SHARED PRINT PARTNERSHIP

Matching Algorithms Task Force

Submitted to the Shared Print Partnership Executive Committee May 2025

Executive Summary

Determining whether two bibliographic records describe the same item is fundamental to a wide range of activities essential to managing and stewarding library collections. In 2022, the Partnership for Shared Book Collections, now the Shared Print Partnership[1], charged the Matching Algorithms Task Force (hereafter, 'the Task Force') with examining algorithms used with bibliographic databases. The Task Force set out to document levels of matching needs and explore current algorithms, with the premise that there is no single benchmark for an adequate match but rather a range of needs for various use cases. The project's purpose was not to determine which algorithm is "best," but rather to shed light on how different algorithms can yield different results.

Each chosen algorithm represents a fundamentally different approach to the record matching problem, ensuring a comprehensive evaluation of matching strategies, and was chosen from organizations that provide transparency on their matching process. The approaches examined included a 'match key' approach derived from bibliographic data elements, a machine learning approach, and a record number matching approach:

- The Colorado Alliance's Gold Rush algorithm represents a "match key" approach, constructing fixed-length strings from elements of MARC records which serve as identifiers for matching. This has the benefit of being indexable in a database for quickly identifying grouped records.
- The California Digital Library's MARC-AI represents a machine learning approach, accommodating common variations in cataloging by weighting the similarity of important MARC fields.
- ReCAP's SCSB employs a control number dependent approach, using a mixture of unstructured text fields and identifiers to determine if records match.

And finally, while an opaque method, simple OCLC matching with and without OCLC number merging were used for comparison.

Findings pointed towards most algorithms performing best on modern English language records, while older records and non-English language records presenting varying degrees of difficulties to the different approaches.

Several areas for future research are presented, along with the need for open algorithms and open record identifiers.

[1] Shared Print Partnership website: https://sharedprint.org

Introduction

Why Matching Algorithms Matter

Determining whether two bibliographic records describe the same title is fundamental to a wide range of activities essential to managing and stewarding library collections. Identifying analogous records aids in efficient cataloging, resource sharing, collection management, and preservation. The algorithms computers follow to find matching records assume an ever growing importance as libraries increasingly seek to engage with their collections in a multi-library context by sharing resources at a very large scale. Understanding how record matching is performed and the varied effects of different approaches are critical for managing collections within and across libraries. Matching algorithms for bibliographic records are essential infrastructure for several aspects of work involving Shared Book Collections.

The Partnership charged the Task Force in 2022 to "Investigate and document current algorithms used by vendors, service providers, and open source tools" and related activities. In approaching this work, the Task Force is addressing its report to a wide range of stakeholders charged with managing, stewarding and providing access to print monograph collections. Task Force members have looked thoughtfully at the matching algorithm landscape and examined a set of representative algorithms; in sharing our work, we hope to allow planners, strategists, and decision makers to form deeper understandings of the strengths and weaknesses, precision and uncertainty, and implications of using different types of algorithms in common use cases of bibliographic matching. Broadening the community's understanding of how different algorithms operate and the effect on decision-making is essential for all libraries regardless of their involvement with shared print.

Looking particularly at shared book collections, bibliographic matching underpins functions such as identifying copies for retention or withdrawal, determining the number of copies committed to retention programs, determining overlap of copies within and between shared collections, and supporting discovery and fulfillment of shared print items. Activities such as metadata enhancement or remediation, digitization of print copies, preservation treatment, and validation of physical items, all rely on effective matching. Thus, matching algorithms are essential infrastructure because they support crucial decision-making. If two records are confidently matched, they can be treated as if the items they represent are the same. If two records match, there is some confidence that the items they represent can serve as replacements for each other if necessary, for instance. Items attached to records that do not match with others may be considered unique and good candidates for investments in future persistence such as protection from withdrawal, digitization, physical stabilization, or placement in environmental conditions that extend their lifespans, etc.

To understand why matching algorithms vary in their approaches and their effectiveness, it is helpful to know a bit about where bibliographic records came from. Wikipedia gives a nice

summary of the history of cataloging.[2] Bibliographic records for contemporary library collections were created over many decades, even over centuries, following evolving cataloging norms. For centuries, libraries kept lists of books. Rudimentary steps in what we now think of cataloging emerged in the 15th century when a bibliography was created in chronological order and with an alphabetical author index. The French government created the first standard for cataloging collections after the French Revolution. The basis for modern cataloging, at least in the Western World, was developed by the British Museum Library in the mid-19th century. There have been many changes to the standards since then. Today, we can create and share bibliographic records electronically, but many of these records were converted from paper records.

Digitization and sharing of bibliographic data greatly increased the need for matching infrastructure. Greater use of bibliographic matching highlights the difficulty that bibliographic inconsistencies can cause. Inherent to the matching process are false positive matches, where two different items are incorrectly matched as the same, and false negative matches, where two items that are the same do not match. Different algorithms produce different patterns of incorrect vs. correct matches; they also perform differently on different categories of content. Understanding these patterns, their shape, scale, and distribution in library collections enables more effective decision-making, planning, resource investments, and long-term access to the shared research base and cultural heritage shared and stewarded by libraries.

The work of the Matching Algorithms Task Force

The Partnership recognized the importance of understanding the relationships between records and the items they represent. In 2021 the Partnership charged the Unique Materials & Metadata Task Force with defining *unique* in the context of shared collections. The Unique Materials Task Force identified a set of primary and secondary data points for assessing the relationship between records.[3] Primary data points describe the entity represented by the record. This includes information about title, authorship, publication, physical characteristics, among others. Secondary data points include control numbers that have been assigned to publication, such as LCCN, ISBN, and OCLC numbers.

In 2022, the Partnership charged the Matching Algorithms Task Force with examining how concepts articulated by the Unique Materials Task Force are applied in practice, focusing on algorithms used with bibliographic databases. The Matching Algorithms Task Force set out to document levels of matching needs and explore current algorithms, including seeking input from experts in the field. The Task Force began with the premise that there is no single benchmark for an adequate match but rather a range of needs for various use cases. Similarly, the project's purpose was not to determine which algorithm is "best," but rather to shed light on how different algorithms can yield different results.

[2] Wikipedia summary of history of cataloging: https://en.wikipedia.org/wiki/Library_catalog
[3] Unique Materials Task Force's "Defining Unique Bibliographic Entities for Shared Print Infrastructure": https://sharedprint.org/wp-content/uploads/Defining_Unique_Final_Document_2021.pdf

To frame its work, the Task Force identified five use cases that have different needs for precision in matching. These use cases are valuable context for project findings.

Description	Use case 1: Identifying materials to withdraw	Use case 2: Resource sharing fulfillment	Use case 3: Identifying digitization candidates	Use case 4: Identifying resources that lack retention commitments	Use case 5: Collection analysis at scale for planning
Impact of matching errors	higher	medium	medium	medium	lower
Optimum matching	Reduced need for storage space and collection care, achieved without inadvertent loss.	Requester receives the item that meets their needs.	Eligible items are identified for digitization with few omissions and minimal unintended duplication.	More content is covered by retention commitments	Libraries are better able to allocate resources to fill gaps in collecting and reduce unnecessary duplication.
Consequences/ Risks of mismatching	Inadvertent loss of unique or scarce content of enduring value.	Requester receives the wrong item or a variant that does not meet needs.	Digitization candidates are omitted from consideration because they did not appear. Resources are wasted on re-digitizing items.	Retention candidates are omitted from consideration.	Matching errors lead to skewed plans for collecting or missed opportunities for resource sharing.

Figure 1: Core use cases for the work of the Task Force

The heart of the Task Force's work was a structured comparison of results for multiple algorithms applied to the same data sets. The algorithms used for this study were selected to represent salient differences in approach, such as match key construction, control number dependent matching algorithms, and the use of machine learning. The Task Force began its study using modern records for English-language materials for a clearer view of the algorithms' fundamental differences. Subsequent comparisons included data sets from different sources, used cataloging covering a longer period, and explored records using different character sets. Algorithm and data set selection also reflects some practical considerations. Due to the proprietary nature of some algorithms and operational factors for potential partners, the study relied heavily on systems managed by the Task Force members. Data sets were drawn from

collections of records already compiled for other purposes. The selected algorithms, data sets, and project methodology are described in detail below.

Environmental Scan

The Task Force approached known organizations producing or using record-matching algorithms that satisfied the criteria for algorithmic diversity and organizational transparency (discussed in more detail in the next section). The Colorado Alliance of Research Libraries, the California Digital Library, and ReCAP were principal participants. While there has been some rotation in the membership over time, the Task Force has also included a range of perspectives and expertise from HathiTrust, Eastern Academic Scholars Trust (EAST), and the Big Ten Academic Alliance (BTAA), as well as members knowledgeable in cataloging, collection development, preservation, and shared print/collaborative collections drawn from individual research libraries that belong to or support these consortia.

To further establish a background on the current state of record-matching efforts and to ensure input from a wide range of stakeholders, the team sought the input of a number of producers and consumers of record-matching processes beyond those directly involved in the work of the Task Force. The Task Force interviewed representatives from Project ReShare and the Partnership for Academic Library Collaboration and Innovation (PALCI); Ex Libris; Backstage Library Works; and EAST. Input from other relevant businesses and organizations were solicited, but our invitations to participate were not taken up.

