Probabilistic Data Generation for Deduplication and Data Linkage

http://datamining.anu.edu.au/linkage.html

Peter Christen

Department of Computer Science, Australian National University,
Canberra ACT 0200, Australia
peter.christen@anu.edu.au

Abstract. In many data mining projects the data to be analysed contains personal information, like names and addresses. Cleaning and preprocessing of such data likely involves deduplication or linkage with other data, which is often challenged by a lack of unique entity identifiers. In recent years there has been an increased research effort in data linkage and deduplication, mainly in the machine learning and database communities. Publicly available test data with known deduplication or linkage status is needed so that new linkage algorithms and techniques can be tested, evaluated and compared. However, publication of data containing personal information is normally impossible due to privacy and confidentiality issues. An alternative is to use artificially created data, which has the advantages that content and error rates can be controlled, and the deduplication or linkage status is known. Controlled experiments can be performed and replicated easily. In this paper we present a freely available data set generator capable of creating data sets containing names, addresses and other personal information.

1 Introduction

Finding duplicate records in one, or linking records from several data sets are increasingly important tasks in the data preparation phase of many data mining projects, as often information from multiple sources needs to be integrated, combined or linked in order to allow more detailed data analysis or mining. The aim of such linkages is to match all records related to the same entity, such as a patient or customer. As common unique entity identifiers (or keys) are rarely available in all data sets to be linked, the linkage process needs to be based on the existing common attributes.

Data linkage and deduplication can be used to improve data quality and integrity, to allow re-use of existing data sources for new studies, and to reduce costs and efforts in data acquisition. In the health sector, for example, linked data might contain information that is needed to improve health policies, and that traditionally has been collected with time consuming and expensive survey methods. Businesses routinely deduplicate and link their data sets to compile

Table 1. Data sets used in recent publications.

Data set	Publication
Cora [16]	[16] (2000), [7] (2002), [2, 3] (2003), [12] (2004)
Restaurant [22]	[7, 22] (2002), [2, 3] (2003), [20] (2004)
Citeseer [2]	[21] (2002), [2, 3] (2003)
Proprietary or confidential	[24] (2000), [7, 10, 21, 22] (2002), [20, 23] (2004)
Artificially generated data [1, 6, 14]	[14] (1995), [10] (2002), [1] (2003), [12] (2004)

mailing lists, while in taxation offices and departments of social security data linkage can be used to identify people who register for benefits multiple times or who work and collect unemployment money. Another application of current interest is the use of data linkage in crime and terror detection, which increasingly rely on the ability to quickly bring up files for a particular individual that may help to prevent crimes or terror by early intervention.

As data linkage and deduplication is often dealing with data sets that contain (partially) identifying attributes (like names, addresses, or dates of birth), it can be difficult for a researcher to acquire standard data for testing and evaluation of new linkage algorithms and techniques. For a user, it is challenging to learn how to use and customise data linkage systems effectively without data sets where the linkage status is known. An alternative is the use of artificially generated data, which we will discuss in the following section.

2 Data Linkage, Deduplication and Artificial Data

Computer-assisted data (or record) linkage goes back as far as the 1950s, and the theoretical foundation has been provided by [11] in 1969. The basic idea is to link records by comparing common attributes, which include person identifiers (like names and dates of birth) and demographic information (like addresses).

In recent years, researchers started to explore machine learning and data mining techniques to improve the linkage process. Clustering [5, 10, 16], active learning [21, 22], decision trees [10, 22], graphical models [20], and learnable approximate string distances [2, 3, 8, 17, 23, 24] are some of the techniques used.

In these publications various data sets (some publicly available, others proprietary or even confidential) were used in experimental studies, as shown in Table 1. This variety makes it difficult to validate the presented results and to compare new deduplication and linkage algorithms with each other. Tuning of parameters can result in high accuracy and good performance for a certain algorithm on a specific data set, but the same parameter values might be less successful on other data or in different application areas.

There is clearly a lack of publicly available real world data sets for deduplication and data linkage, which can be used as standard test beds (or test decks) for developing and comparing algorithms, similar to data collections used in information retrieval (TREC) or machine learning (UCI repository [4]). However, because many real world data sets contain personal information, privacy and confidentiality issues make it unlikely that they can be made publicly available. Using de-identified data, where e.g. names and addresses are encrypted or removed, is not feasible either, as many linkage algorithms specifically work on name and address strings [6, 11].

