Building a MARC-to-OLAC Crosswalk: Repurposing Library Catalog Data for the Language Resources Community

Christopher Hirt1,2, Gary Simons1,2, and Joan Panne1
1SIL International, Dallas, TX 2Graduate Institute of Applied Linguistics, Dallas, TX

Searching for language resources

The Open Language Archives Community (OLAC) is an international partnership of institutions (numbering 35 and growing) who curate collections of language resources (e.g. recordings, texts, dictionaries, grammars, language learning materials) concerning the majority of the world’s known 7,500+ languages. OLAC seeks to support precise search for language resources by aggregating the catalogs of the participating archives. Catalogs are expressed using a community-specific refinement of qualified Dublin Core [1] and harvested with a community-specific refinement of the OAI Protocol for Metadata Harvesting [2].

Many institutions with sizable collections of language resources already have catalogs in MARC format. In the effort to expand its coverage, OLAC has recognized the need to build a crosswalk system that maps language-resource records in an existing MARC catalog into the qualified Dublin Core format used by OLAC.

Language identification and MARC

Standard practice in MARC cataloging has been to identify the main content language with a three-letter code from ISO 639-2/b [3] in 008/35-37 and additional content languages in 041. When a language is the subject of a work, it is identified by an LCSH in 650$a. For instance, the MARC record for a Cheyenne-English dictionary might include:

008/35-37 eng
041 #0 SaengSachy
650 #0 Cheyenne language\$vDictionaries\$xEnglish

The ISO 639-2/b codes can identify only 360 individual languages. LCSH has much greater coverage with headings for about 3,500 specific languages. But still this covers under half of the known languages. A new best practice (which OLAC recommends) is to use ISO 639-3 (adopted 2007) which has over 5,000 three-letter codes covering all known languages, past and present [4]. For instance, the following encodes a work in the Bagirmi language of Chad:

041 #07 Saam\$b2iso639-3

Given the fact that cataloging standards have not covered all known languages until very recently, collections with a focus on minority languages have developed local practices to fill the gap. For instance, the Graduate Institute of Applied Linguistics (GIAL) uses a locally defined SI (language) subfield of field 590 (local note) to code a language which the resource is about, e.g.

590 #8 $b8mi$2iso639-3

The National Anthropological Archives (NAA) have designated field 695 (local subject access) to specify the name of the ethnolinguistic community, e.g.

695 #5 $aAlabamu

Design of the crosswalk

The crosswalk follows a basic three-stage process:

1. MARC XML Collection
2. Stage 1 filter: Select candidate language resources
3. Stage 2 filter: Reject non language resources
4. Stage 3 conversion: Map to OLAC metadata format

The National Anthropological Archives (NAA) have designated field 695 (local subject access) to specify the name of the ethnolinguistic community, e.g. 695 #5 $aAlabamu

Implementation of the crosswalk

Our crosswalk implementation was created with several goals in mind. Open source: Our solution is open source and leverages open source technologies like Python and XSLT, enabling OLAC community members to freely download and run the crosswalk on their local MARC catalogs. Scalability: Our solution scales to large MARC XML input files that a typical XSLT processor could not handle due to memory constraints. Local customization: Our solution provides for a local XSLT file to override any delivered crosswalk mapping and for the delivered filter to be customized with local filter rules. A Python script drives the entire crosswalk. It begins with a SAX process that splits the MARC XML input into manageable chunks of (by default) 1000 records each. The filter stages are then implemented as XSLT scripts that are applied in succession to each chunk. These scripts are generated at run time by an XSLT script that compiles the XML file describing the filter tests into XSLT. The Python script uses the Java Saxon XSLT 2.0 processor to apply XSL transformations.

The final mapping stage applies an XSLT script which defines an XSL template for each MARC field that is relevant to OLAC. E.g., the template for the 650 subject field uses a mapping from LCSH to ISO 639-3 compiled by the project [5].

For instance, the following test is used in the GIAL filter to select a record if it contains a local note with a language code:

\[<\text{field}\text{name}="\text{data-field}\text{entry}\text{exists}\text{tag}="\text{590}\text{codes}"/>\]

The filters for a given collection are configured via an XML vocabulary for expressing tests on the MARC leader, control fields, and data fields, e.g. the following is the RELAX NG definition for a data field test:

\[
\text{element data-field | }
\# x is a wildcard; only xxx, dxx, ddx are allowed attribute tag | (xsd:string ( pattern = "[0-9x](0-9x)?-9x") ), # if @code is empty, all subfields are searched. If @code is not # empty, it is a list of the subfields to search in attribute code | xsd:string ( pattern = "[a-z0-9-]*" ) },
\text{attribute test | } ("exists" | "contains" | "starts-with" ), (element text | text ) | }
\]

For instance, the following test is used in the GIAL filter to select a record if it contains a local note with a language code:

\[<\text{field}\text{name}="\text{data-field}\text{entry}\text{exists}\text{tag}="\text{590}\text{codes}"/>\]

The crosswalk follows a basic three-stage process:

1. MARC XML Collection
2. Stage 1 filter: Select candidate language resources
3. Stage 2 filter: Reject non language resources
4. Stage 3 conversion: Map to OLAC metadata format

Acknowledgements

The work reported in this poster has been supported by a grant from the National Science Foundation (BCS-0723864, “OLAC: Accessing the World’s Language Resources”) awarded to the Graduate Institute of Applied Linguistics, Dallas, TX.

Results

The complete crosswalk package with the configuration for the GIAL and NAA data sets is available for download [6]. The crosswalk has been successfully run over a collection of records supplied by the NAA and over the complete GIAL catalog; the resulting repositories will soon be operational at OLAC [7] [8]. The following table summarizes results in terms of the original collection size, the size after filtering, and the number with a language code of interest after mapping (e.g. more than just “eng” as content language).

<table>
<thead>
<tr>
<th>Records in MARC</th>
<th>Records in OAI repository</th>
<th>Records with ISO 639-1 code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAA</td>
<td>5,654</td>
<td>4,226</td>
</tr>
<tr>
<td>GIAL</td>
<td>32,654</td>
<td>8,176</td>
</tr>
</tbody>
</table>

We have begun work on two fronts to improve the precision in the final column: developing a stochastic classifier trained on Ethnologue data to extract language identification from titles and descriptions, and incorporating a final evaluation to reject records for which no language is identified.

References

7. http://www.language-archives.org/archive/naa.nmnh.si.edu