NACO Normalization

A Detailed Examination of the Authority File Comparison Rules

Thomas B. Hickey, Jenny Toves, and Edward T. O'Neill

Normalization rules are essential for interoperability between bibliographic systems. In the process of working with Name Authority Cooperative Program (NACO) authority files to match records with Functional Requirements for Bibliographic Records (FRBR) and developing the Faceted Application of Subject Terminology (FAST) subject heading schema, the authors found inconsistencies in independently created NACO normalization implementations. Investigating these, the authors found ambiguities in the NACO standard that need resolution, and came to conclusions on how the procedure could be simplified with little impact on matching headings. To encourage others to test their software for compliance with the current rules, the authors have established a Web site that has test files and interactive services showing their current implementation.

Sharing data between bibliographic systems requires the ability to compare two pieces of information to determine if they are intellectually equivalent regardless of the ways in which they are stored. The authors attempted to compare data created by disparate systems but theoretically normalized by the same rules, and discovered discrepancies. Researching the problem headings revealed that the NACO normalization rules are vague in some aspects and possibly too restrictive in others. Three independently developed implementations of the Program for Cooperative Cataloging's (PCC) Name Authority Cooperative Program (NACO) normalization rules were brought into agreement with each other through the use of a common test environment, which the authors have made publicly available. Areas in need of clarification and simplification were identified during the testing.

Normalization rules can be used to create a standard or generic form for headings and other similar alphanumeric strings. This standard form is essential for clustering logically identical headings and differentiating between logically different headings. The need to determine the equivalence of two headings arises frequently in work with both name and subject authority files. What characteristics of the heading are significant? Should capitalization, spacing, and punctuation be ignored? What about special characters? Are *Smith-Jones* and *Smith & Jones* the same? What about *Black jack* and *Black, Jack*? Depending on which rules are followed and how they are implemented, these may or may not be considered equivalent.

Normalization is the transformation of a string of characters into a more generic form. Typical transformations include reducing all alphabetic characters to a single case and eliminating diacritics and punctuation. The justification for this is that minor differences between headings do not affect whether the headings are considered the same. In actual use, normalization rules go beyond simple string transformations and often take into account the context in which the strings are used. For example, should headings representing corporate enti-

Thomas B. Hickey (hickey@oclc.org) is Chief Scientist, **Jenny Toves** (tovesj@oclc.org) is Software Architect, and **Edward T. O'Neill** (oneill@oclc.org) is Consulting Research Scientist, Office of Research, OCLC Online Computer Library Center.

ties be unique not only from other corporate headings, but also from cross personal headings? The authors will focus on the basic string transformation rules.

A wide variety of approaches to normalization are used; some are very simple, others quite complex. A simple scheme is to retain only digits and alphabetic characters (as lowercase characters), dropping all other characters. Using this scheme, the string

A. Hann & Son (Bridgeton, N.J.)

normalizes to:

ahannsonbridgetonnj

This simple approach works well in many situations, but better options are usually available.

Several principles contribute to a good normalization algorithm. A normalization algorithm should be:

- Intuitive. The result is consistent with human judgment. Two strings that are generally perceived as equivalent produce the same normalized result.
- Simple. Normalization, especially when it is used across various systems or applications, is as simple and straightforward as possible.
- Repeatable. Running the normalization routine on previously normalized strings does not result in additional changes.
- Generalizable. The algorithm avoids content- or application-specific rules. This enables systems using the rule to more easily accommodate new types of data and promotes interoperability.
- Sortable. Ideally, the normalized strings can be used to sequence or sort the original entries.

Normalization can be used to meet the general need to group headings, titles, and other strings that are logically identical but have different representations. For example, one of the authors has a surname that contains two capitals and a special character, and is, therefore, often represented in a variety of ways, such as *Oneill*, *O Neill*, *O'neill*, and so on. The simple scheme of retaining only alphanumeric characters would normalize these three variants to *oneill,* effectively grouping these variants.

