ARTICLES

D-Lib Magazine
April 2001

Volume 7 Number 4

ISSN 1082-9873

Automated Name Authority Control and Enhanced Searching in the Levy Collection

Tim DiLauro
G. Sayeed Choudhury
Mark Patton
James W. Warner
Digital Knowledge Center
Milton S. Eisenhower Library
Johns Hopkins University
timmo@jhu.edu, *sayeed@jhu.edu*
jwarner@jhu.edu, *mpatton@jhu.edu*
http://dkc.mse.jhu.edu

Elizabeth W. Brown
Cataloging Department
Milton S. Eisenhower Library
Johns Hopkins University
ebrown@jhu.edu

Introduction

This paper is the second in a series in *D-Lib Magazine* and describes a workflow management system being developed by the Digital Knowledge Center (DKC)[1] at the Milton S. Eisenhower Library (MSEL)[2] of The Johns Hopkins University.[3] Based on experience from digitizing the Lester S. Levy Collection of Sheet Music,[4] it was apparent that large-scale digitization efforts require a significant amount of human labor that is both time-consuming and costly. Consequently, this workflow management system aims to reduce the amount of human labor and time for large-scale digitization projects. The mission of this second phase of the project ("Levy II") can be summarized as follows:

- Reduce costs for large collection ingestion by creating a suite of open-source processes, tools, and interfaces for workflow management
- Increase access capabilities by providing a suite of research tools
- Demonstrate utility of tools and processes with a subset of the online Levy Collection

The cornerstones of the workflow management system include optical music recognition (OMR) software and an automated name authority control system (ANAC). The OMR software generates a logical representation of the score for sound generation, music searching,

and musicological research. The ANAC disambiguates names, associating each name with an individual (e.g., the composer Septimus Winner also published under the pseudonyms Alice Hawthorne and Apsley Street, among others). Complementing the workflow tools, a suite of research tools focuses upon enhanced searching capabilities through the development and application of a fast, disk-based search engine for lyrics and music and the incorporation of an XML structure for metadata.

The first paper (Choudhury et al. 2001) described the OMR software and musical components of Levy II. This paper focuses on the metadata and intellectual access components that include automated name authority control and the aforementioned search engine.

Overview

During the first phase of the Levy Project, which focused on the digitization and mounting of sheet music images on the Web, an online index was created at the sheet music item level. The record for each piece of music included (when available): the unformatted transcription of title, statement of responsibility, first line of lyrics, first line of chorus, dedication, performer, artist/engraver, publication information, plate number, and box and item number. In addition, subject headings were applied using a controlled vocabulary derived from the Library of Congress' Thesaurus of Graphic Materials. As a result of these efforts, users can search the online Collection by keyword or phrase.

One of the ultimate goals of the overall Levy Project is to utilize the bibliographic "raw material" described above as the basis for powerful searching, retrieval, and navigation of the multimedia elements of the collection, including text, images and sound. The existing index records have been converted from text files to more structured metadata using XML tagging. Between now and the end of the project, name information from the index records will be extracted into specific indices such as composer, lyricist, arranger, performer, artist, engraver, lithographer, dedicatee and, possibly, publisher. At the end of Levy II, cross-references will direct users to index records that contain various forms of names.

Consistent with the philosophy of Levy II and the workflow management system, we have developed automated tools that will reduce the amount of human labor (and therefore costs) necessary to accomplish the metadata and intellectual access goals described above. On the collection ingestion side is the automated name authority control system (ANAC) that is described below. On the intellectual access side, we have developed a search engine that augments metadata searching with unique search capabilities for lyrics and music. By combining metadata-based searching with full-text and fuzzy searching via the search engine, the full range of rich intellectual information will be made more easily accessible from the online Levy Collection.

Automated Name Authority Control

In a collection as large as the Levy Collection, where an individual author may have multiple works, it is highly likely that some authors have their names listed in more than one form. This is certainly the case for the Levy Collection. For example, the same individual wrote the lyrics for both Levy Collection titles *My Idea of Something to Go Home to* and *Pretty as a Picture*, but the lyricists are listed as "Robert Smith" and "Robert B. Smith", respectively. The automated name authority control system (ANAC) enhances access to such a collection by identifying other works listed with variant forms of an individual's name based on the canonical, or authoritative, form of that name.

