The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0737-8831.htm

Duplicate detection algorithms of bibliographic descriptions

Anestis Sitas
School of Philosophy, Aristotle University of Thessaloniki, and School of Library Science, Technological Institute of Thessaloniki, Thessaloniki, Greece, and
Sarantos Kapidakis
Archive and Library Sciences Department, Ionian University, Paleo Anaktoro, Greece

Received 11 October 2007
Revised 22 October 2007
Accepted 27 January 2008

Abstract

Purpose – The purpose of this paper is to focus on duplicate record detection algorithms used for detection in bibliographic databases.

Design/methodology/approach – Individual algorithms, their application process for duplicate detection and their results are described based on available literature (published articles), information found at various library web sites and follow-up e-mail communications.

Findings – Algorithms are categorized according to their application as a process of a single step or two consecutive steps. The results of deletion, merging, and temporary and virtual consolidation of duplicate records are studied.

Originality/value – The paper presents an overview of the duplication detection algorithms and an up-to-date state of their application in different library systems.

Keywords Cataloguing, Algorithms, Bibliographic systems, Records management

Paper type Research paper

Introduction

The ideal setup for a library catalogue would be to register a unique bibliographic record for each bibliographic entity. However, bibliographic databases include several types of duplicate records. Even if the search cues are clearly specified, locating the correct entry is still an issue that requires further investigation as new materials are added in a variety of media. Duplicate records slow down the indexing process and significantly increase the cost for saving and managing data not to mention that their retrieval is delayed. As a result, duplicate records constitute a system deficiency and compromise quality control for all parties involved, namely users, catalogers, and technical staff. Shared cataloging further aggravates the problem as, through the automated systems, each library-member of one system can access the other members' records. Administrators have to have to improve the bibliographic database quality and keep the database functional and "clean".

Duplicate records

In the environment of bibliographic databases, a duplicate record could be defined as two or more records which stand for or describe the same document (defined as any information resource). Duplicate records can cause problems to the following areas:

Library Hi Tech
Vol. 26 No. 2, 2008
pp. 287-301

0737-8831
DOI 10.1108/07378830810880379

- *User information overload.* Because of the recall of a larger number of documents the user is presented with more information than he or she can actually handle.
- *Reduced system efficiency.* The actual number of records in the database is increased and therefore complicating the efficiency of indexing. This also hinders searching, cataloging decision-making and affects end-user satisfaction.
- *Low cataloging productivity.* Identifying duplicate records and cleaning the database requires valuable time by catalogers, which could be spent on other essential tasks.
- *Increased cost for database maintenance.* More time spent on database maintenance results to an increased cost.

Possible reasons for the existence of duplicate records include novice searchers, the inability for successful searches, and the wish for a "perfect" record to be entered (Wanninger, 1982). Additional factors for record duplication include:

- local practices and policies of cataloging;
- cataloging inconsistencies;
- careless record entering; and
- errors in the syntax of MARC format.

Record matching algorithms

The existence of duplicate records constitutes a problem which is becoming increasingly alarming in networked environments, as the size of individual databases increases and new cooperative networks or consortia are created. In order to reduce the existence of duplicate records, new software is developed using special detection algorithms. Record matching algorithms are programs used to maintain the integrity of bibliographic databases. It would be quite easy to create a process that will match two identical bibliographic descriptions but it is not as easy to match similar records (Hunstad, 1988).

Developing a detection and deduplication process

Designing the process of detection and deduplication of records within a bibliographic database should take the following into consideration:

- *Design goal.* Specifying which types of documents will be represented in the records to be processed (articles, journals, etc.).
- *Specification of duplicate records.* Detailed definition of the term "duplicate record" based on the needs of the particular database.
- *Application of the process.* Specifying whether the process will be applied automatically, semi-automatically, or manually.

Creating a record-matching algorithm

In order to develop an effective algorithm, it is essential to define the application steps, the MARC fields to be used as matching keys, and the criteria for identifying and assessing record similarity/supplication.

Application steps

The algorithm can be applied as a one- or two-step comparison. A final step follows which deals with the management of the duplicate records.

The single step application of the algorithm is, in most cases, a compromise in order to achieve a fast and inexpensive deduplication. In general, these algorithms are more general and with loosely defined criteria resulting in a large number of duplicate records in need of further control.

