Record Matching System for Publication Dataset using Multi-pass Sorted Neighborhood Method

Soe Lai Yi
Computer University (Pathein)
soebetyeepky@gmail.com

Abstract

Record matching is the task of identifying records that match the same real world entity. Detecting data records that are approximate duplicates, is an important task. Datasets may contain duplicate records concerning the same real-world entity because of data entry errors, unstandardized abbreviations, or differences in the detailed schemas of records from multiple databases. This paper describes a record matching algorithm, is based on the multi-pass sorted neighborhood method for publication datasets. It also detects data duplication over publication xml database, produces a higher percentage of correct duplicates and a lower percentage of false positive, on multiple key sorting pass. Multi-pass approach is used, which is based on the combination of keys. Since no single key is sufficient to catch all matching records, combining results of individual passes produces more accurate results at lower cost. According to experimental results, multi-pass approach is at lowest false positive error (FPE) and lowest false negative error (FNE).

Keywords : record matching, approximate duplicate, multi-pass sorted neighborhood method

1. Introduction

Many of the current technological improvements have lead to an explosion in the growth of data available in digital form. Because of widespread distribution and publishing of data, the availability of these data sources increases not only the amount of data, but also the variety of and quality in which such data appears. This paper presents the detection of multiple representations of a single entity.

Information systems are employed by users of different organizations under a common goal. Record Matching, also known as Record Linkage[5] or Object Identity problem[6], is the problem of identifying if two records are related to the same real world entity. Sorted Neighborhood Method (SNM) is based on the idea of comparing only records included in a sliding window, in order to establish their matching. Multiplass algorithm is applied to SNM in key creation phase and record matching process to improve accuracy.

This paper is organized as follows. Section 1 is the introduction, section 2 is related work. In section 3, proposed system design and duplicate detection process. About record matching algorithm is presented in section 4. Section 5 is the system implementation and sample case study for record matching process. Section 6 is the conclusion and future work of the system.

2. Related Works

Record matching problem arises whenever records that is not identical, in a bit-by-bit sense – or in a primary key value sense. Information systems must do record matching to correlate different pieces of information about the same person when social security numbers are missing or incorrect.

Errors due to data entry mistakes, faulty sensor readings or more malicious activities, provide scores of erroneous datasets that propagate errors in each successive generation of data. The problem of merging two or more databases has been tackled in a straightforward fashion by a simple sort of the concatenated data sets followed by a duplicate elimination phase over the sorted list [2]. However, when, the databases involved are heterogeneous, meaning they do not share the same schema, or that the same real-world entities are represented differently in the datasets, the problem of merging becomes more difficult. The first issue, where databases have different schema, has been addressed extensively in the literature and is known as the schema integration problem [1].

3. Record Matching

The fundamental problem in merge purge is that the data supplied by various sources typically include identifiers or string data that are either different among different datasets or simply erroneous due to a variety of reasons. Determining that two records from two databases provide information about the same entity can be highly complex. One important task in data cleansing is to deduplicate records. In a normal client database, some clients may be represented by several records for various reasons:

- Incorrect or missing data values because of data entry errors;
- Inconsistent value naming conventions because of different entry formats and use of abbreviations such as 'ONE' vs. '1';
- Incomplete information because data is not captured or available;
- Clients do not notify change of address;
- Clients misspell their names or give false address (incorrect information about themselves).

3.1 Categories of Record Matching

Record matching can be broken down into three categories:

- Duplicate record removal or linkage: The same person, business, or thing is present more than once in a database. Duplicate records are identified and then linked together, merged or one is removed (purged).
- Database linkage: Two databases are linked or merged.
- Approximate database search: A database is searched for records similar to an input record.

Record matching problem is the process of record linkage or object identify problem. It is the problem of identifying if two records represent the same real world entity. The records may belong to a same database or to different databases.

4. Proposed System Design

This paper presents record matching over publication dataset. It is a method to make automatic record matching in order to improve quality in large information systems. It is based on the Sorted Neighborhood Method. It is an automatic choice of the key to perform the matching. Multi-pass algorithm is applied in key selection part of SNM. It is different key selection over multiple combinations. XML Parsing is necessary to parse XML publication records into the system. In this system XML Parser is used. Process flow of the System is shown in Figure 1. It shows how record matching is performed in each single pass and multi-pass (combining all single passes) and performance analysis is performed for them.