The primary goals for our automated name authority control system are to:

- Improve access to the Levy Collection by introducing authorized name searching.
- Produce a tool that will reduce the cost of introducing authority control to large digital collections. In the case of the Levy collection, which contains more than 29,000 pieces, doing manual authority work for the entire collection would be time consuming and prohibitively expensive.
- Improve interoperability of Levy metadata with other collections by using standard data sources (LC name authority file[5]) and formats (XML). (As part of Levy II, but outside the scope of the ANAC, we plan to create mappings to the Z39.50 Bib-1 Attribute Set, which will further enhance interoperability.)
- Produce a name match confidence value so collection managers can establish thresholds that trigger manual intervention. The confidence measure enables the ANAC to operate as a tool in the workflow framework we are developing. The thresholds allow collection managers to select the confidence above which they trust the system to make reliable decisions, thus reducing manual intervention (and therefore cost).

The foundation of the automated name authority control system is the Library of Congress (LC) name authority file, the canonical source of authority data for most U.S. collections. This choice will allow our collection to interoperate with other collections and, more generally, make the ANAC a tool that will work with many collections. To improve efficiency and take advantage of standard tools, the authority file is loaded into a relational database management system with appropriate field indices.

The Metadata

Index metadata records were created during the first phase of the Levy Project, while the sheet music pages were being imaged. These records were created by the Levy Project Coordinator, who transcribed information directly from the scores. The metadata format was simple, consisting of multiple lines of attribute/value pairs. In order to minimize cost, the statement of responsibility was transcribed, but no name entries were created. For example, the statement of responsibility from *A Nation Mourns Her Martyr'd Son* appeared as follows:

```
Composer, lyricist, arranger: Words by Alice Hawthorne. Music by
Sep. Winner.
```

This scheme has since been updated to an XML format to permit fielded searching. In the example below, "LSM" is the XML namespace and is an acronym for "Levy Sheet Music". After this enhancement, that same metadata had the following form:

```
<LSM:ComposerLyricistArranger> Words by Alice Hawthorne. Music by
Sep. Winner.
</LSM:ComposerLyricistArranger>
```

As you can see, a statement of responsibility may consist of more than just the names of the "authors". In some cases, the honorific is quite effusive (e.g., one entry contains "Composed by a Distinguished American Song Writer, Arthur Sullivan"). Searching on records with such widely varying formats would produce inconsistent matching and make it more difficult to group results.

In order to create name entries retrospectively, we needed to extract words that are part of a name while determining the role or roles (e.g., lyricist or composer) associated with that name.

With a dictionary and a list of first and last names, we were able to automatically extract the names and their respective contributions from each entry.

To accomplish this, we treat every sentence as a vector of words. Stepping through the vectors, we mark each word as either an English word or part of a name. On the first pass, we mark only those words which are unambiguous. A word is considered unambiguous if it occurs only in the list of names or only in the English dictionary. On the second pass, ambiguous words are changed to either words or names depending on the situation. For example, if an ambiguous word occurs between two parts of a name, we mark it as part of that name.

This process achieved 97.2% precision and 98.7% recall. Recall was favored (perhaps at the cost of precision) since the next step (disambiguation) should be able to catch some precision errors. After we completed this processing, the metadata looked like this:

```
<LSM:ComposerLyricistArranger>Words by Alice Hawthorne. Music by Sep. Winner.
</LSM:ComposerLyricistArranger>
<LSM:Name>
    <lyricist>Alice Hawthorne</lyricist>
    <composer>Sep. Winner</composer>
</LSM:Name>
```

Name Disambiguation

This format is a substantial improvement over its predecessors, since it is now quite straightforward to implement powerful searching over the collection using available tools. (Note also that we have retained the original data transcribed from the statement of responsibility.) However, we still need a mechanism to locate all works created by a given individual. To do this, we need an unambiguous reference to the individual. This is the authorized name. The process for deriving this reference is "name disambiguation".

Like the adaptive optical music recognition (AOMR) system described in the previous paper, the name disambiguation system is a learning system. It must be trained with a representative subset of the collection on which it will operate. To facilitate this learning, to aid in developing a confidence measure, and to allow an objective evaluation of our success, we randomly distributed names from the collection into four sets: a seed group (small enough to process manually), a test group (larger than the seed group, but small enough to review for accuracy), and two large training sets. The entries in each group were then automatically clustered based on name similarity (Warner and Brown 2001).

