Experiments in Operationalizing Metadata Quality Interfaces: A Case Study at the University of North Texas Libraries

Mark Edward Phillips
University of North Texas Libraries,
United States
mark.phillips@unt.edu

Hannah Tarver
University of North Texas Libraries,
United States
hannah.tarver@unt.edu

Abstract

This case study presents work underway at the University of North Texas (UNT) Libraries to design and implement interfaces and tools for analyzing metadata quality in their local metadata editing environment. It discusses the rationale for including these kinds of tools in locally-developed systems and discusses several interfaces currently being used at UNT to improve the quality of metadata managed within the Digital Collections.

Keywords: metadata quality; user interfaces; metadata quality interfaces; Web interfaces

1. Introduction

Digital collections in cultural heritage institutions including libraries, archives, museums and galleries have grown steadily over the past decade. As technologies for the digitization of analog collections and the accumulation of born-digital materials has become more accessible to institutions of all sizes, these same institutions have made great efforts toward making digital resources available via the web. With this increase, many have begun to focus on the quality of the metadata that describe these resources. Analysis of metadata for digital resources has been conducted on large aggregations in the Digital Public Library of America (DPLA) in the United States, and Europeana in the European Union (Harper, 2016; Tarver, Phillips, Zavalina & Kizhakkethil, 2015). This work has led to discussion on how to communicate needed metadata improvements to local repositories (Dangerfield, 2015). While this remains an unsolved problem, there is another gap that is not as often discussed: mainly, how are local repositories experimenting with tools and interfaces to understand the quality of the metadata in their own systems, and how are these same tools and interfaces used in practice?

Of course, one concern for anyone working with metadata is determining the quality of the data, such as the existence of typos, missing or mislabeled information, or improper formatting. These errors can be introduced in a number of ways, including data input errors, importation of data that has different formatting, and values based on outdated rules. The larger the collection, the more difficult it can be to check for errors manually. The Digital Library Federation Assessment Interest Group Metadata Working Group has started to collect documentation and tools as a first step toward providing guidance for local repositories (DLF AIG MWG Metadata Assessment Toolkit, n.d.), though there is a wide range of needs.

It is almost expected today that there are tools and interfaces built into digital library platforms to help metadata editors assess and understand the quality of the metadata that they are creating. From our research we have not discovered that this is the case. One of the challenges that we see in this area of research is that most of the tools and interfaces that have been developed by institutions may be focused solely on their local situations, workflows, and data models, and therefore have not been broadly shared with others. This is unfortunate because there is much that we can learn from others related to what they are trying to accomplish, how they are working toward these goals, and the interfaces and systems that they are putting in place. This case study does not attempt to define or characterize specific quality measures in the Digital Collections, but

it discusses the work underway at the UNT Libraries, focused on building tools and interfaces for reviewing and generally improving metadata.

1.1 Background

The University of North Texas (UNT) Libraries' Digital Collections comprise more than 2.2 million items, housed in a single administrative system and publicly accessible via three interfaces. The Portal to Texas History (https://texashistory.unt.edu/) contains materials owned by nearly 400 different partner institutions across the state of Texas; the UNT Digital Library (https://digital.library.unt.edu/) contains items owned, created, or licensed by UNT, including current scholarly works; the Gateway to Oklahoma History (https://gateway.okhistory.org/) contains items owned by the Oklahoma Historical Society. The level of collaboration across the Digital Collections means that a number of metadata editors work within the system to create or change metadata. Since 2009, more than 700 unique editors have edited records in the metadata editing system, including trained staff members, catalogers, library science students, and volunteers.

The current digital library system was developed in-house using open-source components. It was completed in 2009 and has undergone a number of iterative changes to both the public and administrative interfaces. Metadata in the Digital Collections is based on Dublin Core with the addition of local fields and qualifiers for a possible twenty-one fields used for all items in the system, including eight that are required for every record. There are extensive guidelines in place outlining the technical and semantic expectations for metadata in each field.

This paper seeks to discuss some of the experiments in tools and interfaces being developed at the University of North Texas Libraries that help metadata creators identify and improve deficiencies in their collections of metadata.