Algorithms vary in their strength. Strong algorithms will generate the same normalized result for most variants, but may also include numerous mismatches, and weak algorithms often will fail to create the same result for all variants but will rarely mismatch strings. For example, the simple normalization algorithm discussed above would normalize *Edward O'Neill* and *Edwardo Neill* identically, while a weaker algorithm that retained spaces would produce different results for *O'Neill* and *O Neill*.

The weaker algorithms may only standardize case and drop diacritics. The OCLC Online Computer Library Center (OCLC) search keys are an example of strong normalization.[1] Personal names are reduced to a 4, 3, 1 key containing the first four characters of the surname, the first three characters of the forename, and the middle initial. OCLC's search keys have been studied extensively, and much of that methodology is applicable to evaluating normalization algorithms. Llinas examined search keys in detail and provides a detailed review of related studies.[2] While these search keys are very effective at collecting variants, they generally lack precision.

In some cases, the data and normalization scheme are closely linked. This is the case with both the LC/NACO authority file and the Library of Congress Subject Headings (LCSH), both of which adhere to the NACO *Authority File Comparison Rules (NACO Normalization)*.[3] NACO participants create and maintain name authority records. NACO rules define the normalization procedures used to detect conflicts. Each established heading must be unique, and the file comparison rules are used to determine uniqueness. These rules are used not only to match variants, but also to define what differences are significant. By definition, if two headings have different normalized forms, they are different.

The NACO rules have been designed for use with bibliographic data and, in particular, with authority records. As a result, the NACO rules have a special status; they are the only widely used standard for normalization of bibliographic data. Because the rules are used to define what differences are significant, any deviation from the rules could produce erroneous results. Algorithms implementing the NACO rules must be accurately and consistently applied to avoid the creation of either duplicate or undifferentiated authorities.

Bibliographic Applications

When all library operations were performed manually, some sort of normalization was done, either consciously or unconsciously, when establishing headings and filing the resulting entries. These were probably first codified as filing rules (for example, Cutter's *Rules for a Dictionary Catalog*, *ALA Filing Rules*).[4] As machine-readable cataloging and authority records began to be exchanged, a systematic way of normalizing headings was essential to improve matches and avoid inadvertent collisions of headings across systems. The Linked Systems Project developed the set of rules that have now become the NACO *Authority File Comparison Rules*.[5]

Appendix 1 of the PCC Cataloging Standing Committees of Automation and Standards Joint Task Group on Streamlining Authority Record Creation *Final Report*

reviews the rationale behind much of this work.[6] It cites the main aspects of normalization, which include determining:

- what to regard and what not to regard;
- how to treat case; and
- the conventions for translating special characters and symbols.

This is necessary in order for the headings to function properly for indexing and searching, and for checking for uniqueness.

Motivation

As the authors began to investigate how best to implement the Functional Requirements for Bibliographic Records (FRBR) and plan authority control for the Faceted Application of Subject Terminology (FAST) subject heading schema, it became obvious that normalization would fill an essential role.[7] This led the authors to the NACO normalization rules used to establish these headings as they sought to match headings in existing records to those in the LC authority file.[8]

Over the years, OCLC has developed several implementations of the NACO normalization rules. Even within OCLC's Office of Research the authors found at least three versions written in different computing languages for different applications. Reviewing these implementations, the authors found discrepancies in how some headings were handled and realized that, without systematic testing, bringing the algorithms into agreement was impossible. The resulting NACO Normalization Testbed is the authors' attempt to share the results of this work with the rest of the library community.

The Rules

The algorithm for normalizing headings is contained in Appendix A of the NACO *Authority File Comparison Rules*.[9] These rules are summarized in figure 1.

Figure 2 lists non-ASCII characters that are translated into the ASCII character set.

Resolving Ambiguity in the Rules

A few unresolved issues exist at the character mapping level:

- The musical sharp, ♯, is now a separate symbol from the hash mark, #. The authors map both of these to the hash mark.
- The rules refer to the "logical OR" and "logical NOT" symbols, which do not appear to exist in the MARC character set.