These exchanges were generally an opportunity for a give and take, with the interviewees often as curious about the work of the Task Force as we were to inform ourselves. But it must be noted that many, whether for-profit or not, are engaged in this work as a revenue-generating operation; in some cases this could have affected their willingness to engage with the Task Force, or the level of detail or specificity they felt they could reasonably share.

Project ReShare

Project ReShare supports and enables open-source consortial resource sharing through a suite of products for consortium that have been implemented by PALCI, ConnectNY, and the Ivy Plus Libraries Confederation. The Task Force interviewed representatives from Project ReShare, including a session with one of their consortial partners, PALCI. These conversations provided the most details on any record-matching initiative not explored directly by the team itself.

Project ReShare has been investigating issues unique to multi-type consortia (e.g., including public libraries, community colleges, and research libraries) with varying levels of record quality and metadata types, staffing levels and in-house expertise. The FOLIO inventory model served as a suitable starting point for the Shared Inventory, but it proved too limited because it did not allow for clustering records from the individual libraries. Records had to be merged and then holdings attached—a resource intensive process that is difficult to undo. While this can work

well for large datasets from a single library, it is too slow for a large consortium and it makes iterative re-indexing costly and time consuming.

In response, Project ReShare developed purpose-built solutions that could be more agile and required less infrastructure. A new aggregation module was designed to ingest data from many catalogs and to cluster records according to one or more "pluggable" algorithms. Using record clusters, rather than merging records, gives consortia options. A single record can be a part of many different clusters, each of which can be output for a different purpose or use case: discovery, collection analysis, resource sharing, digitization, etc.

The intent is to allow implementers to use more than one strategy concurrently—like “Gold Rush OR ISBN+material type”, for example—or to build different cluster tables to meet different needs based on a single set of input records. And to do this efficiently: an early version of the PALCI shared index based on FOLIO’s inventory module took almost 6 months to load 80 million records; the new system can ingest the same number of records in a few days, largely by virtue of not doing anything to the data beyond what is necessary.

In terms of errors, from the Project ReShare perspective, false negatives (failing to find actual matches) translate directly into lower fill rates in resource sharing, and false positives (incorrect matches) are problematic for collection management. At this time, the effectiveness of the algorithms are being measured in terms of false positives, largely because these are easier to identify; false negatives require having a known dataset to test, or the items in hand. They have also found that error rates are higher for some types of materials (e.g. monographic series, music, maps) or for records from a particular library or group of libraries.

Areas for future development include resolving issues with non-Roman scripts (e.g., CJK, Cyrillic), better preprocessing protocols for data cleanup before creating deduplication vectors, and improvements in scaling/indexing time. Another priority would be authority control. Ideally, Project ReShare would like to see the library community have access to shared tools and practices for collection management and resource sharing that are independent of commercial companies.

Ex Libris/Clarivate

Ex Libris/Clarivate offer a suite of library system modules including Alma, Primo, Rialto, and Rapido/RapidILL that rely on record matching for acquisition, discovery, collection analysis, and resource sharing that can serve consortia as well as individual libraries. In support of collection analysis both for title/work level matching and comparing digital editions to print, Ex Libris is investigating both tight edition matching and higher level, FRBR-type groupings of bibliographic entities. Alma currently provides some normalization of title information (e.g. punctuation), and individual customers can tune the algorithm for different material types. Ex Libris does reconciliation in shared catalog implementations (Network Zone), and they have been exploring ways in which AI might improve matching efficiency and accuracy. In discussing best practices and the reliability/dependability of various MARC fields, they referenced the report from the Partnership’s Unique Materials & Metadata Task Force (see above, page 5).

Ex Libris has been working with data analysts on record matching, and they would welcome more development in this area. Specific topics mentioned include guidelines for classifying false positive and negative errors depending on the application, along with acceptable error rates for each. It would also be useful to have standardized baseline match criteria.

The Eastern Academic Scholars Trust

The Eastern Academic Scholars Trust (EAST) shared print program uses record matching to establish unique bibliographic expressions and the holding of each across their member libraries. EAST has used both GreenGlass and Gold Rush to help with their analyses, particularly in onboarding new members individually or as part of cohorts or consortium. EAST relies on OCLC number matching, with retention commitments being recorded in both the local catalog of the member library and OCLC's Shared Print Registration Service (among other national and regional catalogs). Because of this, records that lack OCLC numbers–such as legacy RLIN records–can be very problematic.

Backstage Library Works

Backstage Library Works has decades of experience providing retrospective catalog conversion services to libraries, as well as original cataloging (in many languages). As such, the team thought it worthwhile to see if BLW could provide data or insight on particular patterns or persistent issues in these areas, especially from their experience with re-con projects. However, because the work is largely bespoke and reliant on human-powered record matching rather than algorithms, their business has not produced the type of processes or data gathering that could inform the work of the Task Force.

OCLC

Record matching is fundamental to OCLC. Their WorldCat database is the largest union catalog, and the OCLC number serves as a base-level identifier for library records. They also provide systems and tools like ILLiad for resource sharing and GreenGlass and Choreo Insights for collection analytics. While we were unable to engage directly with OCLC, two recent documents shed valuable light on the directions of their efforts toward "continually exploring new methods for enriching, repairing, and de-duplicating WorldCat records": "Machine Learning and WorldCat: Improving Records for Cataloging and Discovery"[4] and "Implementing AI to Further Scale and Accelerate WorldCat De-duplication."[5]

[4] "Machine Learning and WorldCat: Improving Records for Cataloging and Discovery": https://hangingtogether.org/machine-learning-and-worldcat-improving-records-for-cataloging-and-discovery/

[5] "Implementing AI to Further Scale and Accelerate WorldCat De-duplication": https://www.oclc.org/en/news/announcements/2025/ai-worldcat-deduplication.html

Algorithms

Matching Algorithm Concepts

Matching algorithms attempt to solve the problem of determining equivalent entities across datasets (often called entity matching and record linkage). In the context of MARC records, these algorithms serve a critical function in library and archival information management, including for shared print. The core challenge addressed by these algorithms is reconciling bibliographic records that may represent the same physical work despite variations in cataloging practices, data entry, or metadata completeness.

Rationale for selecting algorithms in this study

The selection of algorithms for this comparative study was guided by two primary criteria:

1. **Algorithmic Diversity**: Each chosen algorithm represents a fundamentally different approach to the record matching problem, ensuring a comprehensive evaluation of matching strategies.
2. **Organizational Transparency**: The research prioritized algorithms from organizations willing to provide detailed insights into their approaches and open to testing across varied datasets. This transparency is crucial for understanding the strengths and limitations of each matching technique.

Algorithm types compared in study

Three types of algorithms were examined:

- "Match keys" based on bibliographic data
- Machine Learning/AI based matching
- Matching on standard control numbers

The Colorado Alliance's Gold Rush[6] algorithm represents a "match key" approach, constructing fixed-length strings from elements of MARC records which serve as identifiers for matching. This has the benefit of being indexable in a database for quick identification of grouped records.

The California Digital Library's MARC-AI[7] represents a machine learning approach, accommodating common variations in cataloging by weighting the similarity of important MARC fields.

Representing a control number dependent approach, ReCAP's SCSB (Shared Collection Service Bus)[8] uses a mixture of unstructured text fields and identifiers to determine if records match.

---

[6] Colorado Alliance's Gold Rush: https://coalliance.org/software/gold-rush
[7] California Digital Library's MARC-AI: https://github.com/cdlib/marc-ai
[8] ReCAP's SCSB: https://recap.princeton.edu/collections-services/shared-collections/shared-collection-service-bus-scsb

And finally, while an opaque method, simple OCLC matching with and without OCLC number merging were used for comparison.

Bibliographic Match Key Algorithm (example: Gold Rush)

To match bibliographic entities, one common approach is to use a combination of metadata from various fields in a cataloging record to build a unique hash string to use as a match key. Many different organizations have used this approach for applications such as union catalogs or resource sharing systems (e.g. INN-Reach, ReShare) or bibliographic analytic tools such as Gold Rush. A number of commercial library vendors may also be incorporating such an approach but their algorithms are proprietary and opaque to the user. One of the unique opportunities for the match key approach is that it works well for different versions of MARC and could also be used for BIBFRAME solutions using linked data.

An example of a Gold Rush match key can be found in Appendix I.

Motivation for the algorithm

This section will focus on the match key algorithm used by the Gold Rush analytics system as developed by the Colorado Alliance of Research Libraries.[9] The purpose of the Gold Rush system was to support individual and group library collection analytics for purposes such as shared print programs, collection building and analysis, weeding, space planning, retrospective purchases, and other use cases. Although Gold Rush is not a resource sharing system, where the system may have slightly different needs, the algorithm can be modified for different primary use cases.

The match key approach for Gold Rush was developed by the Colorado Alliance of Research Libraries for the sixteen member libraries in Colorado and Wyoming. The Colorado Alliance has fifteen academic and one public library. The libraries use a half dozen different library management systems and incorporate a mix of OCLC and non-OCLC MARC records. The non-OCLC records come from various vendors and some older RLIN records were still in the mix when the algorithm was developed. Gold Rush is now used by a variety of other consortia and shared print programs and the matching algorithm has been periodically modified to take into account the broader base of users.