2. Analysis Tools

As the UNT Libraries' Digital Collections have grown, we have become increasingly aware that we need tools to allow us to understand the quality of the metadata that is being created in these collections and to analyze or compare larger and larger sets of data. The first tool, called the "Metadata Analysis Tool" was built in 2005; sadly, because its features were only used internally, there were only a few external presentations and no published discussion of how we used the tool in our systems. The Metadata Analysis Tool was forgotten in our library as we migrated our digital collections from system to system.

For a number of years, we have been doing some basic analysis on record values by harvesting the records and using Python scripts to look at field values (Phillips, 2013). Although this is useful, there are some downsides: it is not always easy to check values across multiple collections or the whole system; it can be difficult to check everything systematically without a particular concern in mind; and importantly, this method is not particularly accessible to the many editors working on metadata in our system. We wanted to move toward tools that could be used by metadata editors to check their own work, or to identify problems throughout the system and start correcting them.

Some institutions have had success with tools like OpenRefine for cleaning up metadata for their digital collections. We, too, have used OpenRefine for projects to improve metadata before it is added to our primary digital library platform (Phillips, Tarver, & Frakes, 2014). Like many, we found that OpenRefine is a wonderful tool for working with spreadsheets and other types of data, but there are a few challenges. First of all our data generally isn't rectangular and doesn't easily fit into a spreadsheet representation. We have some records with one creator and others with dozens of creators. There are ways to work with these kinds of data but it can get complicated. A bigger challenge we have in our local environment is that while many systems can generate a spreadsheet of their data for exporting, very few -- including our metadata management system -- have a way of importing those changes back into the system in a spreadsheet format. This means that while

you can pull data from the system and clean it up in OpenRefine, there is no way to get that nice clean data back into the system. A way that we found that we could use OpenRefine was to identify records to change and then go back into the system and edit records there; however it is a tedious and time-consuming process. In order to overcome this set of challenges we decided that we needed to build analysis tools directly into the metadata-editing interface used for the Digital Collections. That way our metadata editors could identify a problem and immediately fix it in an interface they understand and use every day.

2.1 Facet and Count Interfaces

During summer 2017, our software development team implemented the first of our suite of integrated analysis tools: Count and Facet. For each of the tools -- including Cluster, which was added later and is described further in the next section -- an editor must choose a specific field but has the option, when applicable, to limit to any qualifier, to a specific qualifier, or to values that have no qualifier. Editors also have the ability to filter the record analysis based on other criteria, such as collection or institution, material type, public visibility, or records that the editor has modified. These criteria and results of the analysis assist in identifying obvious problems, such as records without specific types of required values or existing values that do not have qualifiers.

Count sifts records based on the number of entries in a field so that editors could see, for example, that there are 65,772 records containing 0 subject entries, 23,026 records containing 1 subject entry, 12 records containing 87 subject entries, etc. Figure 1 shows counts for physical description entries, for which records should never have multiple entries and, ideally, ought to have a single entry, though it is not required. Currently, 76 records have two description entries labeled "physical description" and nearly 156,000 records have no physical description. Based on these counts, an editor should review the 76 records with multiple physical descriptions to fix qualifiers (if values are mislabeled) and to move or collapse information as needed, to eliminate multiple entries. As a longer-term project, we would also want editors to start adding physical descriptions to the 156,000 records without values and to review those records, a many of them likely have other errors or omissions.

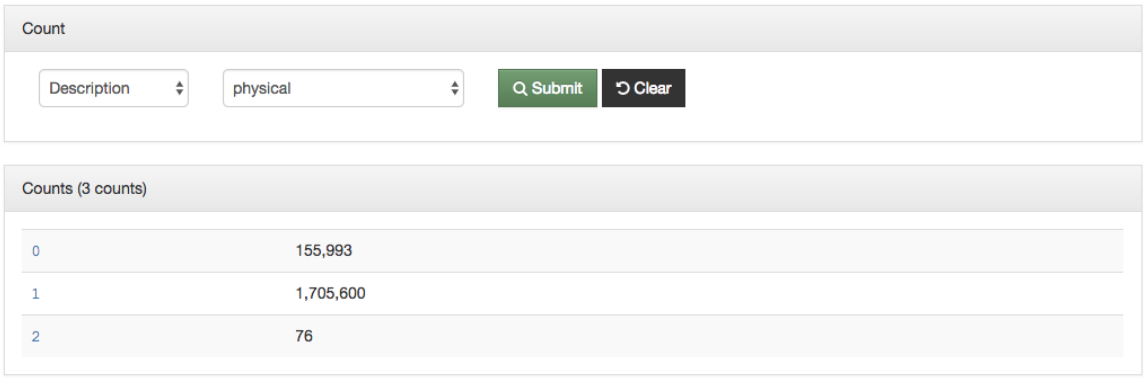

Figure 1. Example values in the "Count" tool for physical description entries.