Names from the seed group were matched manually to entries in the authority file to provide the initial settings for the training process. From this manual process, we determined the characteristics of names from our collection that have a matching authority record and contrasted them against the characteristics of those that do not. Based on this correlation, we selected the attributes to use as evidence for deducing confidence levels for name matches outside of the seed group. Once we complete this process, these confidence levels will be used to determine when human intervention is required.

When we begin processing the remaining records, the confidence measure will be produced using a Bayesian probability model based on the evidence derived from the seed data. The Bayesian approach is effective for a learning system since it enables the systematic update of expectations as new evidence becomes available. Based on the correlation information derived by processing the seed data, we plan to use the following evidence:

- **Information from the authority file note fields**. For a music collection like Levy, terms that indicate a musical affiliation may be useful clues. Other note information may be useful for other collections.

- **Publication date v. author birth/death dates**. It is extremely unlikely (but not out of the question -- the date may be incorrect in the authority file) that a person published before she/he was born. It may also be true that there are trends associated with posthumous publication as well.[6]

- **"Commonness" of a name**. The likelihood of a correct match is lower for very common names. Currently we use the number of LC records that match the name to determine this metric. Other measures of commonness could be substituted.

The first training set will be disambiguated using the probabilities from the seed group. Entries that generate a confidence level below an established threshold are flagged for manual processing. Entries with high confidence can be treated like seed data and used to modify the seed group probabilities (Yarowsky 1995). This process will be repeated until the probabilities stabilize. The second training set will then be disambiguated (perhaps repeatedly) using these new probabilities. Again, the high confidence entries will be used to adjust the probabilities, producing the final evidence/probability matrix.

Finally, the resulting system will be applied to the test set to measure the expected accuracy across the entire collection. Optionally, the two training sets can be reprocessed using the final set of probabilities.

Once the disambiguation process is completed, the authorized form of names that were matched to their LC name authority records will be added to the corresponding collection index record. When completed, the record might look like this:

```
<LSM:ComposerLyricistArranger>Words by Alice Hawthorne. Music by Sep. Winner.
</LSM:ComposerLyricistArranger>
<LSM:Name>
      <lyricist>Alice Hawthorne</lyricist>
      <composer>Sep. Winner</composer>
</LSM:Name>
<LSM:Authname>
      <lyricist>Winner, Septimus, 1827-1902</lyricist>
      <composer>Winner, Septimus, 1827-1902</composer>
</LSM:Authname>
```

In this scheme, the authorized form of the name is listed in the LSM:Authname tag, the entries of which correspond directly to the entries of the LSM:Name tag. Because Alice Hawthorne and Septimus Winner are actually the same person, the ANAC will locate the same authority record, which contains both of these names. 'Winner, Septimus, 1827-1902' is the authorized heading for this person and, thus, will appear in the index record as the authorized form of both composer and lyricist. The authority record contains 'Hawthorne, Alice, 1827-1902' as a "cross-reference." Our implementation of name searching in the Levy interface will eventually provide for searching of cross-references based on the authority records. Similar to searching in many online catalogs, a search against the Levy metadata for Alice Hawthorne will collect the example above but also all sheet music with either Alice Hawthorne or Septimus Winner on the piece.

A Versatile Text Searching Engine

The online Levy Collection is a diverse multimedia environment featuring a large collection of textual data. This data will include metadata such as name entries and authorized names, lyric text, and music notation derived with optical music recognition software. For end users to exploit this diversity, the Levy Collection needs a powerful and flexible searching capability. The search tools should reduce complexity for casual users, while at the same time providing enhanced functionality for sophisticated searchers. For collection managers, the tools must be flexible enough to handle any form of textual data in a meaningful way. For example, it probably does

not make sense to handle a textual representation of music the same way one would handle lyrics. Those distinctions, however, are resolved by front-end tools (e.g., taggers and tokenizers). Once the preprocessing is completed, the search engine operates the same way on both types of data.

Our search engine (still unnamed at this time) is emerging as just such a facility. The search engine and index builder proper have been completed and are written in C. The index-building front-end tools are implemented in C, Perl, and Tcl. Currently, indexing new collections or adding new search features usually requires additional programming of these front-end tools. We do plan, though, to build additional tools to aid in the creation of indices.

The core searching functionality is available as a C library, which allows bindings for scripting languages to be created easily. In fact, we are using these language bindings to create web interfaces for several of our projects. The system will be made available under an Open Source license.