Artificially generated data can be an attractive alternative. Such data must model the content and statistical properties of comparable real world data sets, including the frequency distributions of attribute values, error types and distributions, and error positions within these values. Typographical errors have been analysed in a number of studies [9, 13, 19], and are important issues in the areas of error correction in text [15] and approximate string matching [13]. One of the earliest studies [9] found that over 80% of typographical errors were single errors, either an insertion, deletion or substitution of a character, or transposition of two adjacent characters. Substitutions were the most common errors, followed by deletes, then inserts and finally transpositions, followed by multiple errors. Similar results were reported by others [13, 15, 19].

Names and addresses are especially prone to data entry errors. Different error characteristics will occur depending upon the mode of data entry [15], for example manually typed, scanned, or via automatic speech recognition. Optical character recognition [13, 19] (scanning) will lead to substitution errors between similar looking characters (e.g. *'q'* and *'g'*), while keyboard based data entry can result in wrongly typed neighbouring keys. Data entry over the telephone will mainly lead to phonetical errors, which seem to occur more likely towards the end of names [19]. While for many regular words there is only one correct spelling, there are often different written forms of proper names, for example *'Gail'* and *'Gayle'*. Additionally, names are often reported differently by the same person depending upon the organisation they are in contact with, resulting in missing middle names, initials only, or even swapped name parts.

Artificially generating duplicate records based on real world error distributions will result in data sets that have characteristics similar to real world data. A first such data generator (called *DBGen* or *UIS Database Generator*)[1] that allows the creation of databases containing duplicates records was presented in [14]. It uses lists of names, cities, states, and postcodes, and provides a large number of parameters, including the size of the database to be generated, percentage and distribution of duplicates, and the amount and types of errors introduced. An improved generator is described in [1], that allows for missing attribute values and increased variability in the set of possible values generated.

3 A Probabilistic Data Set Generator

We have developed a data set generator based on ideas from [14] and improved in several ways. Our generator can create data sets containing names and addresses (based on frequency tables), dates, telephone and identifier numbers (like social

[1] Available from: `http://www.cs.utexas.edu/users/ml/riddle/data.html`

security numbers). It is implemented as part of the *Febrl* [6] data linkage system, and freely available under an open source software license. A user can easily modify and improve the generator according to her or his needs.

A user specified number of *original* records are generated in the first step, and in the second step *duplicate* records are created based on these original records by randomly introducing errors. Each record is given a unique identifier as can be seen in Figure 1. This allows the evaluation of error rates (false linked non-duplicates and non-linked true duplicates).

Original records are randomly created using frequency look-up tables for name and address attributes (like given- and surname; street number, name and type; locality, postcode, state or territory). These frequency tables can be compiled for example by using publicly available electronic telephone directories, or by extracting frequencies from data sets at hand, as shown in Section 4. For date, telephone and identifier number attributes, a user can specify the range (e.g. start and end date, or number of telephone digits).

Duplicate records are generated next based on the original records and according to the following parameters.

- The total number of duplicate records to be generated.
- The maximum number of errors to be introduced into one attribute in a record.
- The maximum number of errors to be introduced into one record.
- The maximum number of duplicate records to be created based on one original record.
- The probability distribution (either uniform, Poisson, or Zipf) of how many duplicates are being created based on one original record.

Duplicate records are created by randomly selecting an original record (which has so far not been used to create duplicates), followed by randomly choosing the number of duplicates to be created for it, and then randomly introducing errors according to user specified probabilities. A additional probability distribution specifies how likely attributes are selected for introducing errors (it is possible to have attributes with no errors at all). The following types of errors can be introduced.