Facet lists all of the unique values for a particular field and the frequency, i.e., the number of records in which each term appears (see Figure 2). This tool is most useful for finding typos and small inconsistencies across values in a field, such as "machine gun" versus "machine guns." It also lets an editor see the most commonly used terms in a collection or across the system.

Values for qualifiers, some sub-fields, and five required fields (language, resource type, format, collection, and institution) are managed in local controlled vocabularies and are connected directly to the edit interface as drop-down menus to prevent non-valid terms. Though we also encourage the use of other sources of controlled vocabularies -- e.g., VIAF Virtual International Name File) or LCNAF (Library of Congress Name Authority Files) for names; LCSH (Library of Congress Subject Headings), Legislative Indexing Vocabulary (LIV), Chenhall's Nomenclature for Museum

Cataloging, etc. for subjects -- we don't currently have a way to validate against external controlled vocabularies and the terms are not always consistent. Facet can be useful in these instances to compare controlled and uncontrolled terms alphabetically to see where there might be overlap and to determine when it might be appropriate to change values.

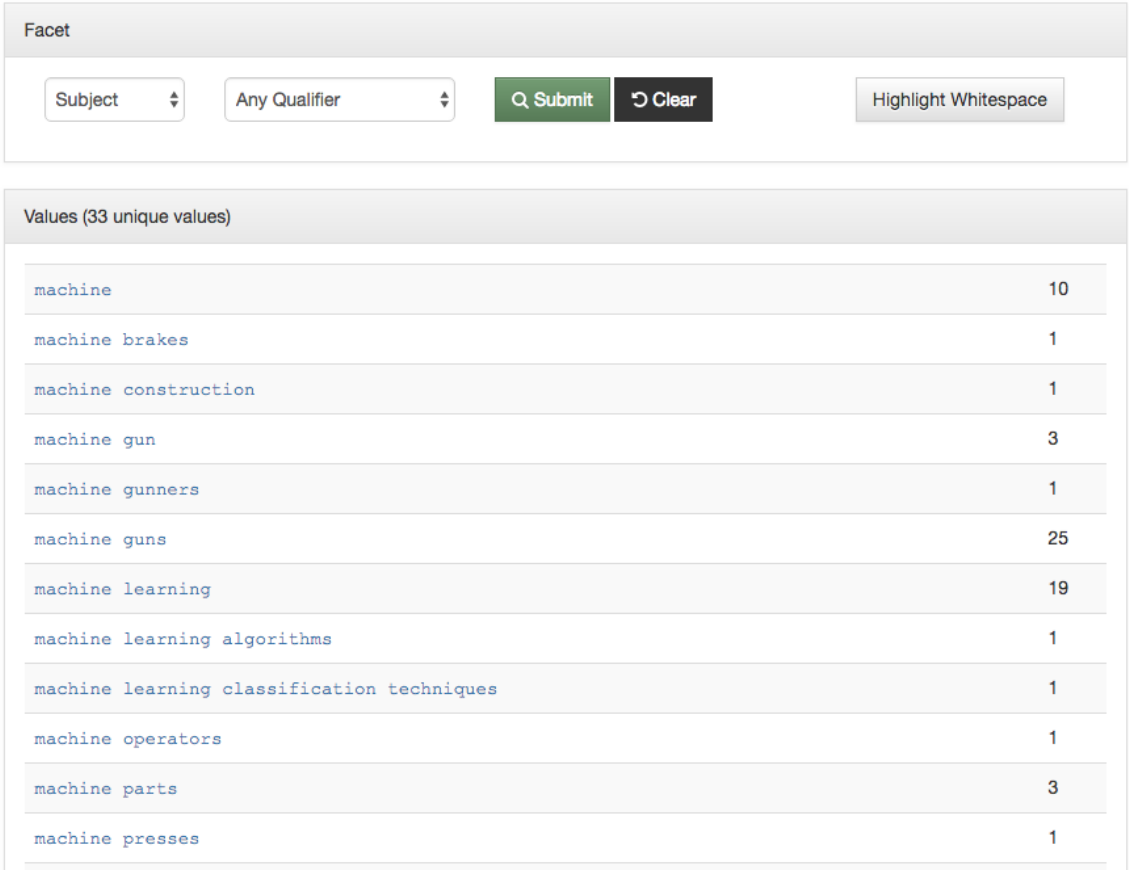

Figure 2. Example values in the "Facet" tool for all subject entries.

2.2 Cluster Interface

The third tool that we introduced is more complicated and uses algorithms to normalize values and look for matches, the same way that OpenRefine can cluster possible matches in spreadsheet data. In this tool, a user has the same options, with the addition of a drop-down menu to choose an algorithm. Cluster is intended to automatically group together values that are most likely to be differently-formatted versions of the same term. Adjusting the algorithm used to normalize the values can result in different clusters, depending on the types of variations that an editor might want or expect to find (see Table 1).