During the initial step of a two-step algorithm, a file of duplicate records is created based on a limited comparison of fields. Its principal aim is to minimize the number of comparisons during the second step and reduce mismatches that could lead to the deletion of unique records. The second step verifies matches from the first step and then applies a detailed and accurate comparison to determine actual duplicates.

Selection of fields

In order for such an algorithm to be created, it is important to select fields which exhibit significant stability regardless of who created the record (specific cataloger or bibliographic agency). The fields with less stable data offer low probability for record matching (Meir and Lazinger, 1998). Although deduplication based on a control number (ISBN, etc.), is the best method of detection, it does not always ensure full detection. Other data serving as sources for detection include author, title, publisher, pagination, place and the year of publication (Coyle, 1992).

Matching keys

The algorithms for detection of duplicate records use matching keys, which are strings constructed from a pre-selected field or combination of fields. A field can be used as a key in part (e.g. ISBN), or whole (e.g. title proper). Moreover, a combination of fields or a combination of field parts can also be used. Before these keys are created, the data are processed for normalization of spacing, punctuation, special fonts or characters, and capitalization.. In addition, a variety of techniques are used to accommodate for field content differences such as spelling errors, missing data, and small variations of words. These techniques include truncation, keywording, Harrison Keys, Hamming distance, USBC, and others (Toney, 1992).

Matching evaluation

Two methods are used to evaluate the matching of duplicate records:

(1) *Field comparison.* This is based on binary comparisons of selected fields, that is, if fields appear to be the same or not. The software uses YES/NO indications. When the entire field is used, the comparison is safer but the process is time-consuming. This method is very strict and complicates the detection of records that have variations in cataloging or data entry errors (O'Neill and Oskins, 1990).

(2) *Weight assigning.* This method concerns the matching of strings that estimate the similarity assigning weights/values which do not reflect bibliographic significance of the data, but their usage in the recognition of similar records (Coyle, 1992). The matching algorithm allows the merging or deletion of entries only if the assigned weight reaches a pre-determined value, a threshold. This method is open to the existence of minor differences in field content, spelling

errors, completeness or missing data, and variations in cataloging practice (Coyle and Gallaher-Brown, 1985).

Duplicate records handling

Another element in the design of the duplication detection algorithm is the decision of how to handle duplicate records once they are detected. Toney (1992) presented three main practices:

(1) one record is selected as the master record and all others are deleted;

(2) one record is selected as the master record and all non-matching fields from the other records are added to the master (merging); and

(3) all records are kept but clustered around a master record.

Several variations can be added to the above practices. These include: to retain and maintain the record that was entered first in the database and delete the most recent ones; to retain and maintain the most recent record and delete all previous ones; and to retain either the first or the most recent record and merge into it the unique information from all others. Finally, one may choose to merge duplicate records only during the process of searching or retrieval (on the fly). Merging can be made instantly and "virtually" just for the purpose of displaying a single record to the end user.

Results of duplicate detection algorithms

In every effort of duplicate record detection the matching process may bring about the following results:

- *Exact matches.* Records which are absolutely identical.
- *Partial matches.* Only some parts of the records are duplicated.
- *Mismatches, false matches.* Although indicated as duplicates, the records do not represent the same document.
- *Missed/undetected matches.* Existing duplicate records that are not detected by the algorithm.

Mismatches are considered a more important problem than the missed matches, since when deleted there is a permanent loss of information. To avoid this problem, the algorithm should use a loose method so that it gathers records with a degree of variations but avoids possible deletion of bibliographic information. On the other hand, it should be a tight method so that it restricts the accumulation of a large number of possible duplicate records and at the same time it does not allow the loss of genuine duplicate records (Meir and Lazinger, 1998).

Algorithm categorization

Types of material and status

This paper describes ten algorithms. Table I presents these algorithms based on the type of records they are designed to detect. In other words, it specifies whether they refer to the detection of duplicate bibliographic records of monographs, serials journals, journal articles, or other types of material. In addition, the current status of each algorithm is noted. Their status may be defined as:

- *Prototype systems.* Applied in a lab environment.