Searching is implemented by first building one or more indices and then issuing search requests. We will first discuss the index generation process and then provide some examples of the search capabilities of this engine.

Indexing

The system treats a corpus of text as a sequence of tokens (words, punctuation, etc.). Indexing a corpus consists of assigning each unique token a number, creating a representation of the corpus using those numbers, and making several tables of information indexed by those numbers. The inverted file, for example, holds a list of positions for each token in the corpus. Once this "primary" index is created, the locations of a token in the corpus can be found efficiently by looking up the number representation of that token and stepping into the index. In addition, one or more secondary indices may contain entries that point to multiple primary index tokens. For a complete discussion of inverted file search engines, see (Whitten et al. 1999).

Secondary Indices

A secondary index maps a secondary token to one or more primary tokens. For example, suppose that the primary token consists of a word and part of speech[7] tuple such as *the/DT*. A secondary index could be built for both the word and the part of speech. Consequently, searching for *DT* in a "part of speech" secondary index might return the locations of all the primary tokens such as *the/DT* and *a/DT*. A more complicated secondary index on lemmas would create a mapping such as *tear/V* = {*tear/VB, tears/VBZ, tearing/VBG, tore/VBD, torn/VBN*}.

Supposing that the primary tokens are word and part of speech tuples, many useful secondary indices can be built. Examples include secondary indices of the tuple's lemma, syllabified form, phonetic representation (to allow searching for rhymes), and thesaurus category. All of these secondary indices can be built automatically using existing tools.

Partitions

A partition represents a set of disjoint regions in the corpus. Each region has a name and one or more tuples of start and stop positions in the corpus, as regions need not be contiguous.

Partitions are used to restrict searches to certain parts of a corpus and to determine which region encompasses a location. For example, suppose that we have indexed Shakespeare's complete works and built a *play* partition. The *play* partition has regions corresponding to the locations of the plays in the corpus. Then we can search for *mad* and only return matches from *Hamlet* or do a search for *How now* and find all the plays in which that phrase occurs.

Partitions might also be used for searching metadata fields, measures in a piece of music, and couplets in a poem. A partition on metadata fields such as author, title, and subject allows for metadata-based searching and

retrieval.

Parallel Indices

A parallel index maps every position in one corpus, the primary corpus, into another corpus, the parallel corpus. The parallel index links every position in the primary corpus to one or more positions in the parallel corpus.

Generally, text is processed in a linear fashion. We read one word at a time, from the beginning of a sentence to the end. Parallel indices allow us to process information that occurs simultaneously instead of sequentially. In choral music, for example, the tenors might be singing one word while the sopranos are singing another. Using a parallel index enables searching for those two words being sung at the same time.

Parallel indices are also well suited for searching and displaying word alignments in multilingual corpora. One can encode multiple translations of a source collection separately, and then use position alignment maps both to constrain searches to certain translations and to display glosses (i.e., matching or equivalent sections) in other languages.

Searching

The types of available searches depend entirely on which indices have been constructed. In the next several sections, we offer examples that illustrate the power of our search engine.[8]

Shakespeare

This first example displays some matches from a search for the lemma *tear/V* in the complete works of Shakespeare. The results show which play contains each match (using a partition on the plays) and the context of the match.

```
midsummers|Bacchanals , [Tearing] the Thracian sin
kinglear  |islocate and [tear] Thy flesh and bones
kingrichar|k At meeting [tears] the cloudy cheeks
romeoandju| mandrakes ' [torn] out of the earth,
```

The example below shows some output from a search for the word *upon* followed by a determiner (DT), an adjective (JJ), a noun (NN), and a comma. In this case the search engine automatically lines up the text with the part of speech.

```
kingricha|le heaven [upon this fair conjunction ,]
         |P  NN      IN   DT   JJ   NN          ,
kinghenry|is  fixed [upon a  spherical stone ,] wh
         |VBZ VBN    IN   DT JJ        NN    ,  WD
hamlet   |, so jump [upon this bloody question ,]
         |, RB VBP   IN   DT   JJ     NN       ,
romeoandj|e heavens [upon this holy act ,] That af
         |  NNS      IN   DT   JJ   NN  ,  DT   IN
```

The next example is output from a search for the lemma *eat/V*, followed by anything, followed by a word in the *food* category (a secondary index generated from a thesaurus). The output includes identifiers for both the play and play act in which the match occurs.

```
kingrichard III|  Eating the bitter   bread
allswelltha II |  eat    no           grapes
measurefore III|  eat                 mutton
cymbeline   III|  eats   our          victuals
measurefore III|  eaten  up  all her  beef
```

Music

The example below shows the result of searching for the note *C*, followed by the pitch moving up, then down, and ending with an *A*. The output shows the context and the clef (from a partition) for each match. Pitch, direction of pitch change to the next note, and duration are displayed for each note. Pitch, direction, and duration are all secondary indices.

