTimeField
date_modified DateTimeField
extra_fields JSONField
notes TextField
source_uri URLField
correlation (relations)
contentnode (standards_rels)
source (source_rels)
target (target_rels)
ContentNode
<MPTTModel>
id CharIdField
collection ForeignKey (id)
kind ForeignKey (id)
license ForeignKey (id)
parent TreeForeignKey (id)
aggregator CharField
author CharField
concept_keywords CharField
content_id UUIDField
copyright_holder CharField
date_created DateTimeField
date_modified DateTimeField
description TextField
extra_fields JSONField
language CharField
level PositiveIntegerField
lft PositiveIntegerField
license_description TextField
node_id UUIDField
provider CharField
rght PositiveIntegerField
size IntegerField
sort_order FloatField
source_domain CharField
source_id CharField
source_url URLField
title CharField
tree_id PositiveIntegerField
parent (children)
ContentCorrelation
id CharIdField
jurisdiction ForeignKey (id)
license ForeignKey (id)
copyright_holder CharField
creator CharField
description TextField
digitization_method CharField
license_description TextField
publication_status CharField
publisher CharField
title CharField
version CharField
collection (contentnodes)
ContentCollection
id CharIdField
jurisdiction ForeignKey (id)
license ForeignKey (id)
collection_id CharField
copyright_holder CharField
country CountryField
date_created DateTimeField
date_modified DateTimeField
description TextField
extra_fields JSONField
import_method CharField
language CharField
license_description TextField
name CharField
notes TextField
publication_status CharField
source_domain CharField
source_url URLField
thumbnail_url URLField
version CharField

- ContentCollection: schema, model, rocdocs, example.
  - ContentNode: schema, model, rocdocs, example.
- ContentNodeRelation: model, rodocs.
- Content correlations: model, rodocs, example.
  - ContentStandardNodeRelation: schema, model, rocdocs, example.

CHAPTER

TWO

FORMATS

Each "resource" in the ROC server is accessible in several data formats. Using the example of the Term "Mathematics" within the Ghana:Subjects controlled vocabulary, https://rocdata.global/Ghana/terms/Subjects/Mathematics

- https://rocdata.global/Ghana/terms/Subjects/Mathematics = canonical URI of the resource, which returns different formats depending on request headers. By default, returns the `.html` format (see next).
- https://rocdata.global/Ghana/terms/Subjects/Mathematics.html = HTML browsing interface
- https://rocdata.global/Ghana/terms/Subjects/Mathematics.json = JSON (as used by frontend and apps)
- https://rocdata.global/Ghana/terms/Subjects/Mathematics.yaml = YAML [TODO]
- https://rocdata.global/Ghana/terms/Subjects/Mathematics.csv = CSV [TODO]

For standards documents and standard nodes, the following formats are can be implemented in the future based on user needs:

- `https://rocdata.global/Ghana/standardnodes/S12345678.asn.rdf`: ASN RDF graph
- `https://rocdata.global/Ghana/standardnodes/S12345678.case.json`: CASE JSON (tree of `CFItems`)

2.1 Bulk data exports

Other "bulk exports" formats are available for specific use cases like the development of machine learning algorithms for automated discovery of [content correlations][./automation/content_correlations_discovery.md] and *standards crosswalks*.

CHAPTER

THREE

DOMAIN REQUIREMENTS SPECIFICATION

The purpose of this document is to describe the needs and intended use cases for the Repository of Organized Curriculums (ROC) data model and the its reference implementation as the `rocserver` project.

3.1 Domain overview

For a detailed introduction to the domain of digital curriculum standards documents and their use in education, see the report "Digitizing Curriculum Standards to Unlock the Potential of Open Educational Resources in a Global Context".

3.2 Domain problems

The problems can be summarized as (1) lack of access to curriculum standards data in machine-readable formats, and (2) lack of tools for creating, storing, and exchanging curriculum alignment information.

3.2.1 Curriculum standards data problems

- Curriculum standards for most countries is only available in "analog formats" like print documents and PDFs.
- The lack of awareness by ministries of education and curriculum bodies about the benefits and use cases of digital curriculum standards.
- The lack of resources and expertise in MoEs and curriculum bodies to undertake new digital projects.

School administrators and teachers wishing to create standards-aligned learning experiences often have to transcribe curriculum standards documents into Excel sheets in order to use them as part of course preparation and lesson planning. Similarly, content creators and content repository administrators must undertake standards digitization process in order to extract the curriculum standards data from unstructured documents (print and PDF) and import it into their platforms.

The lack of curriculum standards information available in machine-readable form also poses a significant obstacle for the process of curriculum alignment (the cataloging of learning experiences according based on their relevance for the specific learning objectives specified in curriculum standards), which we will discuss next.