	Document type				Status	
	Monographs	Journals	Other	Prototype	Inactive	Active
ALEPH-ULM	✓	✓	✓		✓	
ILCSO	✓	✓				✓
Greek Union Catalog	✓		✓			✓
OAK			Articles	✓		
MDBUPD	✓		✓	✓		
IUCS	✓		✓	✓		
OCLC (Hickey and Rypka)	✓			✓		
DDR	✓	✓				✓
COPAC	✓	✓				✓
MELVYL	✓	✓	Articles			✓

Note: ✓ = Yes

Table I. Document type and status

- *Inactive.* While they were once applied in a real environment, their application is now abandoned.
- *Active.* Algorithms that still applied.

The algorithms that will be presented further on concern bibliographic records of monographs, except than the one by Oak Ridge National Laboratory which addressed journal articles. Algorithms for ALEPH-ULM, MDBUPD and IUCS are also applied to other types of documents (microforms, maps, etc.) while the one for MELVYL handles journal articles apart from monographs and journals. Finally, the Union Catalog of Greek Academic Libraries algorithm manages all sorts of materials except journals.

As far as the state of their use is concerned, four out of ten (40 percent) are of the research type (Oak Ridge National Laboratory, MDBUPD, IUCS and Hickey and Rypka). Half of them (50 percent), including ILCSO, DDR, OPAC, MELVYL and Union Catalog of Greek Academic Libraries, continue to be in use even today. One algorithm was applied to the ALEPHs ULM catalog but its application ended in 1998.

Processes of application and evaluation

Apart from the type of materials they are applied to, these algorithms can also be distinguished according to the following characteristics:

- *Application.* This refers to the number of stages of applications as either one- or two−step processing. Three algorithms (30 percent) are applied in one step (ALEPH-ULM, ILCSO, and Union Catalog of Greek Academic Libraries). The remaining seven algorithms (70 percent) follow the practice of two-step process.
- *Evaluation.* This refers to the methods of comparison used to assess whether two or more bibliographic records are identical. These methods include either a comparison between fields or the assignment of weights. Of the algorithms presented in this paper, 40 percent use the method of the field comparison (ALEPH-ULM, MDBUPD, IUCS, and Union Catalog of Greek Academic Libraries). The remaining 60 percent, assign points/values for weights.

Table II presents each algorithm and their respective application method, whether the application is done during the process of searching or retrieval (on the fly), and their evaluation method.

Final handling and algorithm running

Furthermore, we can distinguish algorithms according to the final handling of the detected duplicate records (deletion or merging), as well as whether this process is done online or offline. Final handling information for each algorithm is presented in Table III.

Final handling

This refers to the final stage of the process of detecting duplicate records. Three programs (ILCSO, MDBUPD, IUCS), 30 percent, delete duplicate records. The MDBUPD and IUCS algorithms end up deleting the spare ones and retaining just one record, while ILCSO selects and retains the most suitable one. In total, five of them, 50 percent, including ALEPH-ULM, Union Catalog of Greek Academic Libraries, DDR, COPAC, and MELVYL, merge duplicate records in one integral record. COPAC merges

	Application		Evaluation	
	Steps	On the fly	Field comparison	Weights
ALEPH-ULM	1		✓	
ILCSO	1			✓
Greek Union Catalog	1		✓	
OAK	2			✓
MDBUPD	2		✓	
IUCS	2		✓	
OCLC (Hickey and Rypka)	2			✓
DDR	2			✓
COPAC	2	✓		✓
MELVYL	2	✓		✓

Table II.
Algorithm application and evaluation methods

Note: ✓ = Yes

	Final handling		Algorithm running	
	Deletion	Merging	Offline	Online
ALEPH-ULM		✓	✓	
ILCSO	✓		✓	
Greek Union Catalog		✓	✓	
OAK	*	*	✓	
MDBUPD	✓		✓	
IUCS	✓		✓	
OCLC (Hickey and Rypka)	*	*	✓	✓
DDR		✓	✓	✓
COPAC		✓	✓	✓
MELVYL		✓	✓	

Table III.
Final handling and time of algorithm running

Notes: ✓ = Yes; * = Not available

the records in two of its three segments (the first segment includes only the British Library records and each one of the other two segments include approximately the 50 percent of the other catalog records). Among the three segments, however, there is no physical merging but it makes possible to present merged records to users in real time during the search. MELVYL's practice does not lead to the physical merging of duplicate records, but to online presentation of merged records during the recall phase. For two out of ten algorithms (Oak Ridge National Laboratory, Hickey and Rypka), 20 percent, there is no information available.

