Record linkage using error-prone strings

Rainer Schnell, Tobias Bachteler, University of Konstanz
Stefan Bender, Institute for Employment Research (IAB)
email: Rainer.Schnell@uni-konstanz.de

Key Words: Record linkage, string similarity

Abstract:

We developed a record-linkage toolbox in order to link data provided by survey respondents to individual work histories. We compare the performance of various deterministic record-linkage procedures for German surnames employing different string-similarity measures. Edit distance overall performed quite well. The application of this similarity measures allowed linking 38% of the respondents by using sex, age and workplace as keys.

1. Introduction

Survey methodologists are often faced with the problem of linking data reported by survey respondents with data from other sources. This can be a difficult task since respondent data are prone to various kinds of error (memory lapses, spelling and typographical errors, and so on). Matching these data to other data is problematic since even small errors prevent the use of exact-match algorithms. Since most database and statistical programs offer only exact-match routines, additional programs must be employed to perform the matches for data that contains errors. AUTOMATCH was one of the few such programs commercially available. Since its discontinuation, there has been a dearth of commercial products that fill this niche, at least of those with a price affordable for a German university or federal agency. Even if such a product were available, it would be difficult to know which string-similarity measure should be used, since only a few comparisons of string-similarity measures have ever been published in journals. Of course, even fewer comparisons have been published for German surnames. Due to the fact that all known comparisons using German surnames are based on artificially generated errors, not on actual human errors, we decided to conduct a study comparing various routines. Doing so required us to develop our own record linkage program.

This work was partially funded by the German Research Foundation and by the Institute for Employment Research (IAB). The survey was financed by the Max-Planck Institute for Human Development (MPI), the IAB and the European Social Fund. Perl and Java programming were done by Joerg Reiher. We would like to thank Elisabeth Coutts for her help with translation.

2. Program development

The program specification resulted in the assignment of subsets of the total task list to three programs: a pre-processing tool, a deterministic record-linkage module and a manual editing module.

2.1 Pre-Processing Tool

The purpose of this tool is to transform raw data entered using such programs as Epi-Data into a form suitable for record linkage. This tool is therefore able to read such common data-file formats as ASCII-CSV or Xbase. The tool then writes such data in a standard format: Since Stata is increasingly used in academic environments and its file format is openly published, we decided to use Stata (version 7) as the internal file format. A further function of the pre-processing tool is its ability to remove typical but unnecessary strings (academic titles, nobility or corporation prefixes, numbers, and so forth) and standardize the spelling of such prefixes as Mc or Mac. To this end, we have collected a number of different prefix lists (for companies, institutions and people). The program also allows for the replacement of German umlauts with usual ASCII codes < 127. A further issue is the treatment of women's married names. Revisions to German marriage laws mean that double and triple names are now quite common in Germany. The pre-processing tool therefore contains an option to parse double and triple names according to delimiters like dashes or blanks. The selection of this option creates separate records for the whole string and each sub-component. Finally, the program has an option to standardize different date formats. This module is written in Java to ensure portability across many different platforms.

2.2 Deterministic Record-Linkage Module

This module operates on Stata files containing only preprocessed key variables and identification num-

bers. The main task of this module is the computation of a wide variety of string-similarity measures. Currently, the module computes string similarities based on the following measures: bi- and trigrams (Ukkonen, 1992); different versions of edit distances (Ukkonen, 1985) including LCS (Hirschberg, 1977), Jaro's String Comparator and its variants (especially the Winkler version (Porter & Winkler, 1997)); Soundex (Knuth, 1998) (including German variants); Metaphone (Philips, 1990); Double-Metaphone (Philips, 2000); Phonex (Lait & Randell, 1996); NYSIIS (Taft, 1970); Guth (Guth, 1976) and Synoname (Borgman & Siegfried, 1992). Within subgroups formed according to several user selectable variables ("blocking"), string-similarities of all possible pairs are computed. The program output is a Stata data set containing both a selectable number of potential matches (pairs) for each identification variable and a similarity measure for each of those pairs. Since many of the program's subroutines were already available as modules in the "Comprehensive Perl Archive Network" (CPAN), the main program was implemented in Perl[1].

2.3 Manual Editing Module

The most labor-intensive part of a record-linkage project is the manual matching of cases not matched by the program. We therefore wrote a manual editing module intended to make this tedious task as easy as possible. The program displays two data sets in horizontally aligned data-browser windows. The program allows independent scrolling in both windows with a user-selectable view of different variables. Data sets can also be sorted according to variables of the user's choosing. The implementation of the AGREP routine means that searching by wild cards is also possible. The actual manual linkage is performed by pointing and clicking with a mouse. Optionally, cases already matched can be hidden from view. This module is also implemented in Java.

3. An experimental comparison of string-similarity measures using German names

Having created a new tool for deterministic record-linkage that employed various string-similarity measures, we conducted an experiment to compare the performance of those measures. A string-similarity measure for the intended application should perform well on German names with human-generated errors. Furthermore, it should have an acceptable running time for data sets that contain even 2.2 million records. The test should be conducted with files with known true links and exactly one true link per key.