Most of the time the default (fingerprint) algorithm is sufficient. Fingerprint normalizes the values by changing all characters to lowercase, simplifying non-ASCII characters, replacing punctuation with spaces, removing spaces at the start or end of the term, collapsing duplicate spaces within the term, alphabetizing the tokens, and deleting any duplicate tokens. Our implementation of this fingerprint algorithm is the same that is used by OpenRefine (Clustering in depth, n.d.). We have found it to be a good baseline algorithm for metadata editors (see Figure 3). Each cluster displays the number of members (unique values), the number of records containing the clustered values, the key (normalized text string), and the member values (existing values with the number of records in which they appear). Clusters can be sorted alphabetically based on the cluster key, by number of total records or members, by total length, and by the amount in variation of length among cluster members.

Members	Records	Key	Member Values
14	148	1906 1975 dmitrievich dmitrii shostakovich	Shostakovich,Dmitriĭ Dmitrievich, 1906-1975. (2) Shostakovich, Dmitriĭ Dmitrievich, 1906-1975. (5) Shostakovich, Dmitriĭ Dmitrievich, 1906-1975 (44) Shostakovich, Dmitriĭ Dmitrievich 1906-1975 (9) Shostakovich, Dmitriĭ Dmitrievich, 1906-1975 (3) Shostakovich, Dmitrii; Dmitrievich, 1906-1975 (2) Shostakovich, Dmitriĭ Dmitrievich, 1906-1975 (1) Shostakovich, Dmitriĭ Dmitrievich, 1906-1975 (1) Shostakovich, Dmitrii˘Dmitrievich, 1906-1975. (23) Shostakovich, Dmitrii˘Dmitrievich, 1906-1975 (13) Shostakovich, Dmitrii Dmitrievich, 1906-1975 (37) Shostakovich, Dmitrii Dmitrievich, 1906 1975 (1) Shostakovich Dmitrii Dmitrievich, 1906-1975 (1) Dmitriĭ Dmitrievich Shostakovich, 1906-1975 (6)
8	44	1876 1946 de falla manuel	de Falla, Manuel, 1876-1946 (4) de Falla, Manuel 1876-1946 (1) Falla, Manuel de, 1876-1946. (1) Falla, Manuel de, 1876-1946 (26) Falla, Manuel de 1876-1946 (3) Falla, Manuel De, 1876-1946 (3) Falla Manuel de 1876-1946 (1) De Falla, Manuel, 1876-1946 (5)

Figure 3. Example clusters for contributor names using the fingerprint algorithm.

Once the basic framework was in place for applying an algorithm to a string to perform normalization and hashing into buckets, we started to experiment with variations on algorithms that would be useful in specific circumstances, outlined in Table 1.

TABLE 1: Clustering algorithms with example values.