Application time

This refers either to the offline or the online process. All algorithms "run" offline. Only three of them (30 percent) have the ability to apply online procedures as well. The Hickey and Rypka algorithm was designed to run both ways, DDR was designed to be applied both ways as well but the offline procedure is preferred. Finally, in COPAC part of the procedure is applied offline and part of it is applied online. The term "online" is used to refer to the real time running.

Fields used for the creation of keys (monographs)

Another significant characteristic of the algorithms are the MARC fields used for the creation of comparison keys. As we can see in Figure 1 the majority of algorithms (nine out of ten, 90 percent) use author, title and publication year for key creation. In addition, the algorithms also use the following fields in key creation: 70 percent of

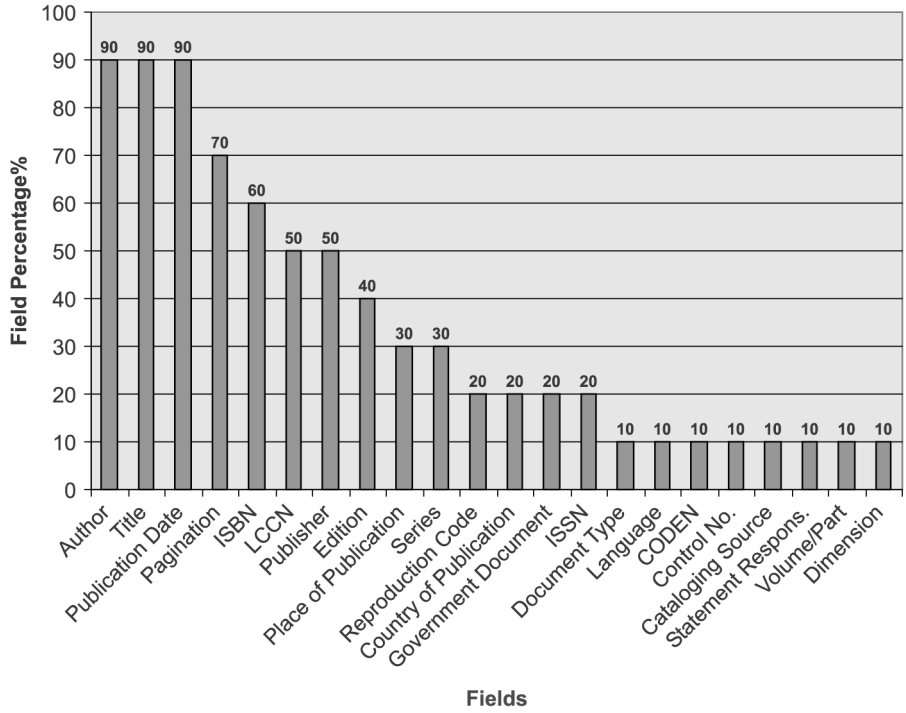

Figure 1.
MARC field use for key creation (monographs)

algorithms use pagination, 60 percent use ISBN, 50 percent use LCCN and/or publisher, 40 percent use edition statement, 30 percent use place of publication and/or series, 20 percent use fields like reproduction code, country of publication, government document number and ISSN, and finally 10 percent use fields such document type, language of the document, CODEN, control number, cataloging source, statement of responsibility, and volume/part and dimensions.

Table IV presents detailed information on all fields that are used for duplicate bibliographic record detection.

Algorithm efficiency

Most organizations that apply duplicate detection procedures have not publicized their algorithm efficiency results. Even the data at hand are not absolutely comparable since each case is distinct and because the results of application depend on:

- the type/types of documents;
- the given definition of "duplicate record";
- the consistency of cataloging and data entry; and
- the target set by each algorithm.

From the data presented in Table V we draw the following:

- efficiency among algorithm applications range between 44.95 percent and 99.62 percent out of the total identified duplicate records, real duplicate records were only the previously referred percentage;
- mismatches range at a percentage below 1.5 percent; and
- missed matches range somewhere around 4 percent with the exception of those presented in ALEPH, which range from 17.4 to 34 percent.