Although the OCLC number is a widely used match point which is carefully managed by OCLC, many libraries tend to obtain a record during copy cataloging and do not update it later if OCLC merges duplicates into a new primary record. ISBNs did not come into widespread use until the 1970s, so earlier records do not contain this unique bibliographic identifier. In addition, during the early days of ISBN implementations, a few publishers conflated it with publisher numbers thus re-using it for multiple titles. LCCNs are another valuable match point but only appear on records which have been cataloged by the Library of Congress, leaving many records without

[9] See the Colorado Alliance website for more information about the Gold Rush platform and its matching algorithm: https://coalliance.org/faq-library-content-comparison-system

this match point, particularly for catalog records created outside of the United States.

Strengths and Targeted Use Cases

The match key approach was developed to enable matching based on character strings in selected fields in records without relying on control numbers. It can be used when many records lack control numbers or have not been consistently updated. In the case of Gold Rush, the matching algorithm was designed to make sure that different editions and formats of materials were treated separately (e.g. print and electronic versions of monographs are on discrete match keys). This was an important element to support the consortium's shared print program and detailed collection analytics. However, in a resource sharing system a match key could be adjusted to merge similar but different bibliographic entities during match key creation.

One of the primary advantages of using a match key for matching is that it is eye-readable and very easy to troubleshoot. In addition, if it is provided as an option in a data export, it can act as a unique tertiary bibliographic identifier for loading into third party software products for further analysis (e.g. Excel, Tableau).

Overview of the Gold Rush algorithm

Due to the wide variety of cataloging sources and the long bibliographic history of most collections, no single number (e.g. OCLC, ISBN) can be used for matching purposes. The match key developed for Gold Rush uses a variety of elements from the MARC record to bring together common bibliographic records but specifically avoids the use of OCLC numbers and ISBNs. One could build a hash string using these match points but coding would need to be added for their absence. No key is perfect, so this particular approach is periodically being modified to improve matching within the system. Other systems use their own matching algorithms to accomplish the same purpose and some of these have been consulted in building this algorithm.

For a match key algorithm to work, a great deal of normalization is necessary, or the matching will be very imperfect. The better the normalization, the better the matching in this approach. For instance, the length of the match key could be either variable or a fixed length. Gold Rush uses a fixed length match key that is easy to read and troubleshoot, although fill characters need to be supplied if a given element is shorter than the fixed length of the field. Fixed length keys can also be broken back out to identify individual sections of the originating record. MARC tags are supplied in the discussion below, but these same data elements could be obtained from a BIBFRAME record.

The order in which the highly normalized pieces of these elements are placed in the hash string is not particularly important if it is consistently applied across all records. In the case of Gold Rush, putting the title first makes sense for readability and troubleshooting. None of these individual fields or fragments of fields will define a unique bibliographic entity, but using these elements in combination with each other makes Gold Rush's approach quite effective.

- **Title** (Combine 245 $a $b $n $p (first occurrence))
  Generally, the primary title and subtitle could be included either with or without spaces.

However, by removing all spaces, the Gold Rush match key excludes errant spaces. All diacritics and punctuation are removed, as they can cause mismatches if differently encoded. Leading spaces, punctuation, and other non-eye-readable characters that are sometimes found at the beginning and ending of these fields are stripped. Typically, filing indicators determine where the title string should start. Number ($n) and part ($p) enumeration help differentiate certain series and other similar but different entities.

One problem presented by the title field is that one record may include a subtitle and another record will not, causing an unwanted mismatch.

In Gold Rush all characters are put in lower case, spaces are removed from the first 70 characters of the title, and punctuation and special characters are excluded from the title, but non-Roman vernacular is used as it occurs.

Where a title has non-Roman vernacular found in the 880 (e.g. Chinese, Japanese, Korean) this vernacular is incorporated into the match key and any transliterated title is not used. Transliterated titles are very problematic, as different records can use slightly different techniques causing a mismatch. Transliterated titles without an 880 vernacular version can pose a title-match problem but typically represent a small portion of library collections (serving users who depend on transliteration, unless the library has a collecting emphasis in some of these languages).

- **Publication Year** (008/11-14, 008/7-10, 264$c, 260$c)
  Determining the date to use in the match key can be done through an iterative process. Using the 4-digit integers found in the 008 field gives the cleanest values, but if usable 4-digit integers are not found in the 008, the algorithm falls back to a four-digit integer in the 264$c or 260$c (for pre-RDA records). If no dates are found or are ambiguous, the Gold Rush match key uses the default value such as 0000. If more than one date is present in the 008/7-14, Gold Rush uses the second date (usually the copyright date), unless the 008/06 is ‘r’ (reprint), in which case the first date is used.

- **Author/Main Entry** (100$a, 110$a, 111$a, 130$a)
  Author fields are particularly challenging for normalization, since some entries will have middle or first names and some not. To solve this problem, Gold Rush has done the following - diacritics and non-alphanumeric characters are automatically removed, convert everything to lowercase and pad with underscores (if needed) to 5 characters. This field is short to reduce the possibility of name variations between records. This is how it works in practice:

  Van Mellon, Richard = vanme

  VanMellon, R. = vanme

  DeSantis, Alan D. = desan

  De Santis, A.D. = desan

Bach, Johann Sebastian, 1685-1750 = bachj

- **Edition** (250$a)
  Representing edition statements in a match key has special complications due to the different ways editions can be represented in a bibliographic record. Variations include spelled out words or integers. For example, an edition may appear as 'Second Edition' or '2nd edition' or '2d' (Spanish) with all combinations of capitalization. To uniformly represent this in a match key it is easiest to turn these variations into integers. Another issue is that "first" editions might appear in some records but in other records the field is left blank. This needs to be treated uniformly in the match key. Gold Rush normalizes these types of edition statements.

- **Publisher** (260$b or 264$b)
  As with the author (main entry) field, the publisher field has many of the same challenges. The portion selected should be very short (e.g. 5 characters used in Gold Rush). Diacritics and non-alphanumeric characters are automatically removed, converted to lowercase and padded (if needed) with underscores to 5 characters. This field is short to reduce the possibility of publisher name variations between records. What this kind of normalization looks like in practice is:

    D.C. Heath = dchea

    D. C. Heath = dchea

    DC Heath = dchea

    D.C. Heath & Company = dchea

    D. C. Heath and Company = dchea

Additional fields considered for the match key

A number of other fields might be considered in match key design. Pagination can help differentiate between different printings. Standard numbers such as SuDocs (for enhanced matching of government documents), LCCN, ISBN/ISSN, and OCLC are present in many but not all records, along with a variety of other fields. But in practice, all of these can pose problems. For the Gold Rush match key, the decision was made to include a trailing character of either a "p" (print or physical) or "e" (electronic) which keeps these two formats separated in the system. This is particularly important if a system includes both print and electronic records so that they are discrete.

Known issues and development in progress

Match key development is all about field selection and data normalization. As with any matching algorithm, it is not perfect and there are some issues that cannot be completely resolved. The result is that some records inevitably become unmatched orphans in a system despite representing the same entity due to variations in data.

Another issue is that the decision to include indicators of print versus electronic content into the match key hinders the use case for comparing print and electronic data sets. For the use case of print/electronic comparison either this approach to match key construction needs to be altered, or it must be handled in a different way in the system.

One of the toughest issues in title normalization is for records with transliterated titles in the 245 where there is no original vernacular in a corresponding 880 field. The match rates can be improved by replacing the title with something like an LCCN (010) or ISBN/ISSN (020/022) but this does not correct all cases since these numbers are absent in many records. The LCCN only appears if the record was originally cataloged by the Library of Congress, and the ISBN did not come into use until the 1970s.

Both the source of records and local cataloging practices can present variations in records that could affect matching. For example, multi-part DVDs for a television series can be handled differently in local cataloging practice. Also, some libraries may not obtain cataloging records from OCLC but get them from other sources such as their integrated library system vendor, other vendors, or by grabbing them from some other library via a Z39.50 cataloging link.

The Colorado Alliance has learned that the match key approach for connecting MARC records should not be static and must change to accommodate new insights and metadata. Because the method of generating match keys is updated periodically, libraries should avoid the temptation to use the Gold Rush key as a permanent identifier.

Control Number Dependent Matching Algorithms (example: SCSB, Shared Collections Service Bus)

There are several identifiers in the form of control numbers that have been used to identify titles and works. Some of these like International Standard Book Number (ISBN) or the International Standard Serial Number (ISSN) are assigned at the time of publication. Other control numbers like the Library of Congress Control Number (LCCN) or OCLC number are attempts at identifying unique records to help libraries catalog resources or fill requests. Individuals and libraries can use these numbers as reference points to identify matching records without looking at bibliographic elements of a record. ReCAP's SCSB matching algorithm is an example of using control numbers to identify matches.

Motivation for the algorithm

ReCAP's Shared Collections Service Bus (SCSB) software is an open source platform developed to support its Discovery to Delivery Platform.[10] At the time it was developed, the primary intent of the matching algorithm was to set a retention commitment on one copy of each title in ReCAP's shared high-density storage facility. The idea was that items in storage that had a retention commitment may have a different fee structure than those that did not. As the program turned into reality, the participant's goals have evolved. ReCAP is actively working,

[10] See ReCAP's website for more information about the Discovery and Delivery program: https://recap.princeton.edu/collections-services/shared-collections

through a new understanding of how the matching algorithm should work as one of several pieces of information required for setting retention commitments.