Algorithm	What it Does	Example Input	Example Output
Fingerprint	Normalizes capitalization & punctuation, deletes duplicate words	Wereszczak, Andrew A.	a andrew wereszczak
Fingerprint - No spaces	Same as fingerprint & removes punctuation without changing spacing	F.B.I.	fbi
Fingerprint - No dates	Same as fingerprint & ignores dates	Schmidt, Brian A., 1980-	a brian schmidt
Caseless	Makes all values lowercase	Austin, Stephen F.	austin, stephen f.
ASCII	Converts letters with diacritics to their plain ASCII representation	Castillo, José	Castillo, Jose
Normalize Whitespace	Replaces repeated whitespace with a single whitespace character	David S. Castle Co.	David S. Castle Co.

Alternative algorithms or customized versions of algorithms can also be added as needed, to isolate or eliminate particular kinds of values within clusters. We have noticed that some algorithms work better for certain fields; for example, the Fingerprint - No Dates algorithm works most effectively on the creator and contributor fields that contain many values that only differ by the inclusion of dates, such as authorized forms of names in the Library of Congress authority file versus unauthorized forms. This also works for numeric symbol codes. For example, the cluster for Shostakovich (Figure 3) gains a fifteenth member using the "no dates" version -- Shostakovich, Dmitriĭ Dmitrievich, 1906-1975 – and composer Gabriel Faure (not visible in the example) gains the variation "Faure, Gabriel" alongside "Fauré, Gabriel, 1845-1924." and Faure, Gabriel, 1845-1924, among others.

2.3 Sampling

When working with the normalization algorithms described above, the goal was to identify groups or clusters that contain two or more values. If a cluster only had a single value, it was

ignored and not displayed in the interface. We found that this was not always desirable, but when we wanted to analyze values in groups that contained a large number of members, we ran into problems with the interface and how to display these sets.

The Cluster tool has a useful framework to group values by specific features, such as length or alphanumeric patterns. For these cases, because every value is included in the results, some clusters get extremely large and would be prohibitive to display. Instead, clusters with over 100 values are displayed by sampling according to chosen criteria -- random values, first or last values alphabetically, most or least frequent -- so that each cluster is a reasonable size (see Figure 4).

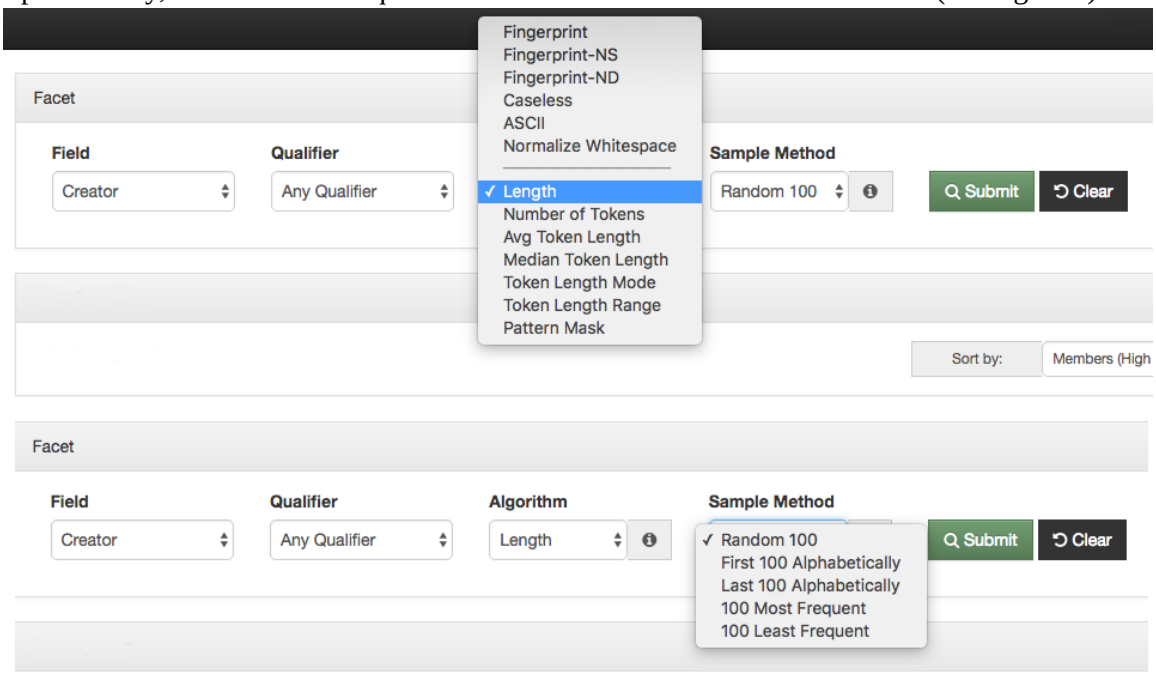

Figure 4. Menu options in the "Cluster" tool for algorithms that use sampling.