Following is an analysis of each individual algorithm.

One step algorithms

ALEPH-ULM

ALEPH is the network of the research libraries of Israel, which maintained the Union List of Monographs. The entries were loaded with the use of their detection and merging algorithm. It was based on the comparison of a stable number of not frequently met letters that came from four fields: author (five characters), title proper (seven characters), publication date and language (Lazinger, 1994).

In a 1996 research study examined the efficiency of the algorithm when applied to monographs. It was reported that it yielded 0 percent mismatches for records describing Hebrew materials and 1.4 percent for English but it failed to detect existing duplicates in for 17.4 percent of English and 34 percent of Hebrew records (Meir and Lazinger, 1998). ULM, now named Union List of Israel, decided that their algorithm did not satisfy their demands and in 1998 stopped all deduplication efforts.

Illinois Library Computer Systems Organization

For duplicate record detection, the system uses indices of the following control numbers: OCLC, LCCN, ISBN, ISSN, and publisher number. When the data of these indices overlap, they are given specific values. Then, further actions, based on the sum

Fields	ALEPH-ULM	ILCSO	Greek Union	OAK	MDBUPD	IUCS	Hickey and Rypka	DDR	COPAC	MELVYL
Document type							✓			
Reproduction code							✓	✓		
Country of publication								✓		✓
Language	✓									
LCCN		✓			✓		✓	✓		✓
ISBN		✓	✓				✓	✓	✓	✓
ISSN		✓							✓	
CODEN				✓						
Control number		✓								
Cataloging source								✓		
Government document							✓	✓		
Author	✓		✓	✓	✓	✓	✓	✓	✓	✓
Title	✓		✓	✓	✓	✓	✓	✓	✓	✓
Statement of responsibility								✓		
Volume/part				✓						
Edition			✓				✓	✓	✓	
Place of publication					✓		✓	✓		
Publisher					✓		✓	✓	✓	✓
Publication date	✓		✓	✓	✓	✓	✓	✓	✓	✓
Pagination				✓	✓	✓	✓	✓	✓	✓
Dimension								✓		
Series							✓	✓	✓	

Note: ✓ = Yes

Table IV.
Fields used in the creation of keys (monographs)

of the weights, are determined. This is an offline process. The following are the values recommended for the bulk import of records (ILCSO, 2004) (see Table VI).

Once the comparison is done and the matching shows that two bibliographic records represent the same document, they are evaluated so that the most suitable is selected to remain in the database while the other will be deleted. For each field used for the matching process, there is a corresponding field weight to help decide which record will remain. The fields used for matching include: cataloging source, encoding level, agency that has modified the original record and bibliographic level of the record. In dubious cases the final decision is taken by comparing the records manually (ILCSO, 2004).

Union Catalog of Greek Academic Libraries

Use of this algorithm started in April of 2005. At the time of import, records are checked for duplicate detection and merging. Imported records are created in a variety of software and therefore records have differences in format, the number of letters, the holdings of existing records, etc. After loading these records are processed so there are no such variations. To accommodate this, the key is formed by taking data from the fields further down: Title, Author, Edition statement, Publication date and ISBN (Vougiouklis, 2007). Questionable duplicate records are kept in a work to be examined manually.

Based on the algorithm evaluation, it was estimated that 44.95 percent of actual duplicate records were detected. Among the detected problems, 17.8 percent were mainly due to the applied key, while 12.47 percent were due to the policy issues, 7.05 percent represented cataloging problems, and 17.62 percent referred to other kinds of problems.

	Effectiveness %	Mismatches %	Missed matches %
ALEPH-ULM	*	0-1.5	17.4-34
Greek Union Catalog	44.95	*	*
IUCS	56.58-99.62	0.54	*
OCLC (Hickey and Rypka	54-69	1.3	*

Table V.
Algorithm efficiency

Note: * Not available

Duplicate replace	=	100
Duplicate warn	=	30
Indexes and weights		
035O	=	100
010A	=	20
020A	=	25
022A	=	15
028A	=	10

Table VI.
Recommended values for bulk import of records

Two step algorithms

Oak Ridge National Laboratory

In 1976, Oak Ridge National Laboratory created an algorithm aiming at detecting duplicate records of cited journals articles. It was used offline and it produced fixed length keys (Hickey and Rypka, 1979). Publication date, initial page number, journal CODEN, volume number, and samplings from the author, journal title, and article title elements were used for record matching.

