Detecting Approximately Duplicate Bibliographic Records with Text Algorithms: Experience of Creating a Union Catalogue of Libraries at the Warsaw University of Technology

Grzegorz Płoszajski

Warsaw University of Technology, Main Library, Faculty of Electronics and Information Technology, Politechniki 1, 00-661 Warsaw, Poland, g.ploszajski@ia.pw.edu.pl

(Received 5 November 2002; revised manuscript received 24 March 2003)

Abstract: The paper describes a fault-tolerant method of selecting duplicate bibliographic records in catalogues. The method is based on the use of text algorithms; decisions are suggested to librarians who make the final decision. The method was applied to four library catalogues at the Warsaw University of Technology which were compared with the catalogue of the main library. Process of joining catalogues was conducted differently for non-duplicate records and for duplicate ones. Thanks to this method, a significant portion of records in the catalogues of the joining libraries had been found to be duplicate before the catalogues were added. The algorithms proved helpful in assuring high quality of information.

Keywords: duplicate record resolution, *n*-grams, text algorithms

1. Introduction

Information in library catalogues should be unique: all identical books should be referred to by one and only one bibliographic record. A union catalogue collects information from catalogues of a group of libraries. To avoid duplicate bibliographic records in a union catalogue such records should be found before or during the process of the joining catalogues of the participating libraries [1, 2].

Selecting duplicate bibliographic records which are identical is an easy task. There is, however, a less straightforward problem of finding duplicate records which actually refer to the same books, but are not identical. The differences between such approximately duplicate records occur due to errors in typing as well as due to habits and experience of individual librarians. In most cases, these differences are small and records are "similar" in some way.

The task described herein was to compare catalogues and find duplicate bibliographic records, either identical or having minor differences. It is important especially in the case of union catalogues [1, 2], but also in the case of multi-database searching [3], where elimination of duplicate records significantly improves data quality. This task can be facilitated by means and procedures known as "duplicate record resolution".

2. The method of comparing bibliographic records: the concept

The comparison of bibliographic records is based on the comparison of corresponding fields and subfields. In the present project the scope of comparison of records was limited to the following elements of bibliographic information:

- ISBN – field 020 (10 characters without hyphens),
- code of language – field 041,
- author – field 100,
- title, subtitle, coauthors, part – field 245: subfields a, b, c, n,
- edition – field 250,
- place and year of publishing – field 260: subfields a, c,
- series – field 440.

Publishers (field 260b) were not included because, according to librarians, there were too many differences and errors; entries in this subfield were compared and corrected separately.

The comparison of bibliographic records could not have been limited to ISBN (International Standard Book Number) because many records had no ISBN at all. Moreover, the possibility of errors in ISBN was assumed. Nevertheless, ISBN was treated as an important point of comparison and had a large weighting coefficient.

A bibliographic record may have multiple fields and subfields. In such a case, the process of records comparison is more complex. It was assumed in our exercise that, in the case of multiple subfields 260a, only two (the first and the second) are considered. In the case of subfields 245c, considered are: the second author, the third author and the editor. Due to these assumptions, the above mentioned set of fields and subfields can be treated as an entity in a relation, *i.e.* as columns of a table in relational database. One row in such a table corresponds to one bibliographic record in a catalogue. One table was created for each library.

A pair of records in two tables was defined as "similar" (*i.e.* the corresponding bibliographic records were assumed to be approximately duplicate), if:

1. some fields are identical, or
2. some fields have minor differences (*e.g.* lack one or a few characters, have mistyped characters, have a changed sequence of characters in a word), or
3. there are no fields having big differences, or
4. in case of big differences between titles (245a) and/or subtitles (245b) and/or series (440), the condition of minor differences was first checked for a concatenation of "title & subtitle" and series and secondly for a concatenation of "title & subtitle & series".

The fourth condition was added to treat a common type of record discrepancy resultant from cataloguing by individual librarians.

It is worth noting that even in cases when all fields of two records in the compared tables are identical, the corresponding bibliographic records are often only "similar", but can in fact differ, and thus need to be compared.

Information contained in the above mentioned fields was preprocessed and stored in additional fields. In the case of subfields 245n and 260c and field 250, only numbers were selected for comparison and were written as numerals (if field 250 was empty, the number was assumed to be 1). Text information from these subfields (*e.g.* "reprint") was not taken into consideration.

In the case of field 100 and subfield 245c, the names of authors and/or editors were abbreviated (except for the last name) and written in additional fields. At the same time, the original form was checked as to whether the names are written in an abbreviated form or in full. If, in two corresponding fields, the authors were written in a non-abbreviated form, then this form was used for comparison, otherwise the abbreviated form was used.