- If an attribute value from an original record is found in a look-up table with misspellings (for example of real typographical errors), then randomly choose one of it's misspellings.
- Insert a new character at a random position into an attribute value.
- Delete a character at a random position from an attribute value.
- Substitute a character in an attribute value with another character. Substitution is based on the idea of keying errors, where the substituted character will more likely be replaced with a randomly chosen neighbouring character in the same keyboard row or column, than with another character (that is not a keyboard neighbour).
- Transpose two adjacent characters at a random position in an attribute value.
- Swap (replace) the value in an attribute with another value (similar to when a value was randomly created when the original records were generated).
- Insert a space into an attribute value and thus splitting a word.

```
REC_IDENT,   GIVEN_NAME, SURNAME,  STR_NUM, ADDRESS_1,        ADDRESS_2, SUBURB,    POSTCODE
rec-0-org,   james,      whiteway, 2,       maribyrnong ave, aird,      red hill, 2611
rec-1-org,   mitchell,   devin,    26,      knox st,          chelvy,    holder,   2606
rec-2-dup-0, james,      sayl,     73,      chauncy cres,     ,          watson,   2913
rec-2-dup-1, jame,       ,         73,      chauncy cres,     ,          watson,   2913
rec-2-dup-2, jaems,      salt,     73,      chauncy pl,       ,          watson,   2913
rec-2-org,   james,      salt,     73,      chauncy cres,     ,          watson,   2913
rec-3-org,   mitchell,   polmear,  341,     fitchett st,      ,          o'connor, 2605
rec-4-dup-0, isaad,      white,    15,      tyrrell circ,     tagarra,   rivett,   2906
rec-4-dup-1, isaac,      wiglht,   15,      tyrrell circ,     ,          rivett,   2906
rec-4-org,   isaac,      white,    15,      tyrrell circ,     ,          rivett,   2906
rec-5-dup-0, elle,       webb,     5,       burnie pl,        ,          bruce,    2617
rec-5-org,   elle,       webb,     3,       burnie pl,        ,          evatt,    2617
```

Fig. 1. Generated example data set with 6 original and 6 duplicate records, a maximum of 3 duplicates per record, and maximum 2 errors per attribute and per record.

- Delete a space in an attribute value and merge two words (this is obviously only possible if an original attribute value contains at least two words).
- Set an attribute value to missing (with a user definable missing value).
- Given an original attribute value is missing (or empty), insert a randomly chosen new value (similar to when creating the original records).
- Swap the values of two attributes in a record (e.g. surname with given name).

Following studies on real world typographical errors [15, 19], single character errors (inserts, deletes, etc.) are more likely introduced in the middle or towards the end of attribute values when the duplicate records are created.

4 Experimental Study

In order to evaluate the generation of artificial data, we conducted a study using the New South Wales Midwives Data Collection (MDC) [18]. We extracted 175,211 records from the years 1999 and 2000. The eight attributes used in our study were the mother's name (given- and surname), address (street number and name, locality and postcode) and date of birth, as well as the baby's date of birth. The data set contained 5,331 twin and 177 triplet births (which were assumed to be duplicates in the attributes describing the mother). Additional duplicates were from mothers giving birth twice (or even three times) within the two years period (possibly recorded with changed names and addresses). Unfortunately we did not have access to the duplication status.

We first extracted frequency tables for the attributes listed above, and then generated three artificial data sets using these tables, containing 5% , 10% and 20% duplicates, respectively. Table 2 shows the average frequencies and standard deviations of the attributes in the original and generated data sets, and Figure 2 shows a selection of the corresponding sorted frequency distributions.

As can be seen all three generated data sets have frequency distributions as well as standard deviations similar to the original data set. Different error types were introduced into the various attributes. These were mainly typographical errors in the attributes containing name strings, while in the date attributes values were mainly swapped with another value from the corresponding frequency table

Table 2. Average frequencies and standard deviations of attribute values in MDC data sets (original and generated with given percentage of duplicates).