For duplicate record detection the keys were sorted in many and various fields. When fields matched perfectly, a weighted matching of the remaining fields was used. The algorithm was completed with a page/year and author/title sorting.

Online Computer Library Center (OCLC): MDBUPD

This program was created by OCLC shortly after 1976; it was named Master Data Base Update (MDBUPD) and was used offline. This algorithm was designed as a two-step application (Wanninger, 1982).

Initially, it searched the database using LCCN and keys produced by OCLC. These keys were derived from the name/title fields or just from the title field.

Then, it checked additional fields for verification. These were: Publisher, Place of publication, Title, Date of publication, Pagination. Towards the end of this process, after the absolute matching of all compared fields, duplicate records were deleted.

University of Illinois: IUCS

IUCS (IRRL [Information and Retrieval Research Laboratory] Union Catalog System), was developed to detect non-monographic documents as well as maps, filmstrips, etc. (Williams and MacLaury, 1979). Once the data were normalized, they were processed by comparing fields and applied in two steps/passes.

The first step involved the creation of a matching key. The "title-year" keys were sorted and the keys of the documents that were identical were later recalled and compared in the second step (Hickey and Rypka, 1979).

For the second step, a number of detailed matching processes were applied so that the first estimation was either verified or rejected. A title mapping key different from that of the first step was used. The author names, titles and pagination of records that were recalled in the previous step as possibly duplicate ones were compared and it was then specified, which were ultimately duplicate ones.

The efficiency of this algorithm ranged from 56.58 to 99.62 percent depending on the database which was being tried. Mismatches accounted for 0.54 percent of the total number of duplicate records (Cousins, 1998). When it was not possible to reject or accept records as duplicates, a non-automated comparison of records was used (Hickey and Rypka, 1979).

Online Computer Library Center (OCLC) – Hickey and Rypka

During 1978-1979 OCLC tried once again to develop a research program for detecting duplicate monographs. This algorithm was developed by Hickey and Rypka and could be applied both online and offline. It was applied in two steps/sections (Hickey and Rypka, 1979):

(1) The first step or exact-match section aimed at clustering of related keys in order to reduce the number of full key comparisons.

(2) In the second step all other keys of selected fields that matched in part or whole were applied.

These Keys were derived from the following fields of bibliographic record: Reproduction code, Record type, Title (only the beginning), Publication date, Place of publication, Author, Pages, Publisher and Hashed title. SuDoc number, ISBN, Edition statement, Series, and LCCN were incorporated only *f* present in bibliographic records. This algorithm was checked against a decision table to determine if the keys were duplicates. This table specified 16 alternative ways by which two keys could be matched. The comparison of the two keys could yield a quote which took any one of the three values: $-$ = mismatch, P = partial match, E = exact match.

It was found that mismatches were 1.3 percent of the total records identified as duplicates (Hickey and Rypka, 1979). The algorithm located approximately 54-69 percent of duplicate records depending on whether reprints were defined as duplicates or not.

Online Computer Library Center (OCLC): DDR

In 1990, OCLC created a new algorithm for duplicate record detection. It is applied to monographs and journals and consists of two steps.

In the first step, with the application of the clustering algorithm possible duplicate records are clustered with the use of a key consisting of eight characters after the data have been normalized. Only records with the same key titles using seven more elements are included. These elements include LCCN, ISBN, Publication date, Pages, Author, Publisher, and Full title. Records with the same key titles and identical LCCN or ISBN, either identical at least two out of the other five elements are considered as possibly duplicates (O'Neill and Oskins, 1990).

In the second step, the evaluation algorithm is applied. This estimates the similarity between possible duplicate records. The similarity values range from "0.0" for not identical ones to "1.0" for the absolutely identical records (O'Neill and Oskins, 1990). The elements are considered partial matches if their similarity is greater than 0.85 percent. When no automated decision is possible, the records are identified for non-automated control.