Instead of analyzing bibliographic elements of records separately, either by computer or human, ReCAP decided to leverage control numbers instead. The theory is that control numbers are existing attempts in grouping records and using that information was more efficient than attempting to create matches from scratch.

Overview of the SCSB algorithm

The SCSB algorithm is simple in concept, although computationally intensive. The algorithm pulls all active OCLC numbers, LCCNs, ISBNs, ISSNs, and the first four words of the title from each bibliographic record. SCSB checks each record against all of the others. If any two of these elements match, the two records are identified as matching. Through this method, match groups are identified, and all records that have matched are given a unique identifier. Because the system uses the transitive property of equality, there are instances where only some records within a group technically match together. For example, if RecordA and RecordB match on OCLC and title, and RecordB and RecordC match on ISBN and LCCN, they will all be grouped together even though RecordA and RecordC did not directly match with each other. These identifiers are not permanent. New identifiers are generated if records are added or removed from a group.

The parsing of these fields is minimal. OCLC numbers are pulled from the 035$a. The field must contain the string "OCoLC." Only numerical digits are retained, and leading zeros are removed. LCCNs are extracted from the 010$a. Some of ReCAP's partner libraries are using visual placeholders (# or ^) to make spaces easier to see. These are removed as are whitespace at the beginning or end of the field. ISBNs (020$a) and ISSNs (022$a) are parsed similarly. Anything besides numerical digits are stripped away. The title is pulled from several fields.[11] Each field is parsed such that leading whitespace is removed, leading articles (a, an, or the) are removed, and two or more spaces are turned into one space. The first four words are kept and joined together with spaces. The SCSB algorithm has some protection against a bibliographic record having a single control number error because it requires two elements to confirm a match.

Known issues and development in progress

Utilizing control numbers is only possible when they are in the records. There are known cases where control numbers, such as ISBN numbers were reused. Many bibliographic records, correctly or erroneously, have multiple of the same control numbers. In some instances it depicts additional analysis that updates groups, but there are many instances where incorrect numbers are in records. Utilizing control numbers works best when libraries perform periodic reconciliation to keep the numbers up to date. For example, periodically updating their OCLC numbers. SCSB attempts to correct for some of these issues by requiring two match points. The

[11] In order that they are used:
245$a, 245$b, 245$p, 245$n, 246$a, 246$b, 130$a, 730$a, 740$a, 830$a

increased accuracy comes at the expense of complexity.

Computationally, checking for matches can be complicated. Adding in a new record requires several checks against every other record in the system. Records that have several control numbers (like multiple OCLC numbers) require checking many pairs of numbers. Adding records requires not only running back through the entire set but also identifying if new groups are made with records that did not previously match. There is no way to essentially create a permanent matching identifier similar to how key-based algorithms (see Gold Rush) work. All of the records also need to be in one place so that new records can be compared against existing records and possibly regrouped when necessary.

Machine Learning Matching Algorithms (example: MARC-AI)

Machine learning approaches offer powerful new ways to handle the complexities of matching. MARC records commonly have variations that make the design of matching algorithms very difficult, as seen in the previou sections. Titles can be transcribed into different subfields, author names can be abbreviated, physical measurements can be inconsistent, and fields can be omitted entirely. By taking a mathematical approach with machine learning, we can prepare it for these types of issues by training on a dataset of record pairs designed to surface common variations in cataloging.

Motivation for the algorithm

Matching MARC records traditionally relies heavily on identifiers like OCLC numbers assigned during cataloging. However as previously mentioned, not all records have these identifiers, and variations in cataloging styles can complicate matching. To address these challenges, the California Digital Library explored the use of AI to see the potential for improving clusters in Zephir, the metadata management system for the HathiTrust.[12] The result of this exploration was MARC-AI, an open source machine learning package for record matching without identifiers.[13] MARC-AI uses only the bibliographic information from MARC records, so no identifiers must be present in the records. The comparison process is composed of blocking, a method of reducing computations to a set of candidate pairs, feature extraction, and classification with a small neural network trained at the California Digital Library (CDL) that produces the final output.

Overview of the MARC-AI algorithm

Blocking

Similar to SCSB, MARC-AI compares pairs of records individually, so the number of comparisons grows quadratically with the number of records, making large collections a challenge to process. Blocking is a method of removing obvious non-matching pairs to reduce

[12] "CDL's Discovery & Delivery Team is Exploring AI to Improve Zephir Records": https://help.hathitrust.universityofcalifornia.edu/support/solutions/articles/9000246990-cdl-s-discovery-delivery-team-is-exploring-ai-to-improve-zephir-records

[13] See the MARC-AI GitHub page for the code, training dataset, and pretrained model: https://github.com/cdlib/marc-ai

the number of comparisons by forming “blocks” to match within. MARC-AI uses token (word) blocking so that records must share at least one word from the title, author, or publisher to go through the whole matching process. To avoid creating large blocks (e.g. a block for titles containing the word “the”), MARC-AI excludes the top 30% of total words by frequency. Blocking significantly cuts down on comparisons and runtime while retaining high recall.

Feature Extraction

Features are numerical values that represent characteristics or attributes of data that can be used as inputs to a neural network. In machine learning, features are typically scaled to values between 0 and 1 to ensure consistent processing. In MARC-AI, these features capture how similar different fields are between two records, meaning for each field we compare, we must create a numerical feature for how similar they are.

- **Title**

  Comparing titles presents the greatest challenges. The title field contains primary titles, secondary titles, subtitles, and attribution information which varies greatly in completeness. This information is recorded inconsistently between records, making comparison difficult. Using a string comparison algorithm like Levenshtein distance is a common choice, but several issues arise when comparing titles with conventional string edit distance algorithms like Levenshtein:

  - Titles are split into subfields differently, so titles need to be compared holistically rather than on an individual subfield level. Concatenating the subfields doesn’t work well when different pieces of a title are present in a record pair, or title text is cataloged in a different order.
  - Not all text in the title field is equally important for matching. For example, author names and editors are sometimes included even in the primary title subfield.
  - Long titles make it difficult to properly penalize small, but meaningful, differences.

  To address the inconsistencies in catalog record formats, MARC-AI uses a specialized comparison algorithm that prioritizes text sections based on their significance by considering the original cataloger’s decisions. First, titles are normalized by making all characters lowercase, and removing punctuation and diacritics. The algorithm then categorizes MARC fields into two priority levels, essential fields and supporting fields. Essential fields[14] contain primary identifying information that we expect to find in the other record, while supporting fields[15] are used to fill in any missing pieces for the essential fields.

  When comparing two records, the algorithm extracts all of the words from the MARC

[14] Essential fields: 245$a, 245$p

[15] Supporting fields: 245$b, 245$c, 246$a, 246$b, 100$a, 700$a, 110$a, 710$a, 111$a, 711$a

fields, giving words from essential fields weights of 1, and weights of 0 if from supporting fields. MARC-AI calculates the similarity score as the ratio of total word weight missing from each record to the overall total word weight. This approach provides flexibility for the location of the essential words in the record, focusing on the presence of critical identifying information rather than expecting exact field-to-field correspondence.

- **Author**
  To compare authors, MARC-AI aggregates names into three groups, the 100$a/700$a, 110$a/710$a, and 111$a/711$a, separating names by spaces. All characters are made lowercase and punctuation and diacritics are removed. The string comparison algorithm called Token Sort Ratio is used to compare these three groups between the records, comparing the words without regard to order. This approach to comparing authors helps in cases where names are reversed or entered in different fields by the cataloger. If no authors of the same group are found between records, 0.5 is used by default.

- **Publisher**
  The publisher field (264$b) is normalized by making all characters lowercase and removing punctuation and diacritics. Similar to the author fields, the publisher is then compared with Token Sort Ratio. 0.5 similarity is also used as a default if any publisher is missing.

- **Publication Date**
  To compare publication dates, MARC-AI uses a function to map the difference between the years (008/7-10) to a similarity value, such that zero year difference results in a similarity of 1, quickly falling off as the year difference increases. An inverted sigmoid function was chosen to place an asymptote at zero similarity to help the neural network train. 0.5 similarity was used as default where dates were not numeric (e.g. 19uu) or omitted.

- **Publication Place**
  Publication places were matched exactly, getting 1 or 0 similarity based on whether the characters of the 008/15-17 matched. In any case where a record's 008 indicated the publication place was missing, MARC-AI used a default of 0.5 similarity.

- **Pagination**
  To compare pagination, MARC-AI uses regular expressions to extract numbers that look like page numbers from both records. If there is any intersection between the set of numbers found in the records, the pagination matches.

Neural Network

The neural network in MARC-AI is a multilayer perceptron that is responsible for calculating a value between 0 and 1 called the confidence score, based on the individual field similarity values determined in the previous step. This score represents how "confident" the model is that the two records match. The user chooses a threshold based on the use case to determine if records are similar enough to be considered a match. For the purposes of this analysis, we used a 0.99 threshold to minimize false positives while slightly increasing false negatives.

The neural network was trained using a custom dataset of about 50,000 MARC record pairs originating from HathiTrust contributors. The dataset is difficult, containing different monographs with similar MARC fields (doppelgangers), and identical monographs with variation in cataloging. From this dataset, train, validation, and test sets were randomly sampled (60%, 20%, 20% respectively). The model reached 98.57% accuracy on the validation set and 98.43% on the test set.