Attribute	Original	Generated 5%	Generated 10%	Generated 20%
Surname	3.5 / 16.6	4.3 / 17.6	3.9 / 16.4	3.4 / 14.4
Given name	5.0 / 45.9	6.3 / 49.8	5.8 / 46.4	5.2 / 41.8
Street number	12.3 / 125	15.9 / 138	16.1 / 139	3.4 / 137
Street name	2.4 / 5.4	3.0 / 5.7	2.9 / 5.8	2.6 / 5.1
Postcode	224 / 337	167 / 294	154 / 284	138 / 267
Locality	55.7 / 123	30.5 / 91.6	21.9 / 76.8	14.7 / 60.7
Mother date of birth	16.6 / 12.1	16.6 / 12.1	16.6 / 12.1	16.6 / 12.1
Baby date of birth	240 / 42.1	225 / 62.8	225 / 63.8	224 / 64.4

(resulting in nearly consistent average frequencies and standard deviations). For most attributes an increased percentage of duplicates resulted in smaller average frequencies and standard deviations, as the number of different attribute values was increased by the introduction of typographical and other errors.

5 Discussion and Outlook

We have discussed the issues and problems associated with real world test data for deduplication and data linkage, and presented a freely available data set generator. Improvements on our generator include the relaxation of the independent assumption, i.e. instead of creating attribute values independently, use frequency distributions for value combinations. Similarly, the introduction of errors and modifications could be based on statistical dependencies between attributes. For example, if a person moves, most of her or his address attributes (like street number and name, postcode and locality) will change. Another interesting extension would be to generate groups of records representing households (useful for generating census style data). Further fine-tuning the methods of how errors and modifications are introduced (for example character substitution based on scanning errors of handwritten forms) is another area of possible improvements. We are also planning to do further comparison studies, specifically we are interest in comparing the deduplication and linkage outcomes for real world and artificially created data, to see if similar error rates are achieved. Artificially generated data can also be useful for research in the areas of approximate string comparisons as well as correcting errors in text.

Acknowledgements

This work is supported by an Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the NSW Department of Health. The author would like to thank William Winkler, Tim Churches and Karl Goiser for their valuable comments.

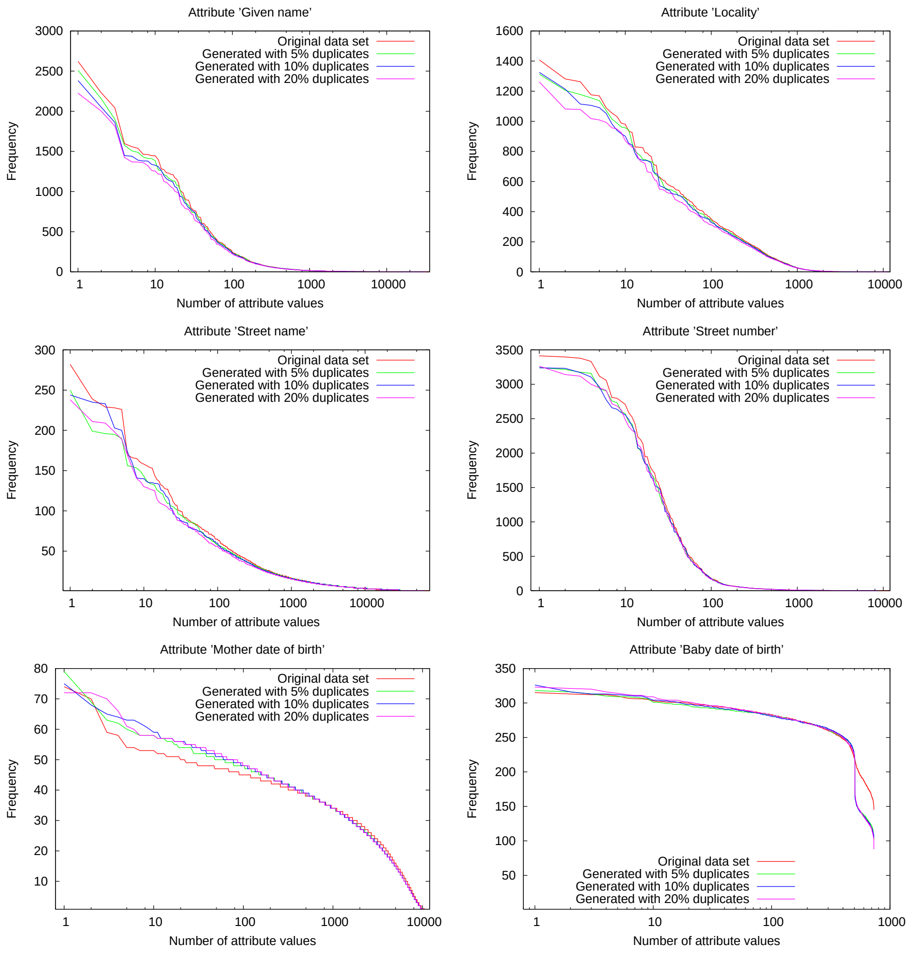

Fig. 2. Selected MDC sorted frequency distributions (log-scale on horizontal axis).