These sampled algorithms assist in identifying values that are outliers, such as subject values that have only one character or that have more than 1000 characters. It also makes it easier to sort by patterns or lengths, which can be helpful for certain fields.

An algorithm that uses this sampling is our Pattern Mask algorithm that takes the input string and converts any digits to the character '0' and any letters to the character 'a', while leaving any punctuation as it is (see Figure 5). An example of this would convert the EDTF (Extended Date/Time Format) date of '194u' into the string '000a'. We aren't the first to find this especially useful for analyzing dates because it allows date values with similar patterns to group together, such as '193u' and '194u', which both convert to '000a' (Van Hooland, 2009).

1	4	000#~	201#~ (4)
1	1	000+0000	197+0427 (1)
1	2	000-	201~ (2)
2	2	000-00 aaa	118-63 BCE (1) 220-83 BCE (1)
4	4	000-00-00	193-08-30 (1) 206-01-01 (1) 200-12-16 (1) 193-12-16 (1)
8	8	000-000 aa	161-169 CE (1) 247-249 CE (1) 238-244 CE (1) 609-610 CE (1) 286-310 CE (1) 138-161 CE (1) 211-260 CE (1) 193-211 CE (1)
1	1	000-000 aaa	295-280 BCE (1)
1	1	000-0000 aa	976-1028 CE (1)
589	307,783	0000	2022 (3) 1818 (35) 1465 (4) 1430 (1) 1982 (2,589)

Figure 5. Clusters for date entries using the pattern mask algorithm and random sampling.

The screenshot provided of the Pattern Mask algorithm demonstrates the need for the sampling mechanism, since '0000' is very common, containing 589 different values that represent 307,783 records. These 589 different values would take up considerable space on the page and generally are less likely to require investigation than clusters that contain only a few different values; instead only 100 random values are displayed.

3. Analysis Tools in Relation to Quality

Although the tools and interfaces cannot identify problems automatically, each of them is meant to assist editors in finding values that are incorrect or that may need to be verified for accuracy. In some cases, clustered values could all be correct. For example, strings that contain duplicate word tokens will cluster together using fingerprint -- e.g., *United States - Texas - Denton County* and *United States - Texas - Denton County - Denton* -- even though these may be separate, unique terms that are both valid.

However, these tools are still useful for improving the quality of records because they provide editors with relatively easy ways to analyze values in a particular collection to find typos, incorrect formatting, missing values, and other inconsistencies that would be time-consuming and difficult to identify through spot-checking or other methods of proofreading.

Since the Facet, Count, and Cluster interfaces are directly integrated into the editing system, results in the analysis tools connect to item records. Clicking on a count/value in any of the tools opens a new tab with the standard Dashboard displaying search results for only the records that have that count/value and that meet any other selected criteria. Although there is no way to make "batch" edits or "find-and-replace" across our system, this integration makes it relatively easy to identify specific, known problems and provide a link to an editor who can change the records or review them, as needed, to improve the consistency and overall quality.

3.1 Metrics

We have begun to use the number of clusters as a rough but relatively accurate metric for one aspect of quality; i.e., if the number of clusters in a collection is reduced, that would represent an increase in consistency and lower entropy, and therefore an increase in quality, at least for a particular field. For one specific project, a person was assigned to start fixing formatting problems

with names in a technical report collection and to keep track of the number of clusters to show progress. What we actually found in that instance is that the number of clusters often increases before it decreases. For example, if records in the collection have J. D. Smith represented three different ways -- Smith, J.D.; Smith JD; and Smith, J D -- the first two versions would group together using the basic fingerprint algorithm, but the third would not be in the cluster. After some of those names are corrected to the version we prefer in our system (Smith, J. D.), the same algorithm would now cluster together the third original version and the new, correct version, doubling the number of clusters for that name. Since some of the individual records contain multiple authors, fixing all of the names in one record could add four or five new clusters in some cases, during the process of clearing one.