Known issues and development in progress

The largest challenge with MARC-AI is the relatively small number of MARC fields that are reliable for training. It is very difficult to teach the model the importance of fields that are often missing, so the model's small set of features were designed to combine multiple fields so that together they were useful to the model in training. A large language model may not have this issue as it can read the whole content of the record, but sacrifices control, speed, and transparency at a high cost. We see potential for improvement in the use of 008 dates, edition statements, abbreviation handling for author names and publishers, and usage of 880 fields when appropriate.

One limitation of MARC-AI is the training on English-language monographs. Though there is nothing specific to English about the processing, the assumption can't be made that the weights learnt through training on English-language monographs translates well to other languages, especially when string comparison algorithms are designed for alphabet-based writing systems. MARC-AI has been tested on non-Roman scripts at CDL and as part of the Task Force with good results, but feature engineering could be adjusted for even better matching quality.

Similar to SCSB, scalability is a challenge for MARC-AI because it compares individual pairs of records. With the addition of any new records to a collection, they must be compared to every other record. Blocking alleviates this problem to an extent, but only so much blocking can be done before sacrificing matching quality.

OCLC Number Matching Benchmark

Looking for matching OCLC numbers has often been employed as an efficient way to identify related records. For this reason, we anticipated interest in whether there is a meaningful difference in results between matching algorithms and strict matching on OCLC numbers alone. Using OCLC numbers from the 035$a, we created comparison sets for the algorithms we studied, using two versions of the OCLC number matches as benchmarks.

Primary OCLC

Primary OCLC matches used only the first 035$a with an OCLC number (a valid OCLC prefix followed by a number). Records with the same primary OCLC were matched.

OCLC Number Set

OCLC Number Set matching used the primary OCLC number, but also consulted the WorldCat API to create a set of OCLC numbers that have been merged. Records that contain OCLC

numbers in the same set were matched. This approach creates more matches where records contain out-of-date OCLC numbers.

Comparing Matching Algorithms

Methodology and Observations

As described above, three matching algorithms were chosen for comparison, along with two familiar benchmarks based on OCLC numbers present in the record. In looking at the three target matching algorithms we treated the matches made by OCLC and reconciled OCLC numbers as additional matching algorithms. For all intents and purposes, we thus had five algorithms - Gold Rush, SCSB, MARC-AI, OCLC numbers, and reconciled OCLC numbers.

Using these five comparators, we analyzed three scenarios represented by selected record sets. Each scenario compared records from two libraries, with records obtained from a different pair of libraries in each scenario. Scenario 1 used a convenience set of MARC records for English-language monographs from 2013-2017. These were used for our initial analysis. Using these modern English-language records as a starting point allowed for the comparisons to focus on how well algorithms worked on records drawn from large collections at libraries with well-resourced cataloging programs. Focusing on these materials allowed Task Force members (all native English speakers) to more confidently evaluate match success through record inspection. For Scenario 2 we analyzed records from the same two libraries, but limited to non-Roman scripts from 2013-2017.[16] This set was intended to explore how well the algorithms, all of which were developed in the United States by English-speakers and reflected Anglo-American cataloging practices, worked with materials in non-European scripts. Scenario 3 explored older English-language monographs published prior to 1950. The intent of this final comparison was to explore how well the algorithms worked with older records while removing the variable of foreign languages and allow us to compare directly back to the first scenario.

Figure 2 shows an overview of the batches and analyses.[17] The first two lines show the number of individual bibliographic records drawn from each library. In each case, records matched together to form a "match group." Each match group contains at least two matched records, but sometimes more. There were many instances where all five matching processes found that groups of records were matched. We assumed that for match groups that all five algorithms agreed in matching were indeed correctly matched together. Instances where there were differences between algorithm results are of particular interest, and the team conceptualized these as the "Island of Uncertainty."

[16] Identified in position 35-37 in the MARC 008. Language included: *Chinese, Japanese, Russian Arabic, Hebrew, Korean, Thai, Indonesian, Ukrainian, Greek (modern), Greek (ancient), Malay, Serbian, Persian, Hindi, Bulgarian, Yiddish, Tamil, Urdu, and Bengali*

[17] Note that the three batches used different libraries for each. Representations of Library 1 and Library 2 are for readability, while in reality, there were a total of six libraries involved.

	English (2013-2017)		Non-Roman (2013-2017)		English (pre-1950)	
	# Records	# Match Groups	# Records	# Match Groups	# Records	# Match Groups
Library 1	62,276		228,403		50,655	
Library 2	54,402		108,423		18,706	
All Five Matched	36,120	18,051	59,410	29,676	3,411	1,705
Island of Uncertainty	6,051	2,967	90,796	44,886	8,129	4,021

Figure 2: Overview of matching outcomes for all three scenarios

To decide whether items in a match group were matched correctly, a sample of match groups in the Island was selected for manual review by individual Task Force members. Task Force members used a Python script to display side by side the randomly selected MARC records that showed matching discrepancies. Two Task Force members reviewed the first forty match groups in Scenario 1 to ensure consistent application of the criteria. Discrepancies and records of note were discussed with the entire Task Force. Based on a set of matching guidelines developed by the group (see Appendix II), the following fields were manually reviewed :

- Language (008/35-37)
- Place of pub. (008/15-17)
- Form of item (008/23)
- Pub dates (008/6-14)
- Author (100/110/111$abcd)
- Title and subtitle (245$ab)
- Title part (245$p)
- Title number (245$n)
- Edition (250$a)
- Publisher (260$abc and/or 264$abc)
- Physical description (300$ac)

After review, each match group was categorized as ‘yes’ (should have matched), ‘no’ (should not have matched), or ‘unsure’ (unable to determine). We are deeply appreciative of the expertise lent to us by Claudia Conrad and Barbara Cormack of the California Digital Library in providing expert reviewing of the many ‘unsure’ records to make final determinations on whether compared records should be categorized as matches or not or if they should be finally categorized as lacking sufficient data for the match to be evaluated.

With all “unsure” groups resolved into either matching or non-matching categories, the manual

review yielded a data set for analyzing algorithm results. When both an algorithm and manual review indicated a match, the result was deemed a "true positive." Likewise, if an algorithm indicated a match where manual review determined the group did not match, the result was deemed a "false positive." Using scenarios described below, the Task Force examined rates for true and false positives, observing where algorithms returned similar and different results. True and false negatives (where an algorithm failed to identify a group) were also recorded.

The team used Scenario 1 (recent, English-language materials) as a useful standard against which the other two Scenario record sets were compared. We made our sample size of the Scenario 1 records much larger to better support statistical analysis. The review of Scenario 1 matches looked at a bit over 1,600 match groups, giving us a 98% ± 3% confidence. The manual reviews of Scenarios 2 and 3 were of 400 match groups each, reducing our confidence to 95% ± 5%.

Scenario 1: recent English-language monographs (2013-2017)

	True Positives	False Positives
Gold Rush	91.32%	0.61%
SCSB	99.27%	0.97%
MARC-AI	96.46%	0.45%
OCLC Primary	95.05%	0.10%
OCLC Reconciled	97.76%	0.18%

Figure 3: Matching outcomes for Scenario 1 highlighting positive matches (at least 2 records matched across test sets)

For this record set, overall, all five matching approaches performed well. All five agreed on more than 85% of all matched groups leaving a relatively small Island of Uncertainty. False positives for all matching techniques were below 1%. The SCSB algorithm found an extraordinarily high rate of matches but also had 50% more false positives than the next closest algorithm (i.e. it overmatched to some extent, combining entities that were different).

Gold Rush[18] had the lowest rate of finding matches, but there may be relatively easy ways of

[18] A lot of our analysis focuses on Gold Rush. Because the key was constructed using elements of bibliographic records, it allowed the team not only to understand with more specificity what was driving matching, but also to identify what will make it more effective. The ways that OCLC creates OCLC numbers is opaque to us, and commenting on where it falters is difficult. SCSB uses a combination of control numbers, and similarly can not be easily strengthened by adjusting how it parses the numbers. MARC-AI is somewhere in the middle. We may be able to make recommendations, but exactly how it makes decisions is obscured by the machine learning strategy it uses.

increasing the rate without significantly impacting the false positive rate. About 20% of the groups that Gold Rush did not match had the equivalent of “First Edition” in one record with the edition being blank in the other. It may be reasonable to assume that monograph records with a blank edition statement are referencing the first edition.

We saw that some issues may be more difficult for Gold Rush to improve upon. About 12% of groups Gold Rush failed to match had one record with a blank author. About 24% had title problems, half of which were caused by minor variations.[19] About 10% of groups that failed to match had date differences, but about a quarter of those were due to one record having only Date 1 and the other having different Date 1 and Date 2 in the 008, causing mismatched dates. Two-thirds of the issues that involved dates in Gold Rush had different years in the 008. In about half of the instances where the dates were different, we still identified them as matching because of clearly matched or merged OCLC numbers. Other issues included ambiguity in dates, such as the 008 and 264 having different or multiple and overlapping dates within a few years and where the rest of the records matched. Over 16% of match failures were due to minor variations in how publishers were recorded.[20]

The issues above also apply to MARC-AI, although many of which had lower impact. Records where there an 008 with Date 1 alone was compared with records with a Date 1 and Date 2 that differed appeared in about 20% of MARC-AI’s match failures. MARC-AI currently only uses the Date 1, so these could be cases where the Date 2 is important to consider. About 15% of failures to match included minor variations in publishers, cases where improved publisher string comparison could help the model match correctly

Three algorithms - SCSB, OCLC Primary, and OCLC Reconciled - do not directly use bibliographic information to create the matches (or more accurately, for the OCLC matches, we do not have an insight into how bibliographic information might be used) and thus a deep analysis of differences in bibliographic information was not practical.