References

1. Bertolazzi, P., De Santis, L. and Scannapieco, M.: Automated record matching in cooperative information systems. Proceedings of the international workshop on data quality in cooperative information systems, Siena, Italy, January 2003.
2. Bilenko, M. and Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. Proceedings of the 9th ACM SIGKDD conference, Washington DC, August 2003.
3. Bilenko, M. and Mooney, R.J.: On evaluation and training-set construction for duplicate detection. Proceedings of the KDD-2003 workshop on data cleaning, record linkage, and object consolidation, Washington DC, August 2003.

4. Blake, C.L. and Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
5. Chaudhuri, S., Ganti, V. and Motwani, R.: Robust identification of fuzzy duplicates. Proceedings of the 21st international conference on data engineering, Tokyo, April 2005.
6. Christen, P., Churches, T. and Hegland, M.: A parallel open source data linkage system. Proceedings of the 8th PAKDD, Sydney, May 2004.
7. Cohen, W.W. and Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
8. Cohen, W.W., Ravikumar, P. and Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. Proceedings of IJCAI-03 workshop on information integration on the Web (IIWeb-03), pp. 73–78, Acapulco, August 2003.
9. Damerau, F.: A technique for computer detection and correction of spelling errors. Communications of the ACM, vol. 7, no. 3, pp. 171–176, March 1964.
10. Elfeky, M.G., Verykios, V.S. and Elmagarmid, A.K.: TAILOR: A record linkage toolbox. Proceedings of the ICDE' 2002, San Jose, USA, March 2002.
11. Fellegi, I. and Sunter, A.: A theory for record linkage. Journal of the American Statistical Society, December 1969.
12. Gu, L. and Baxter, R.: Adaptive filtering for efficient record linkage. SIAM international conference on data mining, Orlando, Florida, April 2004.
13. Hall, P.A.V. and Dowling, G.R.: Approximate string matching. ACM computing surveys, vol. 12, no. 4, pp. 381–402, December 1980.
14. Hernandez, M.A. and Stolfo, S.J.: The merge/purge problem for large databases. Proceedings of the ACM SIGMOD conference, May 1995.
15. Kukich, K.: Techniques for automatically correcting words in text. ACM computing surveys, vol. 24, no. 4, pp. 377–439, December 1992.
16. McCallum, A., Nigam, K. and Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. Proceedings of the 6th ACM SIGKDD conference, pp. 169–178, Boston, August 2000.
17. Nahm, U.Y, Bilenko M. and Mooney, R.J.: Two approaches to handling noisy variation in text mining. Proceedings of the ICML-2002 workshop on text learning (TextML'2002), pp. 18–27, Sydney, Australia, July 2002.
18. Centre for Epidemiology and Research, NSW Department of Health. New South Wales Mothers and Babies 2001. NSW Public Health Bull 2002; 13(S-4).
19. Pollock, J.J. and Zamora, A.: Automatic spelling correction in scientific and scholarly text. Communications of the ACM, vol. 27, no. 4, pp. 358–368, April 1984.
20. Ravikumar, P. and Cohen, W.W.: A hierarchical graphical model for record linkage. Proceedings of the 20th conference on uncertainty in artificial intelligence, Banff, Canada, July 2004.
21. Sarawagi, S. and Bhamidipaty, A.: Interactive deduplication using active learning. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
22. Tejada, S., Knoblock, C.A. and Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
23. Yancey, W.E.: An adaptive string comparator for record linkage RR 2004-02, US Bureau of the Census, February 2004.
24. Zhu, J.J., and Ungar, L.H.: String edit analysis for merging databases. KDD-2000 workshop on text mining, held at the 6th ACM SIGKDD conference, Boston, August 2000.