Essentially, while the number of clusters can still give a general sense of consistency (particularly for certain fields or within specific collections), using the number of clusters as a relatively exact metric, or expecting that it represents the actual number of corrections appears to be much fuzzier than we had first expected. Despite this we have found this metric useful over longer periods of time to show the improvement in consistency within a given field across a collection of metadata records. Similarly, the number that displays in each tool provides some reflection of work; for example, if the number of unique values decreases (as values are made more consistent) or increases (as missing values are added) when using Facet to analyze fields.

4. Discussion

As we mentioned above the benefit we have found with the approach of building these metadata quality interfaces into our digital library system is that it allows our metadata editors direct access to records that have an identified issue. Another benefit is that as metadata records change, the interfaces are quickly updated.

A limitation that we've found especially with the clustering interface is the amount of time it takes to generate the clusters across all two million records in our system. Because each clustering algorithm has to operate a string transformation on every unique value for a field, there is the possibility of over a million iterations for a large field like subject. With the current implementation some clusters take almost twenty seconds to generate. As we add more records with more values to the system this time will likely increase, causing users to wait for longer periods of time. To overcome some of the wait time, we currently cache clusters for ten minutes before they are regenerated. Although that means that the values do not update in real time as changes are made, this seems to be a reasonable tradeoff for users as they often work on a number of different clusters before needing to have the clusters regenerated.

The tools and interfaces discussed in this paper are useful in helping to identify problems, but they are a first step, both for future development in our system and for additional quality-related research. There may be ways to implement new tools or changes to existing interfaces depending on needs expressed by metadata editors or ways that certain fields affect the usability of public interfaces and usefulness to external users. One ongoing question is how to prioritize corrections. Looking at values across more than two million records results in an extremely large number of known problems and outliers that could be possible errors. There are many ways to organize that information. For example, is there more benefit in fixing clusters that affect a larger number of records, versus clusters that have a larger number of members (less consistency)?

This case study was designed to demonstrate some of the new tools and interfaces that have been developed for the UNT Libraries' Digital Collections. While there is a plan to eventually release the software for these interfaces, it is unlikely it would be adopted by other institutions because of the specific design decisions that were made to meet our local needs. What we do hope is that others who have thought about building metadata quality tools and interfaces will see this case study and will be interested in developing similar tools or interfaces in their local environments. Interfaces for metadata management are often only available to locally-authenticated users, so it is usually

impossible to see locally-developed tools for working with metadata. By sharing this case study we hope that others are encouraged to share their work, both as code, but more importantly with discussion about how and why the tools and interfaces were developed in the first place.

5. References

Clustering in depth (n.d.). In GitHub. Retrieved May 7, 2018 from https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth

Dangerfield, Marie-Claire et. al (2015). Report and recommendations from the Task Force on Metadata Quality. Retrieved from http://pro.europeana.eu/files/Europeana_Professional/Publications/Metadata%20Quality%20Report.pdf

DLF AIG MWG Metadata Assessment Toolkit (n.d.). In GitHub. Retrieved August 13, 2018 from http://dlfmetadataassessment.github.io/

Harper, Corey. A. (2016, June). Metadata analytics, visualization, and optimization: Experiments in statistical analysis of the Digital Public Library of America (DPLA). Code4Lib (33). Retrieved from http://journal.code4lib.org/articles/11752

Phillips, Mark Edward (2013, January). Metadata analysis at the command-line. Code4Lib (19). Retrieved from https://digital.library.unt.edu/ark:/67531/metadc157309/

Phillips, Mark Edward, Hannah Tarver & Stacey Frakes (2014, January). Implementing a collaborative workflow for metadata analysis, quality improvement, and mapping. Code4Lib (23). Retrieved from https://digital.library.unt.edu/ark:/67531/metadc284577/

Tarver, Hannah, Mark Phillips, Oksana Zavalina, & Priya Kizhakkethil (2015). An Exploratory Analysis of Subject Metadata in the Digital Public Library of America. Proceedings of the International Conference on Dublin Core and Metadata Applications, 2015, 30-40

Van Hooland, Seth (2009). Metadata quality in the cultural heritage sector: Stakes, problems and solutions (Thesis). Retrieved from http://homepages.ulb.ac.be/~svhoolan/these.pdf