Research showed that the recall of clustering is 96 percent and that 56 percent of the total duplicate records can eventually be detected (O'Neill and Oskins, 1990). This algorithm led to the creation of the DDR software which is used to specify and merge duplicate records representing books and periodicals. Although it can run offline, OCLC has chosen to apply it as an offline procedure.

Consortium of University Research Libraries (CURL): COPAC

COPAC, the union catalog of the members of CURL, has been in use since 1996. The process of duplicate record detection follows two distinct practices. The first practice deals with the process of detection of duplicate records that is applied only in one part of the database (the second practice is descried in the "Detection and merging on the fly" chapter). The process takes place offline with the aim of merging duplicate records. It is applied in two steps/stages.

Step one: each imported record is compared to very record in the database. To achieve this, two methods are used (Cousins, 1998):

(1) Matching ISBN/ISSN: clusters of matching records are located based on ISBN. After the text is normalized, matching fields are assigned weights/values. In the end, the values of all fields are added up. If the total assigned weight is equal or bigger than 13, the record is identified for merging. If the record has an edition statement, matching of this field is also necessary. In the same way, checks for series volumes and multi-volume works take place (Cousins, 1998).

(2) Matching of author/title acronym: records without ISBN or ISSN and records with ISBN/ISSN which fail to find a similar record are re-examined with the use of an acronym author/title, 4/4 letters of author and publication year. Possible matches are promoted to the next step. At this point no weighting is determined and for each field matching is a simple YES/NO. Matching based on acronyms introduces the matching of two new fields: publisher and total number of pages (Cousins, 1998).

Step two: In order to verify possible matching records, a number of detailed matchings take place. The fields used in this process are: ISBN, ISSN, Publication date, Title, Author, Edition statement, Series, Pagination, and Publisher.

COPAC still continues to apply the process described above, but part of its process is done during the process of searching by end users. This part of the process is presented in the following section.

Detection and merging on the fly

All processes of algorithm applications for duplicate record detection aim primarily at the deduplication or merging of duplicate records. Another practice is the application of the program on the fly. Detection and merging of duplicate records is done during the search or retrieval of records and does not lead to their physical merging, but just to a temporary or "virtual" merging for reasons of presentation to end users. Two programs that apply this method are described below.

COPAC: detection and merging of records upon search

The majority of the duplicate record detection and merging process continues to take place offline as described previously. A process of three sets of data loading is applied which leads to the creation of three segments in the database (Cousins, 2006):

- One set is the data from the British library. These records do not consolidate.
- The other two data sets, each consisting of records from approximately half of the other COPAC libraries, have their records consolidated into a specific segment during the data loading using the process described earlier.

There is no record consolidation between the three segments, which leads to the existence of duplicate records between them. To compensate for this problem, a check for duplicate records is performed as an on the fly process. When a user searches, results are checked for any possible duplicate records before they are displayed to the user. When duplicate records are found, they are displayed to the user as just one record. This record includes all information from the other records that are included in the result set. This matching and consolidation process during loading time, combined with the process of matching during the search process, is a substantial compromise

compared to actual detection and merging of duplicate records with large amounts of data.

MELVYL: detection and merging of entries upon retrieval

The network of the University of California libraries supports the entire system, which runs duplicate record merging on the fly. The records are not merged physically but they are merged and presented dynamically during the search process. Apart from book and journal records, the monographs algorithm is applied to in-analytics, as well as to other non-print materials. This algorithm is applied when each new record is loaded to the MELVYL database, which is basically an offline process. Every time a new record is loaded, its possible identical records are located and the new result is saved in an Oracle table. If a record matches a user's search criteria, the system automatically checks this table and the best record is recalled (Campbell, 2006).

A two-step process is followed for the advancement of identical records to the final phase of merging. Initially, a pool of possible duplicate records is created. In the first step, there is a comparison of LCCN/ISBN, publication year, and the first twenty-five characters of the title. At this point a threshold weight is assigned. The threshold for merging monograph records is 875 points. If during the first step of comparisons identification for merging is not achieved, a second step of comparisons is performed based on data from the title, main entry (normalized), country of publication, pagination, and publisher.