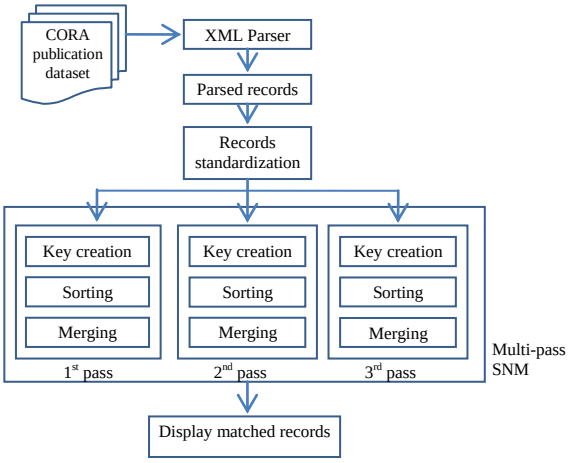

Figure 1: Proposed System Design

4.1. Parsing and Standardization

CORA XML dataset includes publication records of different authors. It includes authornames, title, journal, proceeding in, date, volume, pages, edition, institution, etc. XML parser is used to parse XML documents. It includes Document and Element object model in the representation of the tree structure of the XML document.

```
<CORA>
<NEWREFERENCE id="1">
ahlskog1994a <author> M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. </author> <title> Inganas and M.R. </title> <journal> Andersson, J Appl. Phys., </journal> <volume> 76, </volume><pages>893,</pages> <date> (1994). </date></NEWREFERENCE>
<NEWREFERENCE id="2">
ahlskog1994a <author> M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R. Andersson, </author> <journal> J Appl. Phys., </journal> <volume> 76, </volume><pages>893, </pages> <date> (1994). </date></NEWREFERENCE>
<NEWREFERENCE id="3">
ahlskog1994a <author> M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R. Andersson, </author><journal> J Appl. Phys.,</journal> <volume> 76, </volume><pages>893, </pages><date> (1994). </date></NEWREFERENCE>
</CORA>
```

Figure 2: XML publication datasets

In this paper, author name, date and title are standardized. Author names include all authors participated in the publication. Only first author is extracted and formatted into first character of First Name and Last name only. Date includes one or

combination of year, month and dates. This system extracts only year. Figure 2 shows sample CORA XML publication records and Table 1 is the publication records after parsing. It contains the publication records stored in the XML file. Table 2 describes records before standardization and after standardization. In this paper, author's first name and last name is standardized into first letter of first name with dot '.' and last name. In 'Date' field, it removes all other characters except numeric values of 'Year'.

Table 1: Sample results in parsed records

REF-No	NR-Name	Author	Title	Date	..
1	Ahlskog1994a	M. Ahlskog, J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O.	Instance-Based Learning Algorithms	(1994.)	
2	paloheimo1994a	J. Paloheimo, H. Stubb, P. Dyreklev, M. Fahlman, O. Inganas and M.R. Andersson	Fast perceptual learning in hyper-actuity	(1994.)	
3	fahlman1994a	M. Fahlman, O. Inganas and M.R. Andersson	The Correlation Learning Architecture	(1994).	

Table 2: Data Standardization

Author names	Standardized Author names	Date	Standar-dized date
Steve Benford	S. Bebford	(1994)	1994
Bettian Buth	B. Buth	September (1992)	1992
David Aha	D. Aha	May 1998	1998
David Kibler	D. Kibler	(1997)	1997

4.2. Multi-pass Sorted Neighborhood Method

Multi-pass Sorted Neighborhood method is a record matching algorithm for large databases, based on the idea of comparing only records included in a sliding window, in order to establish their matching.

It consists of three distinct steps:

- **Choice of the matching key**: The choice of the key is a fundamental step of the matching process, as the results of the matching are directly influenced by it. Multiple key selections is performed in each independent pass.
- **Sorting** of records according to the chosen key.
- **Merging**: Moving of a fixed size window through the list of records and comparisons only of the records included in the window. Each couple of records in the window is compared in order to decide if the two records match or not.

4.2.1 Key creation

The effectiveness of the SNM highly depends on the key selected to sort the records. A key is defined to be a sequence of a subset of attributes. Example of records and keys used in this system are shown in Table 3, where key is created by first letter of first name, three consonants of last name and year.

Table 3: Example for Key creation

First	Last	Year	Key
M.	Ahlskog	1999	MHLS1995
Steve	Benford	1993	SBNF1993
S.	Benford	1993	SBNF1993
S.	Fahlman	1974	SFHL1974

No single key is sufficient to catch all matching records. The attributes or fields that appear first in the key have higher discriminating power than those appearing after them. Multi-pass approach is to execute several independent runs of the sorted-neighborhood method, each time using a different key and a relatively small window. Intuitively, different runs cover different true matches, so the union should cover most of the true matches.