The final results can depend on the sequence in which transformations are performed. The authors assume that the rules of stripping leading and trailing blanks and collapsing multiple blanks are done last, so that any blanks introduced during the processing are treated the same as original blanks in the data.

In addition, when processing bibliographic records, one needs to be able to handle data that are questionable or even ill-formatted, such as multiple uses of subfield $a in names. If subfield $a is used multiple times and the first reduces

ASCII characters:	Normalize to:
A–Z, a–z	Retain in single case*
Leading and trailing blanks, all diacritics, ‘ [ ] |	Deleted
Super and subscript 0123456789	0123456789
Super and subscript +-()	Blank
! ¡ ? ¿ “() {} <> ; . / \ @ * % = ± ® £ $ © ℗ °	Blank
Spacing characters ^ ` ~ _	Blank
Subfield delimiters	Subfield delimiters are retained, except for the one preceding the data. The associated subfield codes are deleted and field tags are retained only for the decision on what fields should be matched
Commas	The first embedded comma in the **a** subfield is retained, others become blank
0-9, # & +	Retained unchanged

*The authors prefer lower case for readability.

Figure 1. Summary of NACO normalizations

	Characters											
non-ASCII input:	Æ,æ	Œ,oe	Đ,đ,ð	ɪ	Ł,ł,ℓ	Ø,ø,Ȯ,σ	Þ,þ	Ư,ư	α	β	γ	♭
Normalizes to:	ae	oe	d	i	l	o	th	u	a	b	y	♭

Figure 2. NACO handling of non-ASCII characters

to nothing, is the trailing subfield delimiter retained? The authors' algorithm drops it. Does the "first comma" rule apply to the second subfield $a? The authors only apply it to the initial subfield $a. If a subfield reduces to nothing, or nothing but a blank, should the subfield delimiter be retained? The authors do not retain it. Is the comma retained if nothing precedes it after the other transformations have been carried out (the rules only talk about what to do when there is nothing following the comma)? The authors drop the comma in this circumstance.

Effects of Various Rule Simplifications

How good are the NACO rules? They are quite good, but they lack both general applicability and repeatability. The fact that the rules explicitly rely on the MARC record structure limits their application. The "first comma in subfield $a" rule and the retention of the subfield codes restricts their application to MARC coded fields. As cataloging becomes metadata creation with increased use of such non-MARC formats as Dublin Core, this cross-domain restriction becomes increasingly significant. For example, if the rules were applied to Dublin Core elements, the results could be different from those obtained from MARC fields, as Dublin Core records lack subfield coding.

Another important observation about the normalization rules is that they are very ASCII-oriented. After conversion, the original extended ASCII in MARC21 is a subset of printable ASCII except for the flat sign and subfield delimiters. By using "F" for the flat sign and a backslash for the subfield delimiter, the resulting string becomes much easier to process and display, with no loss of information. In the near future, the normalization rules will have to be extended for Unicode, and they are already causing problems with transliterated Chinese names.[10] Before such extensions, one must assess both the strengths and weaknesses of the current rules.

NACO specifies that the first comma in an $a subfield is retained unless it is a terminal character. All other commas are converted to a blank. This first comma rule appears to violate several of the principles for a good normalization routine, particularly the repeatability principle. Repeatability requires the algorithm to leave a normalized result unchanged. Another way to view this principle is to require that any string (unnormalized, partially normalized, or fully normalized) will normalize the same. For example, repeated normalization will generate the following sequence:

$aMorrison, W. M. $q(William McCutchan), $d1867-1918
morrison, w m$william mccutchan$1867 1918
morrison w m$illiam mccutchan$867 1918
...
morrison w m

Because the character following the subfield delimiter is deleted, the sequence ends only after all commas and trailing subfields have been removed.

The retention of the delimiter is also inconsistent with the intuitive and simple normalization principles, particularly when processing patron input data. Patrons are prone to omit subfield coding and would generally consider

$aMorrison, W. M. $q(William McCutchan), $d1867-1918

and

Morrison, W. M. (William McCutchan), 1867-1918

to be equivalent, although they normalize differently as:

morrison, w m$william mccutchan$1867 1918

morrison w m william mccutchan 1867 1918

Headings with explicit subfield coding (such as MARC records) will frequently normalize differently from headings with implicit subfield coding (such as card catalogs, many OPAC displays). With or without explicit subfield coding, these headings should generate the same normalized form.