3.2.2 Curriculum alignment problems

- The utility of Open Educational Resources (OERs) is without if the cataloging work needed to organize them according to needs of the local educational system (subjects, grade level, topics, and individual learning objectives). The lack of catalog (curriculum alignment data) makes is difficult for librarians, teacher trainers, teachers, and students to find relevant resources.
- The task of curriculum alignment (cataloging of learning resources c1 c2 .. cN according to relevant curriculum standards sX1 sX2 .. sXn of country X) is requires curriculum expertise, context awareness and is very time consuming to carry out for large content collections.
- The task of curriculum alignment for ALL countries in the world is huge: consider large content collections can have N=10000 learning resources, and each the curriculum standards of each country can have n=1000 standards.
- Every learning platforms provides a different mechanism for representing content correlations (e.g. by assigning standards-alignment tags), and the process of curriculum alignment is often done manually (e.g. browse+search+add tags) through time consuming and error-prone workflows.
- There are no methods for exchange content correlations data between platforms.

3.3 Domain opportunities

3.3.1 Spreadsheets

- Government bodies can easily publish curriculum standards data as spreadsheets. Instead of publishing the curriculum standards spreadsheet as a excel table embedded in a Word document, converted to a PDF, just publish standards spreadsheet.
- Access to spreadsheets of curriculum standards information will be immediately useful for teachers (these are the tools of the trade they are most familiar with)
- Spreadsheets data in any format can be easily converted to ROC data format and imported into the `rocserver` by writing an "importer integration script" or by simply re-organizing the spreadsheet data to fit a provided ROC data template).
- The `rocserver` application allows all (1) curriculum standards, (2) content correlations, and (3) standards crosswalks data to be exported as various spreadsheet formats (CSV, .ods, .xlsx, .xls, gsheets, etc.)

3.3.2 Digital-first curriculum data

- Apps for teachers (browse local standard for country X and find relevant learning resources)
- Facilitate the adoption of new standards (e.g. new KICD CBC and Ghana 2019 standards) by making the information widely available coming from the authoritative source.

3.3.3 Easy data publishing in GitHub repositories

- Jurisdiction = GitHub repo
- Website as GitHub pages

3.3.4 Machine learning

- Recent advances in language models offer an opportunity to learn "deep" structure and the nuances of curriculum alignment task (semantic matching = the real goal), and baseline categorization methods based on keywords and old-school similarity metrics.

3.4 Publishing

3.4.1 Jurisdictions

A namespace that corresponds to some-real world region or organization, e.g., `Ghana`, `USA`, `KhanAcademy`, `LE`, etc. The `Global` jurisdiction is used as the namespace for ROC data model constants (e.g. `digitzation_methods`, `content_kinds`, `publication_statuses`, etc.).

3.4.2 Controlled vocabularies and terms

The digital representation of curriculum standards metadata types described below is based on terms chosen from controlled vocabularies defined within the context of a jurisdiction. All examples show in this section assume jurisdiction=Ghana.

When describing the standard statement with notation `B4.1.3.1` within the Ghanian math curriculum standards, we would like to indicate it is part of the "Basic 4" grade level, but instead of using a string value we will use a URI property. The Uniform Resource Identifier (URI) value `https://rocdata.global/Ghana/terms/GradeLevels/B4` is an example of one such identifiers, specifically the identifiers for the "Basic 4" grade level within the Ghana educational system. In this URI, `https://rocdata.global` is the server hosting the controlled vocabulary, `Ghana` is the jurisdiction name, `GradeLevels` is the name of the controlled vocabulary, and `B4` is the term name. Using URIs as property values provides the following affordances for data consumers:

3.4.3 Standards

Curriculum standards documents and the individual standards nodes they consist of.

3.4.4 Content

Content collections and content nodes they consist of. We need to have a way to refer to individual resources, so the data model assumes `source_url` present at the minimum, with preference for additional metadata like `source_domain` and `source_id`.

3.4.5 Content correlations

Sets of content-standard associations that indicate a content resource is useful, relevant, or related to a specific educational standard node (an element of a standards document).

CHAPTER

FOUR

SOFTWARE DESIGN SPECIFICATION

The purpose of this page is to give a high level overview of the technical design decisions that went into the Repository of Organized Curriculums (ROC) data model and its reference implementation rocdata.global.

4.1 Tech stack

4.1.1 Graph data as SQL

4.1.2 Django web frameworks

4.1.3 MPTT Trees

The Note this is a logically "hidden" node that should not be visible to end users of the ROC server.

4.1.4 Linked data

4.2 Jurisdictions

A namespace that corresponds to some-real world region or organization, e.g., `Ghana`, `USA`, `KhanAcademy`, `LE`, etc. The `Global` jurisdiction is used as the namespace for ROC data model constants (e.g. `digitzation_methods`, `content_kinds`, `publication_statuses`, etc.).

4.3 Controlled vocabularies and terms

The digital representation of curriculum standards metadata types described below is based on terms chosen from controlled vocabularies defined within the context of a jurisdiction. All examples show in this section assume jurisdiction=Ghana.