Conclusion

This paper examined the algorithms applied to eliminate the problems caused by the existence of duplicate bibliographic records in a database. When algorithms are applied in one step, a faster application is achieved but the percentage of database cleanup usually remains low. Most algorithms are two step applications., These result in a greater database quality improvement, since with the initial application of a short key, all possible duplicate records are collected and therefore file the rest of the algorithm is applied only to this new file. The methods used for duplicate matching evaluation are field comparisons and weight assignment. Almost all algorithms studied so far run offline. Also presented is the application of another approach which facilitates a temporary consolidation as a user carries out a search or during the recall stage (on the fly process). The result of this method is not the physical merging of duplicate records in the database but their temporary or "virtual" consolidation for the purpose of presentation to the user.

For the creation and selection of the appropriate duplicate records handling algorithms, there is neither an absolute and specific solution, nor a system or a tool which can be simply transferred and applied purely from one environment to the other. Each environment has its own specifications and policies; it applies specific practices and has specific and special needs. In every system the application of these algorithms calls for a special study and modifications to correspond to the given needs.

The focus of future research is the handling of large scale data in a network environment and in real time. Virtual catalogs and Z39.50 protocol are the focus of future study. Users wish for a comprehensive, updated, clear, consistent, and fast catalog, which is capable of incorporating searches between distributed databases in a heterogeneous network with consistency, accuracy and speed. Further research on

conventional ways of duplicate record management including the most current practices such as virtual merging is needed. This research is important in order to fully understand a problem to which no satisfactory solutions have been found while at the same time the needs for such solutions are constantly increasing.

References

Campbell, C. (2006), Melvyl Project Coordinator, information given by e-mail, (accessed 31 January 2006).

Cousins, S.A. (1998), "Duplicate detection and record consolidation in large bibliographic databases: the COPAC database experience", *Journal of Information Science*, Vol. 24 No. 4, pp. 231-40.

Cousins, S. (2006), *COPAC Service*, Manchester Computing, University of Manchester, available at: copac@mcc.ac.uk (accessed 11 January 2006).

Coyle, K. and Gallaher-Brown, L. (1985), "Record matching: an expert algorithm", *ASIS Proceedings*, Vol. 4 No. 1, pp. 77-80.

Coyle, K. (1992), *Rules for merging MELVYL® Records*, Technical Report No. 6, University of California, DLA, Oakland, CA.

Hickey, T.B. and Rypka, D.J. (1979), "Automatic detection of duplicate monographic records", *Journal of Library Automation*, Vol. 2 No. 12, pp. 125-42.

Hunstad, S. (1988), "Norwegian bibliographic databases and the problem of duplicate records", *Cataloguing and Classification Quarterly*, Vol. 8 Nos 3/4, pp. 239-48.

ILCSO (2004), *Using OCLC for ILLINET Online/Voyager Data Entry*, Illinois Library Computer Systems Office, available at: http://office.ilcso.illinois.edu/Docs/using_OCLC.pdf (accessed 15 February 2007).

Lazinger, S.S. (1994), "To merge and not to merge – Israel's Union List of Monographs in the context of merging algorithms", *Information Technology and Libraries*, Vol. 13 No. 3, pp. 213-9.

Meir, D.D. and Lazinger, S.S. (1998), "Measuring the performance of a merging algorithm: mismatches, missed-matches, and overlap in Israel's Union List", *Information Technology and Libraries*, Vol. 17 No. 3, pp. 116-23.

O'Neill, E. and Oskins, W.M. (1990), *Duplicate Records in the Online Union Catalog*, OCLC Office of Research, Dublin, OH.

Toney, S.R. (1992), "Cleanup and deduplication of an international bibliographic database", *Information Technologies and Libraries*, Vol. 11 No. 1, pp. 19-28.

Vougiouklis, G. (2007), ELiDOC, available at: gvoug@elidoc.gr (accessed 2 February 2006).

Wanninger, P.D. (1982), "Is the OCLC database too large? A study of the effects of duplicate records in the OCLC system", *Library Resources and Technical Services*, Vol. 26, pp. 353-61.

Williams, M.E. and MacLaury, K.D. (1979), "Automatic merging of monographic data bases: identification of duplicate records in multiple files: the IUCS Scheme", *Journal of Library Automation*, Vol. 12 No. 2, pp. 156-68.

Corresponding author
Anestis Sitas can be contacted at: sitas@lit.auth.gr