The comparison of records was made by a group of algorithms applied to particular fields. Differences between corresponding fields were expressed numerically, and their weighted sum was treated as a measure of difference (discrepancy) between records; the bigger difference the greater the number. The sum of these numbers gave a value of "dissimilarity" or "distance". Pairs of records with low "distance" were treated as "similar", *i.e.* as approximate duplicates.

If the difference between given fields was great enough, such a pair of records was treated as "dissimilar" (there was no need to compare other fields).

The numerical fields were compared by means of simple algorithms:

1. subfields 245n were checked only for identity (0 if identical; 1 if not),
2. the number of the edition and the year of publication were cross-checked:
   - the absolute value of the difference between edition numbers was multiplied by a small weighting coefficient in the case of identical years, and by a much greater coefficient in the case of different years,
   - the absolute value of the difference between years was multiplied by a small weighting coefficient in the case of identical edition numbers, and by a much greater one in the case of different edition numbers,
   - in the case of a smaller edition number and a greater year (or vice versa) the librarians are alerted to a possible error.

In the case of all other fields, text algorithms were applied to measure "distance" between texts.

Text algorithm is a general name for an algorithm dealing with text data. A typical example of such an algorithm is a spellchecker – a program used in many word processors. Spellcheckers usually compare an edited text with a dictionary. This is not useful in the case of bibliographic records which often contain words from a number of languages, possibly in bibliographic transliteration.

There are text algorithms oriented towards searching for instances of a given string in a text (pattern matching), or the longest common substring [4, 5]. Another group of algorithms is based on the use of the so-called n-grams [6], *i.e.* sequences of n characters. To detect whether a given n-gram appears in a given string of characters (and, if so, how many times), a pattern matching algorithm can be used.

3. Algorithms

A number of text algorithms have been tested elsewhere with respect to the effectiveness of comparing records [6]. In the present project, two types of universal algorithms have been chosen to compare most of the text fields, and a heuristic algorithm has been developed to compare ISBN numbers.

The first type of text algorithm is based on the comparison of the numbers of individual

characters in each of the two texts (strings) compared. It has been assumed that the strings are transformed to lower case before such comparison. The set of characters has been limited to 26 Latin letters, 9 Polish diacritics, 10 digits, space, point and hyphen (minus sign). All other characters have been treated as "other". In total, 49 characters have been considered.

For each character, the number of instances in the text is counted, 49 such numbers constitute a "profile". A comparison of two texts is based on a comparison of their profiles. The measure of "distance" between pairs of texts is defined as the sum of absolute values of differences between 49 pairs of corresponding numbers. In the case when one character in one of two texts being compared is missing, the value for such a distance is 1; in the case of a mistyped character in one record, it is 2. However, in the case of exchanged positions of two characters the "distance" is 0. This kind of error can be "noticed" by the next algorithm.

The second type of algorithm is based on the comparison of the numbers of n-grams, *i.e.* sequences of n characters, present in the texts compared. To detect whether a given n-gram is present in a given string of characters (and, if so, how many times), pattern-matching algorithms are applied. There exist so many possible n-grams that the comparison of texts is not made with respect to a fixed set of n-grams (as it has been in the case of the 49 characters), but with a dynamic set created during the process of comparing a given pair of texts. In an m-character text, there is $m-1$ digrams (2-grams), $m-2$ trigrams *etc.* Each of $m-1$ subsequent digrams from one text (usually the shorter one) is searched for in the second text. The number of digrams not found in the other text is a measure of "distance" between the texts. A similar measure is used for trigrams.

The process of comparison of all text fields (except for 020 and 041) is organized as follows:

1. First, the length of the texts is compared. If the difference between in length is greater than a chosen threshold value $t1$, the difference between the texts is stated as "great". Then there is no need then to go to steps two and three of this algorithm.
2. For each of the 49 characters, the number of instances in both texts is calculated and the absolute values of differences are totalled. If the sum total is greater than a chosen threshold value $t2$, the difference between the texts is stated as "great", and there is no need to go to step three of the algorithm.
3. As an introductory step, one space is added at the beginning and one at the end of each of the two texts. Subsequent digrams are taken from the shorter text (text one); for each such digram, its occurrence in the other text is checked by a pattern-matching algorithm. The number of digrams not found in the second text is totalled; if the total is greater than a chosen threshold value $t3$, the difference between the texts is stated as "great". If the difference is not "great", a numerical measure of the distance between the texts is calculated by adding the sum total from step two and the number of digrams not found in the other text in the present step of the algorithm.