When describing the standard statement with notation `B4.1.3.1` within the Ghanian math curriculum standards, we would like to indicate it is part of the "Basic 4" grade level, but instead of using a string value we will use a URI property. The Uniform Resource Identifier (URI) value `https://rocdata.global/Ghana/terms/GradeLevels/B4` is an example of one such identifiers, specifically the identifiers for the "Basic 4" grade level within the Ghana educational system. In this URI, `https://rocdata.global` is the server hosting the controlled vocabulary, `Ghana` is the jurisdiction name, `GradeLevels` is the name of the controlled vocabulary, and `B4` is the term name.

4.4 Standards documents and standard nodes

4.5 Content collections and content nodes

4.6 Content correlations

4.7 Standards crosswalks

CHAPTER

FIVE

PUBLISHING

5.1 Publishing context

When exporting data, external resources are identified by their `canonical_uri`.

The `uri` and `canonical_uri` for internal URIs is computed depending on the "publishing context" for the jurisdiction, which determines the appropriate hostname for URIs. The different publishing contexts are described below.

5.1.1 Default publishing context

The publishing context `default` corresponds to current hostname, which is determined dynamically at request time. Usually `http://localhost:8000` or `http://127.0.0.1:8000`.

5.1.2 Standards server

The publishing context `rocserver` corresponds to the hostname: `https://rocdata.global`.

5.1.3 Github pages

The publishing context `githubpages` with can be used when the official data source for a given jurisdiction is published to a github pages website.

5.1.4 w3id.org

The publishing context `w3id.org` is used to assign URIs that start with hostname `https://w3id.org` which in turn redirect to another server or github page. For maximum flexibility, `https://w3id.org` URIs will be used as canonical URIs. See https://github.com/perma-id/w3id.org for more info about the redirect service.

When creating the export (static site generator mode), the hostname for URIs is determined by HOST headers by the HTTP client and may need to be manually modified.

5.2 Publishing settings

The publishing context name is controlled by the `settings.ROCDATA_PUBLISHING_CONTEXT`, which in turn is set from the ENV variable `ROCDATA_PUBLISHING_CONTEXT`, or set to the default value `default`.

The dictionary `settings.ROCDATA_PUBLISHING_CONTEXTS` provides data for all each publishing context.

5.3 Publishing context API

The helper method `standards.publishing.get_publshing_context` returns the dictionary containing the info like this:

```
{
    "scheme": "https",
    "netloc": "https://w3id.org",
    "path_prefix": "/rocdata",
}
```

The helper function `standards.publishing.build_absolute_uri(path, publishing_context=None, request=None)` can be used to obtain the absolute URI of for any path `path`, in the publishing context `publishing_context` (if not provided, the default publishing context is used). The keyword argument `request` is required for the default context.

CHAPTER

SIX

IMPORTERS

Importing data into the Repository of Organized Curriculums (ROC) is accomplished by using one of the following "importer" methods.

6.1 Controlled vocabulary terms

Can be loaded using individual commands like:

```
./manage.py createjurisdiction --name Global --display_name "Global Terms" --language "en
↪"
./manage.py loadterms data/terms/ContentRelationKinds.yml
```

but too numerous...

Instead load using fab command:

```
fab load_terms
```

which will create all relevant jurisdictions and load all controlled vocabularies from the corresponding GitHub repositories. See `fabfile.py` source code for details.

6.1.1 Examples terms YAML data

- Global:ContentStandardRelationKinds.yml
- Ghana:GradeLevels.yml

6.1.2 Create controlled vocabularies and terms

1. Create GitHub repo where you will store the data.
2. Add folder called `terms/` in the repo.
3. Add YAML data file in format similar to the examples shown above.

You can now use the command `./manage.py loadterms` <URL> to import the controlled vocabulary data into your local `rocserver` instance, where <URL> is the full path of the "raw" file hosted on GitHub.

6.1.3 Uploading controlled vocabularies and terms using a spreadsheet

1. Prepare data using the spreadsheet template TODOLINK
2. Upload data using the form at TODOLINK
3. Verify and review uploaded data was correctly parsed and validated. Go back to step 1 if something doesn't look right.
4. Change status to `publicdraft` so other users will be able to view the data.

6.2 Standards spreadsheets

6.2.1 Uploading standards using a spreadsheet

1. Prepare data using the spreadsheet template TODOLINK
2. Upload data using the form at TODOLINK
3. Verify and review uploaded data was correctly parsed and validated. Go back to step 1 if something doesn't look right.
4. Change status to `publicdraft` so other users will be able to view the data.

6.3 Content correlations spreadsheets

6.3.1 Uploading content correlations using a spreadsheet

1. Obtain the list of identifiers for the standards document and content collections you will be correlating.
2. Prepare data using the spreadsheet template TODOLINK
3. Upload data using the form at TODOLINK
4. Verify and review uploaded data was correctly parsed and validated. Go back to step 1 if something doesn't look right.
5. Change status to `publicdraft` so other users will be able to view the data.