---

[19] One may ask how often title variations that clearly are still the same item occur. We found it more common than expected. Examples include

"2012-2014 Southeastern Lie Theory Workshop Series :"
"Southeastern Lie Theory Workshop Series 2012-2014 :"

"...and a nti-imperial discourse"
"and anti-imperial discourse"

“reporter's journey through a country's descent into the darkness"
“reporter's journey through a country's descent into darkness"

[20] There are many different ways that publisher names can be expressed. A few examples of minor variations include

“SUNY” vs. “State University of New York”
“United States Holocaust Memorial Museum and German Historical Institute” vs “German Historical Institute.”
“Wiley” vs. “John Wiley & Sons, Inc.”
“Published by Routledge the Hakluyt Society” vs. “Routledge”

Scenario 2: recent non-Roman-language monographs (2013-2017)

	True Positives	False Positives
Gold Rush	46.75%	0.33%
SCSB	94.99%	2.57%
MARC-AI	91.70%	2.82%
OCLC Primary	87.16%	1.59%
OCLC Reconciled	89.04%	1.56%

Figure 4: Matching outcomes for Scenario 2 highlighting positive matches (at least 2 records matched across test sets)

Roughly 400 randomly selected match groups in the Island of Uncertainty were manually reviewed in this batch. As compared to English-language materials, non-Roman-language materials are a smaller, although still important, set of materials owned by North American libraries. The Task Force was also hampered in their review by our lack of expertise in reading non-Roman content. Based on the potential negative effects on matching of working with non-Roman materials in an English-based culture, our main interest was to explore how well the algorithms fared compared to the recent English-language monograph set. Here we were able to see the algorithms struggling, but in different ways.

A few things are striking at first glance. There was much more disagreement among the algorithms compared to the English-language test set. About 40% of match groups were identified as matching by all five algorithms, less than half the rate found in Batch 1. It is also worth noting that Gold Rush was an outlier in this Scenario, matching substantially fewer records overall than the other approaches. Within the Island of Uncertainty, where four or fewer of the matching processes produced overlapping match sets, the other four approaches all agreed on almost 77% of the match groups. Having one of the algorithms being this much of an outlier skews the analysis, including masking the effectiveness of other algorithms and confusing analysis to identify sources of false positive or negative matches.

SCSB and MARC-AI showed modest reductions in their performance identifying matches relative to the English-language record set, but also had much higher false positive rates. The outlier, Gold Rush, found less than half of the match groups that other evidence suggested should have matched, but also had very few false positives. Compared to SCSB and MARC-AI, both OCLC Primary number matching and OCLC Reconciled number matching had greater reductions in their success identifying matches, but, like Gold Rush, created fewer false positives. These results imply a balance that may be inherent to most algorithms, at least when operating on non-Roman record sets. Decreasing the rate of false positives often reduces the proportion of true matches found.

Parsing bibliographic records poses substantial difficulties to any matching activity. In our study

it was clear that diacritics or markings caused many issues.[21] While Gold Rush removes diacritics from some fields, it does not from others. About 42% of records that Gold Rush did not match had diacritics or markings in the publisher's name that were not removed. Similarly, removing them from editions would also improve correct matching, but only affected about 3% of the records. Interestingly, while Gold Rush does remove diacritics from titles, ligatures appear more persistent. About 10% of records that did not match had ligature differences that persisted through Gold Rush's processing. Transliterations, specifically differences between where records placed their vernacular and transliteration (or if one or the other was present at all) were counted in 11% of erroneously unmatched records. Thanks to the Task Force's research, many of these issues have already been corrected in more recent versions of the Gold Rush algorithm.

Scenario 3: English-language monographs (pre-1950)

	True Positives	False Positives
Gold Rush	76.05%	10.62%
SCSB	93.58%	6.13%
MARC-AI	87.21%	2.67%
OCLC Primary	21.85%	3.52%
OCLC Reconciled	91.19%	1.72%

Figure 5: Matching outcomes for Scenario 3 highlighting positive matches (at least 2 records matched across test sets)

The final set of records analyzed were for English-language titles published prior to 1951. Again, 400 match groups within the Island of Uncertainty were manually checked. While it is not possible to know when a title was acquired or cataloged, the team suspected that MARC records for older materials would be less standardized and more heterogeneous, which could contribute to matching errors.

Difficulty in matching older records proved to pose many challenges. Out of the match groups identified, all five techniques agreed on about 8% of them as matching. Most strikingly, and not

[21] The team was forced to learn more than it wanted to about encoding and language markings. Encoding differences at times contributed to confusion while reviewing records because different computers and software handle encodings differently. When software is able to read and interpret encodings incorrectly, it may represent on the screen differently than intended. Additionally, there are ways to encode diacritics as part of a letter or separate from the letter. When diacritics are separate, it counts as a separate character even if on the screen they appear as one. Some algorithms, including Gold Rush and SCSB count characters rather than obtaining an entire field. If the diacritics are not removed, the fields will not match even if on the screen they look identical.

surprisingly, the comparison of primary OCLC number in the record was a poor match, only finding about 22% of matched records. Matching via a primary OCLC number becomes increasingly difficult over time if libraries do not reconcile their records periodically. Reconciled OCLC numbers identified a great number of records that should match (91%) with a relatively low false positive rate (2%).

Side by side record comparison showed that for older records, inconsistencies in how dates are recorded can be a particular issue. As seen by Gold Rush's results that oddly had both a lower rate of finding matches and high rate of false positives. Date issues were prevalent in Gold Rush's false positives (29 out of 34, or 85%, of false positives). Gold Rush uses the 008 to find a date. If there are two dates, it will use the first date only if the records identify it as a reprint. For subsequent printings, Gold Rush uses the copyright date. This is in line with OCLC's guidance for making new records, which states that a "variation in manufacture or distribution date alone" does not justify a new record.[22] We found that 20 of the 29 Gold Rush false positives that had some sort of date issue were flagged as different at least partially because dates were different when the 008 indicated them as "Publication date and copyright date" and not a reprint but there were more than a ten-year difference in dates. For our reviewers, it caused enough ambiguity to mark most situations where there were differences in publication and copyright date in the 008 as different records for doubt's sake regardless of whether they were marked reprints or not.

Dirty data was also prevalent. More than 27% of match groups Gold Rush did not identify occurred because an author was fully missing on one of the records. A similar proportion of records had minor variations in publisher names. Records lacking subtitles plagued over 16% of false negatives.

SCSB, which requires multiple control numbers to match, caught the most actual matched records (93%), but also had a significant number of false positives (6%).

Being trained on a diverse mix of data with varying metadata quality, MARC-AI had the second lowest false positives after reconciled OCLC, but using only bibliographic metadata wasn't able to make as many matches as SCSB or the OCLC reconciled which use identifiers.

While older records did, as suspected, present more matching difficulties, it should be noted that they often account for a much smaller percentage of the total number of records in a system. That being said, these records do merit extra attention when evaluating matches.

Conclusion

The work of the Task Force confirms that at the broadest level the community's assumptions that matching algorithms handle modern English-language materials well, but may struggle

[22] "When to Input a New Record": https://www.oclc.org/bibformats/en/about/input.html

with older or non-Roman records. Additionally, it is important to regularly reconcile OCLC numbers. Matching, regardless of method, relies on data with bibliographic records. The evolution of bibliographic practice is healthy, but it can cause complications unless there are ways to upgrade older records created with previous standards.

Across the board, the various algorithms can be expected to achieve very high success rates of positive matches with only a very small percentage of false positives results for relatively recently cataloged English-language materials. Not only will the specific algorithms continue to be tweaked in small ways for marginal improvements in accuracy, they almost certainly will be joined in the future by new methodologies.

But this takeaway comes with conditions. All algorithm types produce different success rates across the items in our collections depending on the types of materials and the source of the records. Our standards, practices, and technologies have historically been developed for use on Anglo-American publications, including the use of Roman characters. The variability in older records is very real—and this is a large-scale problem. The earlier that a record was created, the more likely that local practices cause it to deviate from current standards. Similarly, success rates are significantly lower for publications in languages using diacritics or non-Roman characters because standards allow for, and local practices produce, variations in how these are rendered in MARC. Such issues as these that make record matching difficult can generally be described and categorized fairly easily, but due to the unpredictable nature of the specific/local variations make our ability to successfully account for them at a granular level using algorithms (including AI) that much more difficult. It should be noted that both of these record sets account for a relatively small, although important, group of materials held in the United States and Canada. The number of publications produced annually increased dramatically around the 1950s, in fact the median publication date is somewhere in the 1980s. Materials published prior to 1950 are estimated to be about 20%-25% of the collective collection within the United States and Canada.[23]

These observations confirm the need to follow standards. While flexibility may be desirable at the local level, it can be problematic in ways that are hard to overcome when we try to assemble collections for shared print programs and other collaborative or consortial initiatives through record matching. Any decisions to deviate from applicable standards have system level ramifications that are significant for matching. The community needs to grapple with the need to increase consistency of performance through both the conformance with cataloging standards and the flexibility allowed within the standard.