In their FRBR work, the authors use NACO normalization to normalize titles as well as names. Non-name fields can also have subfields other than $a as the first one, so they do not fit the comma processing rules very well. For titles, retaining the first comma is often undesirable.

Application to the LC Name Authorities

The only obvious justification for retaining either the subfield delimiter or the first comma is that it prevents a significant number of conflicts. The authors investigated the effect of eliminating these two rules. The NACO file comparison rules were specifically developed for application with the LC authority files, both name and subjects. Because the name authority is larger, and its comparison rules simpler, the name authorities were used for further testing and evaluation. All personal names, corporate names, conference and meeting names, uniform titles, and geographic names except name–titles entries were analyzed—5,664,878 established headings and 4,162,130 cross-references.

The analysis focused on identifying conflicts of the NACO Authority File (2004 LC Distribution version). All established headings and cross-references were normalized following the standard rules, and all conflicts were collected

and analyzed. As specified in the file comparison rules, conflicts occur when two or more established headings normalize the same, or a cross-reference and an established heading normalize the same. Cross-references are not required to be unique; the same cross-reference can appear in multiple authority records.

Two files, one for established headings and the other for cross-references, were derived from the name authority file. Among other elements, each entry included the Library of Congress Control Number (LCCN) of the authority record, the original heading, and the normalized heading. For the cross-references file, any $w subfields were ignored and duplicate cross-references were deleted resulting in a file of unique cross-references. Only 568 conflicts (0.01 percent of the established headings) between established headings were found. Some examples of the conflicts identified are:

100 Jayasree, S. [n 84109744]
100 Jayasree, S. [no2004124022]
100 Nguyen, Kim-Chi [n 78050801]
100 Nguy~^en, Kim Chi [no2004123058]
100 India [no 92007900]
151 India [n 80125948]

Even without normalization, almost half of conflicts observed, such as the first example, were exact matches. The other common pattern was where the same name was used for different types of names: *India* as a personal name versus *India* as a geographic name. Cases (in the second example above) in which the established forms were different but normalized the same were rare; only 183 conflicts of this type were observed. Therefore, variation in the implementation of the normalization routines does not appear to be a significant cause of these conflicts.

Conflicts between cross-references and established headings were more common; 4,424 conflicts of this type were observed. Some examples of these conflicts include:

130 Ship [n 83732520]
410 SHIP [Slater Hall Information Products (Firm); n 88628681]
100 P. C. H. [nr 98022649]
410 P. C. H. [Partido Comunista de Honduras; n 82166958]
100 Snail, A. [nb2003096351]
400 Snail, A. [Walker, Trevor M.; n 97016230]

As with the conflicts among established names, variation in the normalization procedures was not a significant cause of the conflicts.

To determine the potential impact of simplifying the rules, the process was then repeated using the simplified rules. The number of conflicts resulting from both the current rules and the simplified rules is shown in figure 3.

Some examples of the additional conflicts are:

100 Bastia, France [n 97025024]
151 Bastia (France) [n 79086801]
110 Seychelles Police Force. [n 85245780]
110 Seychelles. $b Police Force. [n 85245771]
100 Rajhonson Ramiandrasoa [n 98900710]
100 Rajhonson, Ramiandrasoa [no2003060568]

Not all of the additional conflicts represented different entities. In the second, and probably third, examples, the headings represent the same entity. A quick review of the conflicting established headings pairs indicated that a high proportion were probably duplicate headings that should be merged. If both headings represent distinct entities, switching to the simplified rules would require additional qualification for at least one heading. However, the list is short enough and the error rate sufficiently high to make resolving the conflicts either by merging the duplicate headings or by further qualification of valid headings practical. Even when the headings clearly are different, they often appear similar enough to confuse many users. The fact that a significant proportion of these additional conflicts are likely to be duplicates indicates that these are also difficult, even for skilled catalogers.