6.4 Standards crosswalks spreadsheets

6.4.1 Uploading standards crosswalks using a spreadsheet

1. Obtain the list of identifiers for the standards documents you will be aligning.
2. Prepare data using the spreadsheet template TODOLINK
3. Upload data using the form at TODOLINK
4. Verify and review uploaded data was correctly parsed and validated. Go back to step 1 if something doesn't look right.
5. Change status to `publicdraft` so other users will be able to view the data.

6.5 Kolibri channel content collections

A Kolibri channel consists of two parts:

- Metadata stored in sqlite3 DB file. The channel DB file can downloaded from URL like `/content/databases/{{channel_id}}.sqlite3` from any instance of the Kolibri application, Kolibri Studio, or obtained through direct file transfer.
- A collection of files stored in `/content/storage/{x}/{y}/{xyz.......}.ext`

For the purpose of importing Kolibri channels as content collections, we need to import only the content metadata from the Kolibri database file.

6.5.1 Usage

1. Obtain the `channel_id` of the channel you want to import. You can find the a channel's ID from the URL when viewing the channel in Kolirbi or Kolibri Studio.
2. Run the script in the rocdata/contentcollections-kolibri repository to obtain the kolibritree JSON dump of the channel database: `./kolibri_db/reader.py --channel_id=<channel_id>`
3. Import the channel as a content collection using the rocserver management command:

```
./manage.py ccimport_kolibri  \
   --jurisdiction LE \
   --country US \
   --name <shortname> \
   --source_domain <sourcewebsiteurl> \
   --source_url <weburlofcollection> \
   --kolibritree_url=<urlwherejsoncanbedownloadedfrom>
```

   where `<shortname>` must be a unique, short identifier for the collection, e.g. `KA-en`.

The following properties of the content collection will be set by default:

- `collection_id`: set to `channel_id`
- `version`: taken from imported JSON [TODO]
- `publication_status="publicdraft"`. This can be chanced to "public" through admin.
- `subjects=[]`: can add subject references (ManyToManyField relations to terms in vocabularies of kind `subjects`.
- `education_levels=[]`: can add grade levels references (relations to terms in vocabularies of kind `education_levels`.

6.6 Khan Academy content collections

Content collections imported from the Khan Academy TSV exports bucket.

The Khan Academy collection has been translated in dozens of languages. The math and science resources in certain languages are organized according to the local curriculum structure (localized topic trees) of the following countries: Bangladesh, Belgium, Brazil, Bulgaria, France, India, Mexico, Peru, and USA.

WIP; STAY TUNED

6.7 Kolibri Studio channel content collections

Kolibri Studio is a general purpose editor for Kolibri channels, which allows curriculum designers and educators to retain, reuse, revise, remix, and redistribute the educational learning resources available in the Kolibri Content Library.

Kolibri Studio is often used to for curriculum alignment of learning resources, by re-organizing them into a folder structure (topics and subtopics) that matches the structure of local curriculum standards in a given country.

The structure of content of a Studio channels is similar to the information available in a Kolibri database, but contains additional information about content provenance (where was the content was imported form), and `source_domain` and `source_id` properties for each content node, and `source_url` fields for each file.

CHAPTER

SEVEN

EXPORTERS

The data in the Repository of Organized Curriculums (ROC) can be exported in numerous formats, depending on the use cases and external integration needs.

7.1 Browsable HTML pages

See https://rocdata.global/.

7.2 JSON Data

Every resource on a ROC data server is accessible as JSON data. For example the Term "Mathematics" within the Ghana:Subjects controlled vocabulary identified by the URI https://rocdata.global/Ghana/terms/Subjects/Mathematics (official identifier), can be accessed as JSON data at https://rocdata.global/Ghana/terms/Subjects/Mathematics.json for use in frontend applications and mobile apps.

7.2.1 Usage

Just append `.json` to any canonical URI within the ROC server to obtain the JSON representation of this resource.

7.3 ASN Data

The Achievement Standards Network (ASN) framework is meta data model for machine-readable representations of competencies and standards statements published by education agencies and other organizations.

See http://asn.jesandco.org/content/technical-documentation.

`exporter_asn` is WIP

7.4 CASE Data

The Competencies and Academic Standards Exchange (CASE) standard has widespread adoption in the U.S. and has a well defined spec, see http://www.imsglobal.org/activity/case.

`exporter_case` is WIP

CHAPTER

EIGHT

ROC API

The purpose of page is to give a high level overview of the ROC server API.

8.1 Jurisdictions

A namespace that corresponds to some-real world region or organization, e.g., `Ghana`, `USA`, `KhanAcademy`, `LE`, etc.

- Examples https://rocdata.global/Ghana and https://rocdata.global/KA .

The `Global` jurisdiction, https://rocdata.global/Global , is used for ROC data model constants (e.g. `digitzation_methods`, `content_kinds`, `publication_statuses`).

TODO: figure

8.2 Controlled vocabularies and terms

The digital representation of curriculum standards metadata types described below is based on terms chosen from controlled vocabularies defined within the context of a jurisdiction. All examples show in this section assume jurisdiction=Ghana.