We need to recognize what algorithms do well or, to express it conversely, which of our records algorithms handle most successfully. We must be able to access our record sets, collectively and at the institutional level, to understand where the problem points may be so that they can be

[23] For more information, see:
Roger C. Schonfeld and Brian F. Lavoie, "Books without Boundaries: A Brief Tour of the System-Wide Print Book Collection," *The Journal of Electronic Publishing* 9, no. 2 (2006), https://doi.org/10.3998/3336451.0009.208.

addressed in the process. To do this requires transparency in how the algorithms function to provide a 'standard' toward which our non-standard record sets can be regularized.

Recommendations and Moving Forward

Matching algorithms are invisible infrastructure that are used for discovery, fulfillment, selection, de-selection, preservation, digitization, and other purposes. As we move into an increasingly interdependent environment, understanding how items are matched in any given system is increasingly important. Given that MARC records from a variety of sources will always vary in quality and often differ in significant ways due to variations in local practices, we acknowledge that there will never be a 'perfect' matching algorithm, and what may be 'good enough' in one case is inadequate in others. Below are areas that we feel merit further investigation

General Recommendations

- Matching algorithms should be transparent and flexible for varying use cases. Openness allows the community to assess and improve algorithms.
- In some cases using multiple algorithms can improve accuracy.
- Consider the types of records being matched in a given project. Look for ways to categorize cataloged materials to improve matching rates for specific subpopulations — for example, by publication date, record creation date, publication type, or language.
- Consider the categories of materials being compared and whether they affect accuracy potentially below an acceptable threshold.
- Recognize that catalogers are invaluable and investing in accurate, high-quality metadata is a necessity, not a luxury.
- Creating a standard test set of bibliographic records is highly encouraged. Such a set would include records that have common issues along with records that clearly match and those that don't. It would allow the community to benchmark new and existing algorithms and how well they function.

Continued investigation of current and evolving algorithms

We encourage the community to continue to investigate current and emerging matching algorithms in the following areas:

- Continue research with other vendor algorithms, and continue to advocate with vendors to be more open about how they match materials. Provide open evaluations of matching algorithms and their strengths and weaknesses, particularly compared to specific use cases.
- Do MARC records created outside of North America have special characteristics or fields that help or hurt matching? Are there areas for improvement in cross continent matching?
- Research the extent to which manual review is employed in record matching operations,

and to what extent does it enhance accurate match rates?
- Research into how BIBFRAME-based systems will match or differ from MARC-based approaches. Investigate ways to leverage the linked data model of BIBFRAME to make more confident matches.
- Are there actions that can be taken at the institutional level to improve matching?
- Explore ways to systematically address the most frequent transliteration variations to improve non-Roman character set record matching including the purpose and necessity of transliterations. Some work on this is being explored in the New York University Library's Knowledge Access department using AI tools.

Machine Learning / AI Opportunities

- Developing a national level group to work on sharable AI algorithms would be beneficial to the community. Given that these models can be adjusted for higher or lesser degrees of matching certainty, make recommendations on their uses for specific use cases. While major vendors are keeping their work proprietary, there may already be small pockets of this work taking place in open forums, e.g. AI4LAM[24], and the California Digital Library's work referenced in this article.
- Shared benchmarking datasets are what makes work like this possible. More complete datasets could be crowdsourced or created by the community to test matching algorithms more thoroughly.
- Recognizing that some improvements to matching cannot be based on MARC alone, and consulting the physical item may be necessary. To that end investigate if page scans or OCR can be systematically incorporated to further reduce errors.
- Invest in developing machine learning approaches that may give us much better and quicker ROI than cleaning dirty bibliographic data, which is endemic in library catalogs.
- There is potential in using machine learning to create vector embeddings of MARC records (a numerical fingerprint) which capture semantic and syntactic information from fields. This would allow for faster grouping similar to the match key approach using a vector database.

The need for open algorithms and identifiers

Having a non-proprietary method of matching bibliographic entities is extremely important to the success of many new open ventures. Having a known standard open identifier for works and expressions of works would be invaluable and unlock the potential for many cross comparison collaborative projects to be completed without the need to engage with expensive evaluation tools. While this has been acknowledged for at least a quarter century, with attempts such as the Book Item Component Identifier (BICI)[25] being proposed, it has yet to come to fruition. Other projects such as the Collaborative Collections Lifecycle Project[26] are also calling for open standard identifiers. As more libraries are participating in shared print retention with

[24] AI4LAM Website: https://sites.google.com/view/ai4lam
[25] Book Item Component Identifier: https://en.wikipedia.org/wiki/Book_Item_and_Component_Identifier
[26] Collaborative Collections Lifecycle Project: https://sites.google.com/view/cclifecycleproject/home

withdrawal programs, the need for better tools to leverage matching algorithms to make decisions based on existing retention commitments is increasingly important. Having both an open identifier and an open registry of committed identifiers would greatly benefit the community.

Acknowledgements

The Task Force would like to acknowledge Claudia Conrad's leadership and many contributions to this group. Claudia was the original chair of the Task Force, and her graciousness, patience, and deep expertise guided its work, and were greatly missed after her untimely passing at the end of 2023.

Task Force Members

Sara Amato
Ian Bogus (Chair)
Barbara Cormack
Andy Hart
George Machovec
Steve Smith
Karla Strieb
Raiden van Bronkhorst

Former members

David Almovodar
Claudia Conrad (Chair)
Judy Dobry
Dana Jemison

Appendix I: Gold Rush Match Key Example

"ontyrannytwentylessonsfromthetwentiethcentury_____2017____1__timdua_____snyde_____p"

000	03377cam a22006134i 4500
001	ocn968309193
003	OCoLC
005	20220118021716.0
008	170403s2017 nyu 000 0 eng
010	|a 2017000492
019	|a1201964573
020	|a9780804190114|q(trade pbk.)
020	|a0804190119|q(trade pbk.)
020	|z9780804190121|q(ebook)
024	8 |a99971776779
035	|a(OCoLC)968309193|z(OCoLC)1201964573
037	|bRandom House Inc, Attn Order Entry 400 Hahn rd, Westminster, MD, USA, 21157|nSAN 201-3975
040	|aDLC|beng|erda|cDLC|dYDX|dHBP|dNBO|dOCLCF|dIMT|dMYG|dZLM|dCZA|dJAI|dNDS| |dOCLCQ|dWFB|dOCLCQ|dBDP|dRBo|dIL4J6|dOCLCO|dUKEHC|dOCLCO |dIVU|dWYU
042	|apcc
043	|an-us---
049	|aWYUA
050	00|aJC495|b.S55 2017
082	00|a321.9|223
100	1 |aSnyder, Timothy,|eauthor.
245	10|aOn tyranny :|btwenty lessons from the twentieth century /|cTimothy Snyder.
250	|aFirst edition.
264	1|aNew York :|bTim Duggan Books,|c[2017]
264	4|c©2017
300	|a126 pages ;|c16 cm
336	|atext|btxt|2rdacontent
337	|aunmediated|bn|2rdamedia

338 |avolume|bnc|2rdacarrier

505 00|tDo not obey in advance --|tDefend institutions --|tBeware the one-party state --|tTake responsibility for the face of the world --|tRemember professional ethics --|tBe wary of paramilitaries --|tBe reflective if you must be armed --|tStand out --|tBe kind to our language --|tBelieve in truth --|tInvestigate --|tMake eye contact and small talk --|tPractice corporeal politics --|tEstablish a private life --|tContribute to good causes --|tLearn from peers in other countries --|tListen for dangerous words --|tBe calm when the unthinkable arrives --|tBe a patriot --|tBe as courageous as you can.

520 |aIn previous books, Holocaust historian Timothy Snyder dissected the events and values that enabled the rise of Hitler and Stalin and the execution of their catastrophic policies. With Twenty Lessons, Snyder draws from the darkest hours of the twentieth century to provide hope for the twenty-first. As he writes, "Americans are no wiser than the Europeans who saw democracy yield to fascism, Nazism and communism. Our one advantage is that we might learn from their experience."

650 0|aDespotism.

650 0|aHistory, Modern|y20th century.

650 0|aPolitical ethics.

650 0|aDemocracy|zUnited States.

650 0|aPolitical culture|zUnited States.

650 1|aTwentieth century.

650 7|aDemocracy.|2fast|0(OCoLC)fst00890077

650 7|aDespotism.|2fast|0(OCoLC)fst00891415

650 7|aHistory, Modern.|2fast|0(OCoLC)fst00958367

650 7|aPolitical culture.|2fast|0(OCoLC)fst01069263

650 7|aPolitical ethics.|2fast|0(OCoLC)fst01069286

651 7|aUnited States.|2fast|0(OCoLC)fst01204155

648 4|a1900-1999.

648 7|a1900-1999|2fast

710 2 |aJohn and Diane Cooke Gift.

907 |a.b100093668|b08-25-22|c01-18-22

998 |awc|b01-18-22|cm|da |e-|feng|gnyu|h0|i1

901 |a20220118|blc|cnew|dDLCcopy

994 |aC0|bWYU

946 |g1|iU181026482945|j0|lwcc |nGift of John and Diane Cooke.|o-|p$0.00|q-|r-|s- |t0|u3|v0|w2|x1|y.i4070175x|z01-18-22

Appendix II: Instructions/Guidelines for Reviewing

Matching Records

Picking Up and Viewing Files

Files are available for viewing in eye-readable MARC (.mrk) or MARC XML (.xml). Files will be made available for download from our group Google workspace.