The NACO rules prohibit conflicts between established headings (1xx fields from authority records) or between established headings and cross-references (4xx fields). No cross-reference can normalize the same as an established heading. The name authority file was also analyzed to determine the number of additional conflicts resulting from the use of the simplified rules, and the results are also shown in figure 3.

Simplification resulted in a huge increase in conflicts between established headings and cross-references, but,

Conflict type	No. of conflicts with NACO normalization	Additional conflicts with simplified normalization
Established heading to established heading	568	186
Established heading to cross-reference (internal)	628	44,235
Established heading to cross-reference (external)	3,796	1,304

Figure 3. Number of conflicts

in the vast majority of cases, the conflicts were between an established heading and a cross-reference within the same authority record. While the NACO rules specify that a cross-reference "may not normalize to the same string as any [established heading] in the same or another record," the impact of intrarecord conflicts is very different.[11] A conflict between an established heading in one record and a cross-reference in another record does pose a serious problem by presenting contradictory information. Established headings are valid—cross-references are not. Internal conflicts, however, do not pose similar problems; for example consider the following:

010 n 81050809
040 DLC $b eng $c DLC
151 Naples (Fla.)
451 Naples, Fla. $w nnaa

In this authority record, using the simplified rules, both the established heading and the cross-reference normalize to "naples fla" because the $w subfield is excluded. However, there is no real conflict. At worst, the cross-reference is redundant; at best, it indicates that the form of the heading has changed. These conflicts do not pose a serious problem and could easily be dealt with by changing the NACO rules to specify that a cross-reference "may not normalize to the same string as any [established heading] in *another* record." All cross-references could be retained and no changes to the authority records would be required.

Switching to the simplified rules would then only require changing approximately 1,500 authority records. While not a trivial task, it is certainly possible. Immediately changing any records may not be necessary. Ignoring the internal conflicts, the simplified rules would increase the number of conflicts by only a third. Although these conflicts present a serious problem, catalogers have accommodated the current conflict rate without undue problems. It is not something that requires an instant solution.

FAST Project

The simplified normalization has been applied in the FAST (Faceted Application of Subject Terminology) project. FAST is a new subject heading schema derived from the Library of Congress Subject Heading (LCSH). FAST retains the LCSH vocabulary, but in a simplified syntax, and is designed to be applicable outside of the traditional AACR-MARC environment—environments in which explicit subfield structure is rare. To function in these environments where subfielding could not be assumed, FAST adopted the simplified rules.

The simplified rules have worked very well in this application. There have been a number of conflicts (*Black jack* versus *Black, Jack*), but they have been infrequent enough that they could be resolved by adding a qualifier to one of the headings. While adding the qualifier required extra work, FAST was improved by decreasing ambiguities. Many, if not most, of the conflicting headings would either be indistinguishable or confusing to the casual user.

The NACO Normalization Testbed

In the spirit of Moen's CIMI Z39.50 Interoperability Testbed, the authors have created a NACO Normalization Testbed to help the community come to a consensus on how the rules should be applied to headings.[12] The testbed (www.oclc.org/research/researchworks/naco) consists of:

- Files: Three files contain normalized and un-normalized strings that exercise aspects of the algorithm. The test cases demonstrate handling of all legal Unicode characters, the comma rule, and subfield delimiters. The test files were created according to a strict interpretation of the NACO rules. This means that the subfield delimiter is a 0x1F and the musical flat is an unprintable character.
- Code: Java and Python code is used to implement NACO normalization. These have been tested and produce consistent results with the test files.
- Demonstration Web page: The NACO Normalization Project service (http://labs.oclc.org/nacotestbed) (figure 4) allows visitors to input a heading and see the resulting normalized form. An option is available to invoke the simplified rules, where commas, subfield delimiters, and subfield codes are replaced with blanks.