- Browse https://rocdata.global/Ghana/terms : all controlled vocabularies define within the Ghana jurisdiction
- Browse https://rocdata.global/Ghana/terms/GradeLevels : the Ghana grade levels vocabulary, see also standards-ghana/terms/GradeLevels.
- Browse https://rocdata.global/Ghana/terms/GradeLevels/B4 : a webpage with human-readable info about the term "Basic 4"
- GET https://rocdata.global/Ghana/terms/GradeLevels/B4.json : metadata for term `B4` as JSON

8.3 Term relations

`{juri}/termrels/{jurisdiction}/{termrel.id}`

8.4 Standards documents and standard nodes

{juri}/documents/{document.id}

{juri}/standardnodes/{snode.id}

8.5 Standards crosswalks

{juri}/standardscrosswalks/{sc.id}

{juri}/standardnoderels/{stdrel.id}

8.6 Content collections and content nodes

{juri}/contentcollections/{cc.id}

{juri}/contentnodes/{contentnode.id}

{juri}/contentnoderels/{cnode.id}

8.7 Content correlations

{juri}/contentcorrelations/{cs.id}

{juri}/contentstandardrels/{csr.id}

CHAPTER

NINE

URIS

There are several different types of URIs that exist in the system:

- Internal URIs: for objects that were created on the standards server and thus have `self.canonical_uri == self.get_absolute_url()`.
- Mirrored external URIs: vocabulary terms or standards whose official "home" is located on an external server, but which have been imported to local server. For these `self.canonical_uri == self.source_uri`. Since these resources are also "mirrored" locally, they can be browsed at `self.get_absolute_url()`.
- External URIs: these are references to external resources that cannot be browsed locally, e.g., `target_uri` as part of a relation when `target=None`.

9.1 Browsing context

When accessing a live server instance, navigation links are computed using the resources' `get_absolute_url()` methods and not using `canonical_uri`s.

9.2 Publishing context

See page on *publishing* for details about publishing contexts.

9.3 Resolving URIs

When encountering a URI reference to a resource, we follow the same process as during publishing in order to find the referenced object, including internal URIs, mirrored external URIs, and external URIs.

9.4 Content URLs

Content collections and content nodes are identified by a `source_url`, which is usually represents a an online location where the resource can be accessed and downloaded from.

9.5 Content IDs

Content collections and content nodes are identified by a `source_domain` which represents the hostname where one or more content collections are hosted. Furthermore, `colletion_id`, `source_id`, `node_id` are also used to identify content nodes.

CHAPTER

TEN

URL PATTERNS

The Django REST Framework "format suffix patterns" pattern allows us to handle paths with format extensions like `/terms/Ghana/GradeLevels/B2.json` automatically.

To see this magic, add the following lines to the bottom of `standrads-server/urls.py`:

```
for urlp in urlpatterns:
    print(urlp)
```

Here is the example debug output:

```
<URLPattern 'terms/' [name='api-juri-list']>
<URLPattern 'terms<drf_format_suffix_json_html:format>' [name='api-juri-list']>
<URLPattern '^terms/(?P<name>\w*)$' [name='api-juri-detail']>
<URLPattern '^terms/(?P<name>\w*)\.(?P<format>(json|html))/?$' [name='api-juri-detail']>
<URLPattern 'terms/<name>/' [name='api-juri-vocab-list']>
<URLPattern 'terms/<name><drf_format_suffix_json_html:format>' [name='api-juri-vocab-list
↪']>
<URLPattern '^terms/(?P<jurisdiction__name>\w*)/(?P<name>\w*)$' [name='api-juri-vocab-
↪detail']>
<URLPattern '^terms/(?P<jurisdiction__name>\w*)/(?P<name>\w*)\.(?P<format>(json|html))/?$
↪' [name='api-juri-vocab-detail']>
<URLPattern 'terms/<jurisdiction__name>/<name>/' [name='api-juri-vocab-term-list']>
<URLPattern 'terms/<jurisdiction__name>/<name><drf_format_suffix_json_html:format>'␣
↪[name='api-juri-vocab-term-list']>
<URLPattern '^terms/(?P<vocabulary__jurisdiction__name>\w*)/(?P<vocabulary__name>\w*)/(?P
↪<path>[\w/]*)$' [name='api-juri-vocab-term-detail']>
<URLPattern '^terms/(?P<vocabulary__jurisdiction__name>\w*)/(?P<vocabulary__name>\w*)/(?P
↪<path>[\w/]*)\.(?P<format>(json|html))/?$' [name='api-juri-vocab-term-detail']>
<URLResolver <URLPattern list> (admin:admin) 'admin/'>
```

CHAPTER

ELEVEN

DESIGN LIMITATIONS

The data model we develop for curriculum documents borrows ideas from the linked data universe: controlled vocabularies, and using URIs as properties, etc. For the sake of simplicity of operation we choose to adopt only a subset of the representation linked data and do not implement We do not implement any FULL

The data model has the following representation limitations compared to RDF:

- Terms with multiple `alt_label` cannot be faithfully represented.
-

CHAPTER

TWELVE

AUTOMATION

One of the main objectives the Repository of Organized Curriculums is to enable the automated dicovery of content correlations: given a curriculum standard document (a tree of standard nodes) and a collection of learning resources, use machine learning algorithms to find which content nodes are relevant for each standard node.