Threshold values for all the three steps depend on text length and the type of information. For author names, the threshold values are small, while they are greater for the titles. The following formulas have been used, where li stands for the length of i-text (it was assumed that $l1 \leq l2$) and E stands for the function equal to the integer part of its argument:

for the names of authors:

$$t1 = 5,$$
$$t2 = 1 + (l2 - l1) + E(l1/10),$$
$$t3 = 3 + E(l1/10);$$

for titles, subtitles and series:

$$t1 = 5 + E(l1/15),$$
$$t2 = 3 + (l2 - l1) + E(l1/15),$$
$$t3 = 6 + E(l1/15).$$

Publishers (260b) were compared by means of an algorithm similar to the one used for comparing titles, having some additional rules to deal with differently abbreviated terms.

A heuristic algorithm has been developed to compare ISBN numbers.

1. All corresponding digits (characters) are compared sequentially, *i.e.* the first with the first, the second with the second *etc.*
2. If there are no more than two differences, the measure of the distance is calculated as follows:
   - in the case of a changed sequence of two following digit, a relatively small value of distance is given (*e.g.* 0.3);

- in the case of the following pairs of digits: 3–8, 1–7 and 6–9 on corresponding positions, the difference is small, while in all other cases it is standard (*e.g.* 0.3 and 1);
- these differences are multiplied by a large weighting coefficient when the difference occurrs in the initial position of the ISBN number and by a medium coefficient when in the second position.

3. If there are more than two differences, two cases are checked:
   - part of the digits is "moved" in cyclic way (the "distance" is proportional to the length of the cycle);
   - one digit is missing, others are moved forward and some other digit or character is added at the end (the "distance" is greater than in the case of a cycle).
4. In other cases, the difference between ISBN numbers is treated as "great".

ISBN numbers are compared only when two records have this field non-empty.

The code of language is checked for identity in the case of a single code (three characters). In the case of multilingual codes, the codes are transformed into separate single codes and compared as sets of codes (the sequence of codes is not important, only the presence of a given language).

4. Statistics

The catalogue of the Main Library had 49 000 of records. The catalogues of the other libraries had 14 000 of records in total, and were added sequentially. Each of them was compared to the Main Library (ML) catalogue, and the second of them was also compared to the first, the third – to first and second *etc.* For each library, the following was specified:

- the number of bibliographic records in its catalogue,
- the number of duplicated records found, and
- within this number, the number of duplicated records with non-zero measure of distance (MD).

Lib. 1. 7 334 rec. – 2 472 dupl. rec. to ML (727 with MD > 0).

Lib. 2. 1 669 rec. – 470 dupl. rec. to ML (164 with MD > 0) and 33 to Lib. 1.

Lib. 3. 3 062 rec. – 788 dupl. rec. to ML (217 with MD > 0), 63 to Lib 1. and 7 to Lib 2.

Lib. 4. 1 969 rec. – 477 dupl. rec. to ML i Lib. 1-3 (254 with MD > 0).

The catalogues of the four joining libraries contained 14 034 records. 4 309 pairs of records, *i.e.* approximately 31%, was found to be duplicate, among them 2 848 pairs, *i.e.* approximately 20%, were nearly identical (MD = 0), and so easy to find, while 11%, *i.e.* 1 450 pairs of duplicate records with a non-zero distance, were found mainly due to the application of the text algorithms. About 30 cases of duplicate records with mistyped ISBN numbers were found.

n-grams proved to be helpful especially in the case of short texts, *e.g.* authors. After a number of tests, the comparison was based only on digrams. In the case of short texts, threshold values for trigrams had to be too great in comparison with the length of texts. In the case of long texts, the comparison of trigrams added little value to the information obtained from the comparison of digrams. Generally, in the case of long texts, the comparison of profiles was precise enough, while being much faster then the comparison of n-grams.

The comparison of bibliographic records helped to improve data quality and identify some types of differences and librarian errors.

Generally, the presented approach to joining catalogues into the existing union catalogue was approved at the Main Library of the Warsaw University of Technology as helpful and efficient. This approach can also be applied to detect approximately duplicate information in other catalogues and databases.

References

[1] Cousins S A 1998 *J. Information Science* **24** (4) 231
[2] Preece B 2001 *The Journal of Academic Librarianship* **27** (6) 470
[3] Jolibois S, Mouze-Amady M, Chouaniere D, Gradjean F, Nauer E and Ducloy J 2000 *Work & Stress* **14** (4) 283 (on-line DOI: 10.1080/02678370110040056)
[4] Atallah M J, Chyzak F and Dumas P 2001 *Algorithmica* **29** 468 (on-line DOI: 10.1007/s004530010062)
[5] Baeza-Yates R and Navarro G 1999 *Algorithmica* **23** 127
[6] Tian Z, Lu H, Ji W, Zhou A and Tian Z 2002 *Int. J. Digital Libraries* **3** (4) 325 (on-line DOI: 10.1007/s007990100044)