Table 4: Key Creation by Multi-pass

First Pass	First name of first author	Last Name of first author	Year of Date
Second Pass	Last Name of first author	First Name of first author	Year of Date
Third Pass	Year of Date	First Name of first author	Last Name of first author

The effectiveness of a multi-pass approach depends on which attributes are chosen and the

methods used. In this paper, three passes are used to implement the Multi-pass algorithm. For the key creation in multi-pass, three passes are used and key creation for each pass is shown in Table 4.

4.2.2 Sorting and Merging

After key creation process, in the SNM, data list is sorted according to its key in key creation phase and merged its neighbors according to sliding window. It moves a fixed size window through the sequential list of records limiting the comparisons for matching records to those records in the window.

If the size of the window is w records, then every new record entering the window is compared with the previous w – 1 records to fine matching records. The first record in the window slides out of the window.

5. Experimental Results

This system is implemented using Java programming language. Jdk 1.6 is used to implement system. It uses CORA.xml files for record matching process. It includes 1892 publishing records.

In this paper, performance of the algorithm is measured using False Positive error (FP) and False Negative error (FN). False positive error is a percentage of wrongly identified duplicates.

$$FP = \frac{\text{Number of wrongly identified duplicates}}{\text{Total number of identified duplicates}} \times 100\ \%$$

False negative error indicates the percentage of duplicates that a given algorithm could not identify.

$$FN = \frac{\text{Number of duplicates that escape identification}}{\text{Total number of duplicates}} \times 100\ \%$$

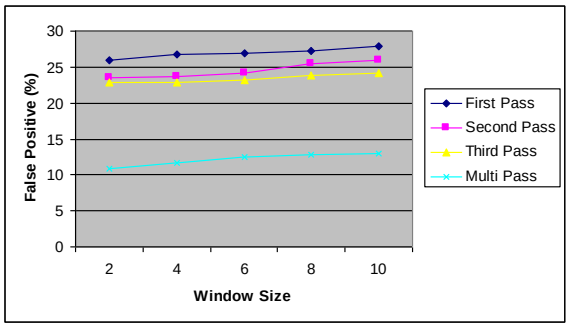

Figure 3: (%) of False Positive for different passes (Threshold = 0.8)

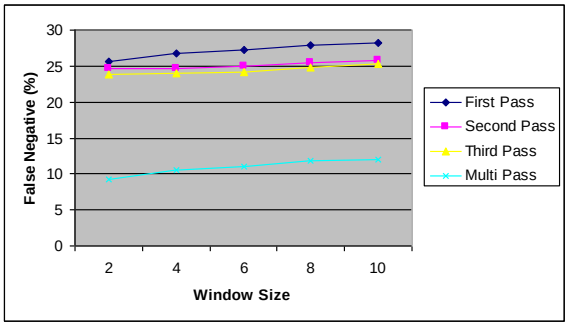

Figure 4: (%) of False Negative for different passes (Threshold = 0.8)

This system is tested with different sliding window size 2, 4, 6, 8 and 10 and threshold value 0.7 and 0.8. After the experimental results, threshold 0.8 with window size 10 has the best results. Performance analysis for different sliding windows with different threshold is shown in Figure 3 and Figure 4.

6. Conclusion

This paper presents an algorithm for record matching based on the Sorted Neighborhood Method. It is an automation of the matching key selection. This algorithm allows identifying the different copies of a given data available in the large information system as the same objects. Multi-pass algorithm is used to detect matched records, which run different keys on independent passes. It satisfies the quality dimensions, including currency. According to experimental results, it can detect more duplicate records than single passes.

7. References

[1] C.Batini, M. Lenzerini, and S.Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surverys, 18(4):323-364, December 1986.

[2] D.Bitton and D.J.DeWitt. Duplicate Record Elimination in Large Data Files. ACM Transactions on Database Systems, 8(2):255-265, June 1983.

[3] M.Hernandez and S.Stolfo. The Merge/Purge Problem for Large Databases. In Proceedings of the 1995 ACM-SIGMOD Conference, May 1995.

[4] Mauricio A.Hernadez and Salvatore J.Stolfo, Real-world data is dirty: Data cleansing and the merge/purge problem, Data Mining and Knowledge Discovery 2 (1998), no. 1, 9-37.

[5] S. J. Axford H. B. Newcombe, J. M. Kennedy and A. P.F. James, “Automatic Linkage of Vital Records,” Science, vol. 130,1959.

[6] Y. R Wang and S. Madnick, “The inter-database instance identification problem in integrating autonomous systems,” in Proceedings of the 5th International Conference on Data Engineering (ICDE 1989), Los Angeles, California, USA, 1989.