12.1 Machine learning tasks

This document describes machine learning tasks inference metadata discovery tasks based on ROC data available thorough rocdata.global.

The end-goal to be able to categorize educational resources (content collections) according to their relevance for local curriculum standards (standards nodes). You can think of the end-user as a teacher in country X that needs to find relevant learning resources and create a lesson. Starting from a un-categorized collection of learning resources is too time consuming and difficult, but if the same resources are categorized according to the local curriculums standards of country X, then the teacher will find relevant resources much more easily.

12.1.1 Tasks

We have identified two specific tasks (ML challenges) related to the overall goal:

- Content correlations discovery
- Standards crosswalk discovery

12.1.2 Prior work

The links below represent a non-exhaustive list of ML research related to the general domain of automated discovery of content correlations and standards crosswalks:

- 2010: The paper "Computer-Assisted Assignment of Educational Standards Using Natural Language Processing" by Devaul, Diekema, and Ostwald describes an approach for a cataloging tool that aids catalogers in the assignment of standards metadata to digital library resources based on natural language processing techniques.
- 2017-2019: multiple consultations and events including educators, curriculum designers, ministries of education, platform developers, machine learning experts, and other key stakeholders from the educational domain with a common interest to make relevant learning resources accessible to teachers and learners in low-resource contexts.
- October 2019: the San Francisco hackathon on automation of curriculum alignment was held that included a prototypes of standards crosswalks discovery task. Read the hackathon report for additional info and links to relevant GitHub repositories, watch the video, and learn about participants' reflections.
  - Starter code:

  - Example colab notebook:
  - Human-judgment user interfaces:
  - Related-standards browsing interfaces:

- January 2021: ROC data report "Digitizing Curriculum Standards to Unlock the Potential of Open Educational Resources in a Global Context," which outlines the use cases for digital curriculum standards for a non-technical audience, and defines data model for curriculum documents, content correlations data, and standards crosswalks data.

12.1.3 Get involved

All datasets and models developed as part of this collaboration have been released as public goods (open source) on GitHub. Feel free to explore the available data, and code samples, and be on the lookout for ML challenges and organized events in the coming year.

12.2 Content correlations discovery

Automated content-standard link discovery (a.k.a content correlations).

12.2.1 Task CS definition

Given a subset of the content collections (imports from OER content repositories) and a subset of curriculum standards statements (as set of standard nodes), discover which content nodes are relevant for each standard node.

Inputs: content subset `cc1[subset1]` and standards subset `dx[subsetdx]`, where `cc1` is some ROC data content collections, and `dx` is some ROC document.

Outputs: ContentStandardNodeRelation list: `[(ca, cskind, sx), ...]` consisting of content-to-standard links of type `csking` between subset of standards and the subset of content nodes specified in the input.

12.2.2 Data

The following relevant ROC data is available for use with this task:

- Data from `ContentCollection`s that consist of `ContentNode` trees. There exist **O(100k)** content nodes organized into content collections like `khanadademy-en`, `kolibri-channel-ck12`, `kolibri-channel-ghana-math`, etc. Each content node has a title, description, source_url, and other metadata.
- Data from `StandardsDocument`s that consist of `StandardNode` trees. There exist **O(10)** jurisdictions (Brazil, Ghana, Honduras, Kenya, UK, USA, Zambia) for which curriculum standards documents are available in machine-readable form and within each jurisdiction **O(10)** standards documents, with each document containing **O(100)** standard nodes. Each standard node has a description (str) that specifies a particular set of competencies expected of learners for a given grade level, within a particular academic subject. Standard nodes can be folder-like (intermediate levels of the hierarchy) or a atomic statements (leaf nodes).
- Existing content correlations `ContentCorrelation`s that consist of multiple content-to-standard links (`ContentStandardRelation`s) available in several jurisdictions (e.g. Khan Academy (`KA`) and Learning Equality `LE`).

12.2.3 Evaluation metrics

The "objective quality" of the output can be measured using precision and recall metrics evaluated against the "ground truth" content correlations produced by human experts (curriculum experts, librarians, and educators) for same task of identifying relevant correlations between content collection subset `cc1[subset1]` and standards document subset `dx[subsetdx]`:

- **Precision**: what proportion of the `[(ca, cskind, sx), ...]` in the output were also identifier by human experts for same task.
- **Recall**: what proportion of the `[(ca, cskind, sx), ...]` identified by human experts are present in the output.