Files can be viewed using Ian's Python viewer, which works with MARC XML, or, you can use a text editor like Notepad ++, which allows for viewing MARC files in either format. Notepad++ is preferable to Notepad, as it allows for searching within files for specific MARC records and allows for viewing files side by side.

To view two files side by side using Notepad++

1. Open Notepad++
2. Open file 1.
3. Open file 2.
4. Right click on the tab for either file 1 or 2
5. Click on "Move Document"
6. Click on "Move to Other View"

You should now be able to view your review files, side by side. If you don't already have Notepad++ on your computer, you can download it here: https://notepad-plus-plus.org/downloads/

Comparing Records

You will be reviewing a portion of records which matched, but did not match across all tested algorithms. The purpose of this review is to identify if/where there may have been problems in matching to highlight strengths and weaknesses of each.

To keep this as simple as possible, we will be reviewing only critical match data fields: minimum data needed to identify a match.

For our first assignment, we will be reviewing a small group of records. Everyone will have the same group of records to review. The purpose of this review is to make sure that everyone understands the instructions, and to clarify where needed, both with the reviewing process, and the actual instructions.

Reviewers will not be required to analyze edge cases, or make assumptions requiring cataloging expertise. This is just a strict Y and N evaluation.

We will be reviewing this assignment, as a group, at one of our meetings where we can talk through our findings. Following this, we'll regroup, modify instructions if needed, and move on to reviewing larger sets of data.

General Guidelines

You will be given a spreadsheet with record IDs from the two libraries. This spreadsheet will also include columns for the following information:

1. Match – Code as Y (yes), N (no), or U (not sure).
2. Field – Input MARC field code(s) which didn't match OR where you're not sure.
3. Notes – Input any notes you think may be relevant, but not required.
4. Review all fields listed in the instructions, below. Use ";" to separate tags and comments where more than one match point does not match.

   Example:

DANA's LIST OF MATCHES TO REVIEW				
Sys ID 1	**Sys ID 2**	**Match (Y/N)**	**MARC Tag**	**Notes**
JH123	DM987	N	245; 260	Different title; different publisher
JH456	DM765	Y		
JH102	DM911	Y		
JH230	DM427	N	008/35-37	Different languages
JH667	DM526	N	100	Author name differs.

Instructions

Review all the following fields.

Note that not all the fields listed may be present in each record. The absence of a field does NOT constitute a non-match. For example, if one record contains an edition statement (MARC 250) and the other does not, consider this a match if all other fields are the same.

For field 008:

Ignore case.

Note that back-slash (\), space ( ), and pound sign (#) are equivalent values for "blank" in the 008 field.

Note that the pipe symbol (|) and (?) have different meanings. "|" means "no attempt to code." "?" means attempted to code, but couldn't figure it out. Treat these as differing values from each other, and from "blank."

For all other fields:

Ignore punctuation (except for hyphens in dates).

Ignore diacritics.

Ignore spacing.

Ignore case.

Field specific anomalies will be specified, below.

Only compare those subfields specified for each field.

Fields for review

1. Language – 008/35-37 – Begin byte count at zero. For example:

```
0   1   2   3
01234567890123456789012345678901234567890
151023s2015\\\\ch\a\\\\\b\\\\000\0\eng\d
```

2. Place of publication – 008/15-17 – Begin byte count at zero. For example:

```
0   1   2   3
01234567890123456789012345678901234567890
151023s2015\\\\ch\a\\\\\b\\\\000\0\eng\d
```

3. Form of Item – 008/23 – Begin byte count at zero. For example:

```
0   1   2   3
01234567890123456789012345678901234567890
151023s2015\\\\ch\a\\\\\b\\\\000\0\eng\d
```

4. Publication dates - 008/06-14. For example:

```
0   1   2   3
01234567890123456789012345678901234567890
151023s2015\\\\ch\a\\\\\b\\\\000\0\eng\d
```

5. Author - 100, 110 or 111.

    a. 1XX field can be lacking in one record and still be considered a match.
    b. Where present, MARC field should match. For example, both records have a 100, or both records have a 110. If one has a 100 and one has a 110, this is considered a non-match.
    c. Review MARC field data from the following subfields only: abcd.

    Example:

    ```
    =100 1\$aMurano, Shir  Ō,$d1901-1975,$eauthor.
    ```

In this example, you are comparing subfields a and d with the other record. Ignore subfield 3, and ignore punctuation and diacritics. Subfield inclusion between records should match exactly. In this example, the matching record should contain a and d. Contents should match exactly, or consider it a non-match.

Example:

```
=100 1\$aBentley, G. E.,$cJr.$q(Gerald Eades),$d1930-
$eauthor.
```

In this example you are comparing subfields acd with the other record. Ignore punctuation. The other record should contain subfields a, c, and d, otherwise it's a non-match.

6. Title - 245.

    a. 245 must be present in both records.
    b. “&” and "and" can be considered equivalents.
    c. Abbreviations for volume, number, or part in subfields n and p can be considered to be equivalents.
    d. Review MARC field data from the following subfields only: abnp.

    Example:

    ```
    =245 14$aThe Avengers.$nVolume 6 /$cby Roy Thomas & John
    Buscema ; editor, Stan Lee.
    ```

    In this example, we are ignoring punctuation. Though 245 contents are considered to be a transcription of what's on the title page, "volume" in subfield n can be considered to be an equivalent in the match record if it's expressed as an abbreviation, where all else is equivalent, i.e., "Volume 6" vs. "V.6"

    Subfield inclusion between records should match exactly between matching records. In this example, a and n should also be present in the other record. If the other record also contained "p", for example, this would be considered a non-match.

7. Edition - 250.

    a. 250 may be lacking in match record. These are still considered to be a match if all else is equivalent.
    b. Numbers may be expressed in alpha or numeric characters and still be considered to be equal, i.e., "First" vs. "1st."
    c. Edition may be spelled out, abbreviated, or lacking altogether. 2nd edition = Second Ed. = 2

d. Review MARC field data from the following subfields only: a

Example:

```
=250 \\$aSecond edition /$bColin Lewis-Beck, Michael S.
Lewis-Beck.
```

In this example, we are ignoring any punctuation. "Second edition" would be equivalent to "2nd Ed." (or equivalent variants) in matching records.

8. Publisher - 260 and/or 264

a. 260 and 264 can be considered equivalents. One record may have a 260, and the other a 264, or they may both have 260s, or both have 264s, or some combination of the two.
b. Where 264(s) are present, second indicator must be "1". Otherwise, it is not publication information.
c. Publisher names may be abbreviated. Use your best judgement. When in doubt, mark as a non-match.
d. Multiple places of publication may be included. If one or more match the other record, you can consider this portion of the publication statement to be a match. Use your best judgement. When in doubt, mark as a non-match.
e. Multiple publishers may be included. If one or more match the other record, you can consider it to be a match. Use your best judgement. When in doubt, mark as a non-match.
f. A copyright, or other date, may be included in the date portion of this field. If included in both, they should match. If lacking in one, this is still considered to be a match. Publication dates must match exactly.
g. Review MARC field data from the following subfields only: abc.

Example:

```
=264 \1$aLos Angeles :$bSAGE,$c[2016]
```

In this example, we check to be sure second indicator is "1". Ignore punctuation and capitalization. If match record had place of publication as "Los Angeles, CA", as shown below, this would be considered a match,

```
=264 \1$aLos Angeles, CA :$bSAGE Pub.,$c2016.
```

Use your best judgement. Note that here, we ignore that one record has square brackets around the publication date, while the other does not. We also assume that SAGE and SAGE Pub. are the same publisher.

Example:

```
=008 121022r20132011enkabe\\\\\\\\001\0\eng\c
=260 \\$aLondon, England ;$aNew York :$bVerso,$c2013, 2011
```

In this example, we have two places of publication. If the match record has one or the other, all else being equal, it can be considered to be a match. Note the 2 dates in the $c. On further investigation of the 008 dates, this is a reprint/reissue of an earlier printing. The match record may or may not contain both values. When in doubt, mark as a non-match.

9. Physical Description - 300.

    a. Physical description is a free text field, so you will need to use your best judgement.
    b. Ignore abbreviations.
    c. If any subfield is lacking in one match record, do not consider this to be a non-match.
    d. If field is lacking in match record altogether, do not consider this to be a non-match if all else is equivalent.
    e. Pagination should be close, but doesn't need to be exactly the same- a three page difference is allowed for anything over 10 pages[27]. Pagination in [ ] (square brackets) means it was hand-counted so may not match exactly. Preface pagination may be lacking, but doesn't constitute a non-match if all else is equal.
    f. Ignore pages of plates.
    g. A difference of more than 2cm that is not a result of local binding or trimming should be considered as a non-match. This may be an error, if all else is equivalent. Use your best judgement.
    h. Review MARC field data from the following subfields only: ac.

    Example:

    ```
    =300 \\$axiv, 519 pages, [8] pages of
    plates :$billustrations (some color), maps ;$c25 cm
    ```

    In this example, we're ignoring punctuation and spellings of free text, ie, "pages." We're ignoring pages of plates and $b contents entirely. Preface pages indicated in roman numerals, may or may not be present in matching record and still considered to be a match.

[27] *When to Input a New Record, https://www.oclc.org/bibformats/en/input.html#3xx*