Simplifying the Reversed Duplicate Removal procedure

Tove Faber Frandsen

Letter to the editor

Simplifying the Reversed Duplicate Removal procedure*

Tove Faber Frandsen

Royal School of Library and Information Science, Birketinget 6,
DK 2300 Copenhagen S, Denmark, E-mail: kk02tofa@db.dk

In 1995 a method called Reversed Duplicate Removal (RDR) was introduced in Christensen and Ingwersen (1995), and it was further enhanced in Ingwersen and Christensen (1997). The RDR-method enables performing analyses of the overlap consisting of identical documents held in different files when performing online isolation and analyses of datasets.

In cross-files searching removal of duplicates is crucial for any subsequent analysis not to mention that the duplicates themselves can be of great interest as pointed out by Pao (1994). The duplicates are removed by an algorithm provided by Dialog called *remove duplicates* (RD) which removes records with identical author and title. When removing the duplicates the file order is important as the records retained is determined by the file order. The duplicate removal in a reversed order is the foundation of the principle of RDR. The procedure isolates the overlap between files. When done in reversed order the overlap can be held in either of the files in cross-files searching.

The procedure of RDR by Ingwersen and Christensen can be illustrated by using elementary set operations. Furthermore, it can be simplified such that fewer commands need to be executed in order to isolate the overlaps. The illustration shown here is done for only two sets of documents but the point extents to any finite number of sets.

Consider records from files A and B, which consist of records only in file A (A\B), records only in file B (B\A) and overlap records found in both file A ({ a_A, b_A, c_A, …, n_A }) and file B ({ a_B, b_B, c_B, …, n_B }). When performing Dialog's *set*-command a union of the two sets including the duplicates from *both* A and B is formed:

Set A,B = *set B,A* = A\B ∪ B\A ∪ { a_A, b_A, c_A, …, n_A } ∪ { a_B, b_B, c_B, …, n_B }

However, as mentioned above the file order determines which records that are retained when using the duplicate removal algorithm. When the files are ordered using the "*set A,B*"-ordering the records retained are C_A = A\B ∪ B\A ∪ { a_A, b_A, c_A, …, n_A }, i.e. all records from file A (including the overlap) together with the records from file B which are not part of the overlap. Similarly, when the files are ordered with "*set B,A*" the records retained are C_B = A\B ∪ B\A ∪ { a_B, b_B, c_B, …, n_B }.

According to the Ingwersen and Christensen-procedure, isolation of the overlaps proceeds by using Dialog's *from*-command on the sets C_A and C_B, which is the same

* I thank Ellen Bonnevie and Rasmus Højbjerg Jacobsen for valuable comments and suggestions.

as finding the intersection of the sets. When isolating the overlap from file A, records from this file are extracted from both C_A and C_B. Hence,

C_A *from A* = (A\B ∪ B\A ∪ { a_A, b_A, c_A, …, n_A }) ∩ A = A\B ∪ { a_A, b_A, c_A, …, n_A } = D_A

C_B *from A* = (A\B ∪ B\A ∪ { a_B, b_B, c_B, …, n_B }) ∩ A = A\B = D_B

The overlap from file A is finally found by using DIALOG's *not*-command, which is the same as set subtraction:

D_A *not* D_B = (A\B ∪ { a_A, b_A, c_A, …, n_A }) \ (A\B) **= { a_A, b_A, c_A, …, n_A }**.

A similar procedure should be followed in order to find the overlap from file B. Thus a total number of 9 Dialog-commands must be executed before the overlaps from both files have been isolated.

When inspecting the above operations it is clear, however, that the same result can be achieved using only 5 commands. Instead of using the *from*-command and finding the intersections of C_A and C_B with the original files A and B, subtracting C_B directly from C_A will yield the overlap from file A:

C_A *not* C_B = (A\B ∪ B\A ∪ { a_A, b_A, c_A, …, n_A }) \ (A\B ∪ B\A ∪ { a_B, b_B, c_B, …, n_B })
= { a_A, b_A, c_A, …, n_A }

Similarly, subtracting C_A from C_B yields the overlap from file B:

C_B *not* C_A = (A\B ∪ B\A ∪ { a_B, b_B, c_B, …, n_B }) \ (A\B ∪ B\A ∪ { a_A, b_A, c_A, …, n_A })
= { a_B, b_B, c_B, …, n_B }.

This letter has shown that the Reversed Duplicate Removal-procedure suggested by Ingwersen and Christensen (1995) can be simplified. Only 5 commands need to be executed in order to isolate the overlaps from 2 files compared to 9 commands in the original procedure. The key point in the argument was that when using the duplicate removal algorithm in Dialog, the duplicates from each of the two sets are retained in the search. By comparing the two sets, only one set of duplicates remains.

References

Christensen, F. H.; Ingwersen, P. (1995). Fundamental methodological Issues of data set creation for the analysis of research publications. In: Fifth International Conference of the International Society for Scientometrics and Informetrics : proceedings.

Ingwersen, P. og Christensen, F. H. (1997). Data Set Isolation for Bibliometric Online Analysis of Research Publications: Fundamental Methodological Issues. In: Journal of the American Society for Information Science, 1, 205-217.

Pao, M. L. (1994).Relevance odds of retrieval overlaps from seven search fields. In: Information Processing & Management, 30(3), 305-314.