12.2.4 Challenges

The problem of discovering correlated learning resources for standard nodes is complicated by the following nuances of the task:

- A vocabulary gap exists between the language used in standard node descriptions and the language used in educational resources titles and descriptions. Standard node descriptions tend to be short, high-level abstract statements about what students should know, do, understand, etc. whereas learning resources use language relevant for concrete instances of these skills. This difference in conceptual level of the text descriptions makes it difficult to find commonalities between standard and resource when using "keyword match" techniques (same problem exists for manual alignment efforts based on search).
- Determining the "alignment" between a learning resource and a curriculum standard is a nuanced, context-dependent, and multi-faceted process. The catalogers working on content correlations must:
  - (S) know the curriculum standards, the cultural and pedagogical context of the teachers and learners who are the target users of the aligned-content
  - (C) know the content resources content, topics, and pedagogical approach, in order to make a correct judgement about the relevance and usefulness of each content resource to the target curriculum standards and educational context. The need for this "deep checks" for each content node is partially what makes the curriculum alignment task so time consuming.

12.3 Standards crosswalk discovery

Knowing the equivalencies and similarities between curriculum standards in different countries will allow content correlations to be reused between countries.

12.3.1 Task definition

Given a subset curriculum standards statements in Jurisdiction X (as set of standard nodes), and a subset of the curriculum standards in Jurisdictions Y (another set of standard nodes), discover all alignments between standard node, but identifying standards statements that describe the same knowledge, competencies, or learning objectives.

Inputs: standards subsets `dx[subsetdx]` and `dy[subsetdy]`, where `dx` is a ROC curriculum document defined in jurisdiction X, and `dy` is a ROC curriculum document defined in jurisdiction Y.

Outputs: a list of `ContentStandardNodeRelations`, `[ (sx, srkind, sy), ...]` consisting of standard-to-standard links of type `drkind` between a subset of the standards nodes specified in the inputs `dx[subsetdx]` and `dy[subsetdy]`.

12.3.2 Data

The following relevant ROC data is available for use for this task:

- Data from `StandardsDocument`s that consist of `StandardNode` trees
- Data from `StandardsCrosswalk`s consisting of `StandardNodeRelation` that define standard-to-standard alignments relations.
- Data from `ContentCollection`s that consist of `ContentNode` trees. There exist **O(100k)** content nodes organized into content collections like `khanadademy-en`, `kolibri-channel-ck12`, `kolibri-channel-ghana-math`, etc. Each content node has a title, description, source_url, and other metadata.
- Data from `StandardsDocument`s that consist of `StandardNode` trees. There exist **O(10)** jurisdictions (Brazil, Ghana, Honduras, Kenya, UK, USA, Zambia) for which curriculum standards documents are available in machine-readable form and within each jurisdiction **O(10)** standards documents, with each document containing **O(100)** standard nodes. Each standard node has a description (str) that specifies a particular set of competencies expected of learners for a given grade level, within a particular academic subject. Standard nodes can be folder-like (intermediate levels of the hierarchy) or a atomic statements (leaf nodes).
- Existing content correlations `ContentCorrelation`s that consist of multiple content-to-standard links (`ContentStandardRelation`s) available in several jurisdictions (e.g. Khan Academy (`KA`) and Learning Equality `LE`).

12.3.3 Evaluation metrics

The "quality" of the output is measured using standard precision and recall metrics evaluated against the ground truth provided by human experts (a curriculum developer, alignment consultants, or other curriculum experts) who produce standards crosswalk based on the same inputs `dx[subsetdx]` and `dy[subsetdy]`.

- Precision: what proportion of the `[(sx, srkind, sy), ...]` in the output were also identifier by human experts for same task.
- Recall: what proportion of the `[(sx, srkind, sy), ...]` identified by human experts are present in the output.

12.3.4 Challenges

One concern/limitation about the overall goal of using standards crosswalks to "port" content correlations data between different educational contexts, is the "compounding of inaccuracy" aspect of alignment relations:

- If `(Lesson)--[lrmi:teaches]->(StdX.x)` is an 80% match, and `(StdX.x)--[asn:narrowAlignment]->(StdY.y)` is also 80% accurate, then the combined two-hop graph traversal will only be ~60% accurate.

This is why it's important to think about the semi-automated workflow strategies based on graph data as recommendations that need to be vetted by humans in the loop (curriculum experts that know about the nuances of alignment work who can accept/reject these recommendations). Still though, if we can use classical NLP and the latest language models to give curriculum experts (and teachers, and learners) a "shortlist" of 10-100 content correlations recommendations based on the graph, this will majorly improve their work (otherwise they have to wade through **O(100k)** learning resources, and must fallback on generic keyword search tools, which are known to have limitations for this task).

CHAPTER

THIRTEEN

ROADMAP

This page lists various TODOs and next steps for the development of the ROC data model and the `rocserver` application.