The corpus is a modified version of music notation format Guido (Hoos and Hamel 1997) generated from Levy Phase II (Choudhury et al. 2000).

```
treble |e    e    d        [c   b    c    a]      c    b
       |-    \    \         \   /    \    /       \    \
       |1/4  1/8  1/8      1/8  1/8  1/8  1/8     1/4  1/4
treble |d    d    d        [c   b    c    a]      c    b
       |-    -    \       \    /     \    /       \    \
       |1/4  1/8  1/8      1/8 1/8  1/8  1/8     1/4  1/4
```

Conclusions

Name authority control provides a powerful mechanism for information discovery; however, manually converting a large digital collection is expensive and time-consuming. Our automated name authority control system will help reduce the amount of human labor required for this process.

The search engine provides a powerful and versatile means of full-text searching over the variety of documents that digital libraries need to index. Secondary indices allow powerful forms of searching, from syllables, to rhymes, to parts of speech. Partitions support metadata searching and provide a convenient way to segment documents. Parallel indices allow unified retrieval from multiple translations of the same document.

Combining enhancements to the metadata with powerful full-text search capabilities will enhance intellectual access to the online Levy Collection. Because the tools are generalized, other collections can benefit from these improvements.

Acknowledgements

We would like to thank David Yarowsky of the Department of Computer Science for motivation and discussion, and Michael Droettboom of the Peabody Conservatory of Music for help with searching Guido.

The second phase of the Levy Project is funded through the NSF's DLI-2 initiative (Award #9817430), an IMLS National Leadership Grant, and support from the Levy Family.

Notes

[1] Digital Knowledge Center, Milton S. Eisenhower Library, <http://dkc.mse.jhu.edu>

[2] Milton S. Eisenhower Library, <http://milton.mse.jhu.edu/>

[3] Johns Hopkins University, <http://www.jhu.edu/>

[4] The Lester S. Levy Collection of Sheet Music, <http://levysheetmusic.mse.jhu.edu/>

[5] Library of Congress (LC) name authority file, <http://www.loc.gov/marc/authority/index.html>

[6] In the Levy Collection seed data, for example, 100% of the authors who had a death date in the authority file had no posthumous publications in the seed data. (This may not be true for the entire collection or in other collections.)

[7] Part of speech tags derived using the Penn TreeBank tag set with Eric Brill's rule-based tagger. The tagger is available at <http://research.microsoft.com/~brill/>.

[8] See <http://dkc.mse.jhu.edu/projects/search> for more examples.

References

Choudhury, G.S. C. Requardt, I. Fujinaga, T. DiLauro, E.W. Brown, J.W. Warner, and B. Harrington. (2000). Digital Workflow Management: the Lester S. Levy Digitized Collection of Sheet Music. *First Monday*, volume 5, number 6. <http://firstmonday.org/issues/issue5_6/choudhury/index.html>

Choudhury, G.S., T. DiLauro, M. Droettboom, I. Fujinaga, and K. MacMillan (2001). Strike Up the Score: Deriving Searchable and Playable Digital Formats from Sheet Music. *D-Lib Magazine*, volume 7, number 2. <http://www.dlib.org/dlib/february01/choudhury/02choudhury.html>

Hoos, H. H. and K. A. Hamel (1997). *The GUIDO Music Notation Format Version 1.0, Specification Part 1: Basic GUIDO*. Technical Report TI 20/97, Technische Universit&#65533;t Darmstadt. <http://www.informatik.tu-darmstadt.de/AFS/GUIDO/docu/spec1.htm>

Warner, J.W. and E.W. Brown (2001). Automated name authority control. (To be published in *Proceedings of JCDL '01*).

Whitten, I., A. Moffat, T. Bell. (1999). *Managing gigabytes*. 2nd Ed.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*: 189-96.

Top | Contents
Search | Author Index | Title Index | Back Issues
Previous Article | Next Article
Home | E-mail the Editor

DOI: 10.1045/april2001-dilauro