Conclusion

The NACO normalization rules provide a very effective means to compare established headings. Ambiguities in the rules lead, however, to inconsistent implementation. The sources of the variation from three independent implementations were examined, documented, and resolved. The resulting normalization software is publicly available in the NACO Normalization Testbed to assist the community in the consistent implementation of the normalization rules.

The suitability of the rules also was explored. The major limitation identified is their reliance on MARC encoding. Format independence is becoming increasingly important, as the use of other metadata schemas, such as Dublin Core, grows. In addition, the rules were found to be only marginally suitable when used to normalize titles, publisher names,

Figure 4. Screen shot of NACO Testbed

and other similar bibliographic entries. To overcome these limitations, two changes are proposed: dropping the first comma exception, and converting the subfield delimiter to a blank. With these relatively minor changes, the normalization rules would be suitable for almost any Latin-1 string, regardless of format. The relatively small increase in the number of resulting conflicts is viewed as acceptable to achieve generalizability and repeatability.

References

1. OCLC, Searching WorldCat User Guide, www.oclc.org/support/documentation/worldcat/searching/userguide (accessed Dec. 19, 2005).
2. James Llinas, *A Method for Evaluating Search-key Performance,* Ph.D. dissertation, State University of New York at Buffalo, 1977.
3. NACO, *Authority File Comparison Rules (NACO Normalization),* Sept, 16, 1998, rev. Feb. 9, 2001, www.loc.gov/catdir/pcc/naco/normrule.html (accessed Mar. 24, 2006).
4. Charles Cutter, *Rules for a Dictionary Catalog* (Washington, D.C.: GPO, 1904); American Library Association, *A.L.A. Rules for Filing Catalog Cards* (Chicago: ALA, 1942).
5. Nita Dean and Karen Calhoun, "The Linked Systems Project Provides Current Access to Authority Information," *OCLC Newsletter* no. 187 (Sept./Oct. 1990): 19–20; NACO, *Authority File Comparison Rules.*
6. Program for Cooperative Cataloging Standing Committees of Automation and Standards, Joint Task Group on Streamlining Authority Record Creation, *Final Report, Appendix 1: Normalization Rules,* 1997, http://faculty.washington.edu/kiegel/pcc/appendix1.htm (accessed Dec. 19, 2005).
7. IFLA Study Group on the Functional Requirement for Bibliographic Records, *Functional Requirements for Bibliographic Records: Final Report* (München, Germany: K. G. Saur, 1998), www.ifla.org/VII/s13/frbr/frbr.pdf (accessed Mar. 25, 2006); Edward T. O'Neill and Lois Mai Chan, "FAST (Faceted Application of Subject Terminology): A Simplified Vocabulary Based on the Library of Congress Subject Headings," *IFLA Journal* 29, no. 4 (2003): 336–42; Thomas Hickey, Edward O'Neill, and Jenny Toves, "Experiments with IFLA Functional Requirements for Bibliographic Records (FRBR)," *D-Lib Magazine* 8, no. 9 (Sept. 2002), www.dlib.org/dlib/september02/hickey/09hickey.html (accessed Mar. 25, 2006); Thomas B Hickey and Edward T. O'Neill, "FRBRizing of WorldCat," *Cataloging & Classification Quarterly* 39, nos. 3/4 (2005): 239–51.
8. Library of Congress Authorities, http://authorities.loc.gov (accessed Mar. 25, 2006).
9. NACO, *Authority File Comparison Rules (NACO Normalization),* Sept. 16, 1998, rev. Feb. 09, 2001, Appendix A, www.loc.gov/catdir/pcc/naco/normrule.html#a (accessed Dec. 19, 2005).
10. His-chu Bolick, "Problems in the Establishment of Nonunique Chinese Personal Headings with Special Reference to NACO Guidelines and Vendor-Supplied Authority Control," *Library Resources & Technical Services* 43, no. 2 (Apr. 1999) 95–105.
11. NACO, *Authority File Comparison Rules.*
12. William E. Moen, *Z-Interop: A Z39.50 Interoperability Testbed Study,* www.unt.edu/zinterop (accessed Dec. 19, 2005).