13.1 Standard node components

Curriculum standard statements often contain additional information like:

- **Teaching time allocation**: guidance about how much class time to dedicate to each part of the curriculum
- **Teaching strategies / Suggested instructional methods**: similar to the above, but with suggestions targeting teachers
- **Content exemplars**: examples of text, images, equations, and formulas that illustrate the concepts students should be learning about
- **Content references**: reference to a specific textbooks or external learning resources
- **Suggested learning activities**: examples of recommended classroom activities that teachers can use to teach the given standard entry
- **Assessment notes**: description or examples of assessment items that can be used to evaluate a learner's knowledge on this standard entry
- **Connections to earlier entries in a progression**: vertical alignment information
- **Connections to other subjects in current grade**: horizontal alignment information
- **Practices / Core ideas / Cross-cutting concepts**: information about aspects of the curriculum that are not captured by the primary hierarchy of the document
- **Benchmarks / Rubrics**: specific criteria used to evaluate the attainment level of the competency described by the standard statement
- **Inquiry Questions**: key questions for organizing classroom discussions
- **Notes**: clarifications, additional information, and non-statutory guidance which can provide useful information about scope and emphasis

Standards documents also often contain sections with general information like:

- Notes about overall progressions between grades
- Cross-curricular competencies (e.g. problem solving, critical thinking)
- General outcomes for the whole educational program
- Values and principles of the educational system

These sections of the curriculum document can be very helpful for understanding the standards since they provide the much needed interpretation context, however these types of additional information differ widely between the standards in different countries, so they have not been included in the initial work on the ROC data model.

13.1.1 Next steps

- Add an new model `StanrardNodeComponent` for storing additional information attached to any `StandardNode`. A standard component's `content` field must be able to support rich text features (formatted text, images, tables, equations, etc.). Each component has a different `kind` based a local controlled vocabulary.
- Update the web-browsing interface and data exports to display and include the standards components when displaying standard nodes.

13.2 Standard node sets

Within each standards document, we can identify subsets of the standards that are relevant for educators teaching a course at a given grade level. Thinking from the point of view of a teacher in charge of Grade 4 math in Country X, the only info this teacher is interested in is the subset `Country X > Math > Grade 4` and the topics and standards statements contained therein. The standards for other grades are not relevant for their day-to-day activities like preparation of course plans, lesson plans, and choosing learning resources and teaching strategies to use with their students.

The top-levels of a standards document hierarchy are rarely meaningful and documents in different countries follow can have different nesting structure. Examples of hierarchies for the "core properties" (subject, grade level, topic) include:

- `subject > grade level > topic`: a document whose sections describes standards for different subjects, with subsection corresponding to different grade levels, and subsubsections corresponding to educational topics (e.g. Algebra).
- `grade level > subject > topic`: a document subdivided by grade levels then by academic subject (e.g. Math), and topic (e.g. Algebra).
- `subject > topic > grade level`: a hierarchy in which a given topic appears at multiple grade levels.

In order to help teachers navigate the standards data, it would be helpful to add "shortcuts" to specific subtrees of the standards document hierarchy, each subset being described by a given (subject, grade levels, topic). We will refer to these grade-level-, subject-, and topic-level subsets of a standards document as `StandardSet`s. We can think of a standard set as a "symlink" to a particular place within a standards document—a way to navigate standards document that is most useful in practice for teachers, because they don't care about the overall document, but only about a specific (grade level, subject, topic) subset.

A good example of such teacher-first organizational structure can be seen in the mapping performed by the importer scripts of the common standards project, which process source documents and organize them into standard sets useful for teachers.

13.2.1 Use cases for standard sets

- As a teacher, I can easily drill-down to the subset of a standards document relevant for my needs.
- Identify meaningful subsets of standards documents to use for human-powered digitization and curriculum alignment efforts (multiple curriculum experts working in parallel on different subsets of a document). The same subsets can also be used for automation workflows.
- Different versions of curriculum standards are identified by including the year of publication in the `StandardSet` title, e.g. `Kenya, Math, Grade 4 (2002)` (the old standards) vs. `Kenya, Math, Grade 4 (2019)` (the new CBC standards).

- Educators can easily find a list of all standards sets for a particular grade level (e.g. a teacher looking to add cross-curriculum links in their lessons).

13.2.2 Next steps

- Add the `StandardSet` model that "points" to a `StandardsDocument` and a `StandardNode` within that document.
- Update the standards document web-browsing interface and data exports to display standard sets shortcuts.

13.3 Search

Frontend applications based on the `rocdata.global` web service would benefit from access to a search interface for standards documents, standard nodes, content nodes, and the relations between them (content correlations and standards crosswalks).

13.3.1 Next steps

- Implement search functionality through a new endpoint
- Provide sample code for using search endpoint
- (stretch goal) Implement fulltext search within content nodes

CHAPTER

FOURTEEN

ROC DATA MODEL

- About
- Glossary
- Requirements Specification
- Report TODO

CHAPTER

FIFTEEN

REFERENCE

- Data model
- Data formats
- Technical design
- Publishing
- API
- URIs

CHAPTER

SIXTEEN

IMPORTERS

- Vocabularies and Terms
- Standards spreadsheet
- Content correlations spreadsheets
- Standards crosswalks spreadsheets
- Kolibri DB
- Kolibri Studio DB
- Khan Academy

CHAPTER

SEVENTEEN

EXPORTERS

- HTML pages
- JSON data
- ASN data
- CASE data