[1] A Java version has been completed recently

3.1 Design

Due to running time considerations, only a subset of the implemented string similarity measures was tested: edit-distance, LCS, bigrams, trigrams, Jaro, Jaro-Winkler, NYSIIS and different versions of Soundex. 1200 hundred different names were randomly selected from a town register. 12 tapes with 100 different names each were recorded with a female voice. Each tape was randomly assigned to one of 12 groups of medical students with an average of 12 students per group. Each student writes (by hand) the name heard. All student notes were typed by a single typist. For each similarity measure, performance indices were computed. The following indices were used (see table 1):

$$sensitivity = A/(A+C) \tag{1}$$

$$Precision = A/(A+B) \tag{2}$$

$$1 - Specificity = B/(B+D) \tag{3}$$

Table 1: Definitions for indices

	true link	incorrect link
link	A	B
non-link	C	D

3.2 Results

The phonetic algorithms form a distinct cluster. They perform much worse than the non-phonetic algorithms. Within the non-phonetic algorithms, the performance of all algorithms is very similar. To keep the graphics clean, these algorithms were excluded from the following figures. As can be seen in Figures 1 and 2, overall the performance of Jaro-Winkler and edit-distance was always among the best, the performance of trigrams nearly always among the worst of the non-phonetic algorithms, regardless which performance is being used. Due to running time considerations, we decided to use Jaro-Winkler in our main application.

4. Application

We tried to link survey respondent data with company register data by using sex, month/year of birth, name of the employer and name of the workplace.

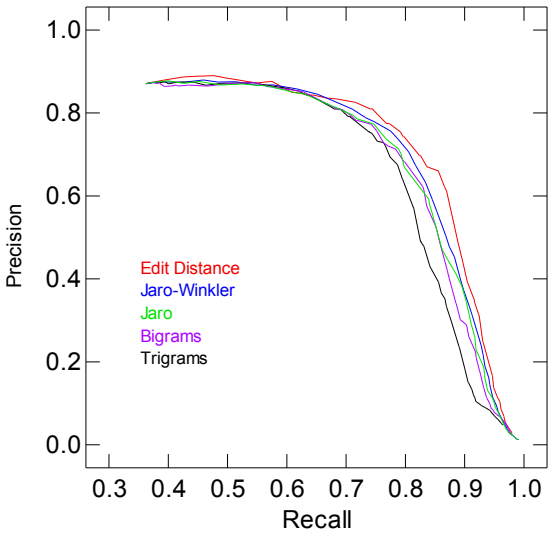

Figure 1: PR-plot

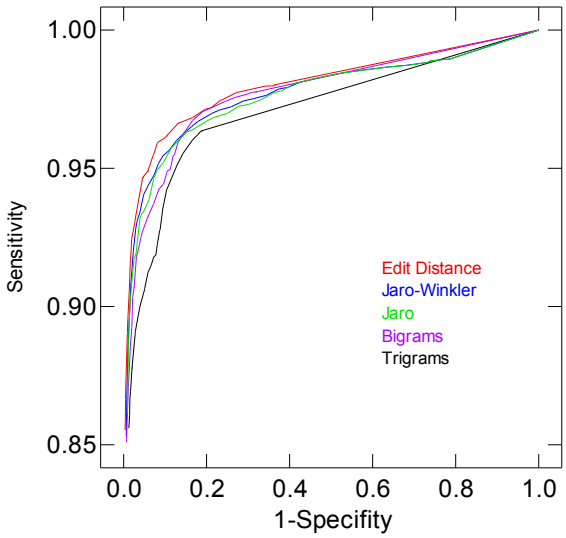

Figure 2: ROC-plot

In order to check the linkage, we used a known true-link file. In total, we used three different data sets for this purpose.

4.1 Data Set I: German Life-History Study

This data set is based on a retrospective survey consisting of quantified life histories (measured in months) for different dimensions of personal life (Brückner & Mayer, 1998).[2] Life-course protocols are edited and corrected using taped recordings. Interviews were conducted in 1998 with 2,911 people born in 1964 or 1971 in West Germany. A social-security number is available for 730 of those people. They gave their permission to match their responses with the data stored by the social security system.

4.2 Data Set II: German work-history file

Employers in Germany are required by law to report information on their employees to the social-security agencies. The data set used is the mandatory official registration of new hires and fires for health insurance, statutory pension scheme and unemployment insurance. Therefore, exact daily information on employment is available. Every person in the data set can be identified by a social-security number. The social-security number is highly reliable because the last digit is produced by a checksum algorithm (Bender, Haas, & Klose, 2000).

[2]The German Life History Study is conducted by the Max-Planck Institute for Human Development

4.3 Data Set III: Employer file

This official administrative data file contains information on companies who have registered employees covered by the social-security system. It contains the name and the address of the company. The employer file and the individual work-history file are directly linked by an employer i.d., whose entry is strictly controlled.

4.4 Check of true links

Names of employers as given by respondents (data set I) were linked to the employer file (data set III) using Jaro-Winkler as similarity measure within blocks formed by workplace municipality as given by respondents for those 730 cases with valid social security number. For 614 persons we have valid social security numbers in both data sets and at least one record in the MPI-data set. Using Jaro-Winkler we found for 372 persons at least one combination of employer number (data set I) and name of the firm (data set III). The thereby added employer number was used for an exact match using the employee file (data set II) within blocks formed by subgroups of firm number, sex, year and month of birth as given by the respondents. After removing statistical twins within the subgroups 325 persons were left for the last step (see fig. 3). We found 234 identical persons in both data sets (correct links) and 91 wrong links. Given response knowledge and a file of the total population (data set II) the keys used resulted in 32.1% (730 persons) or 38.1% (614 persons) of correct links. In order to compare this result with a baseline, we conducted a record linkage with exact

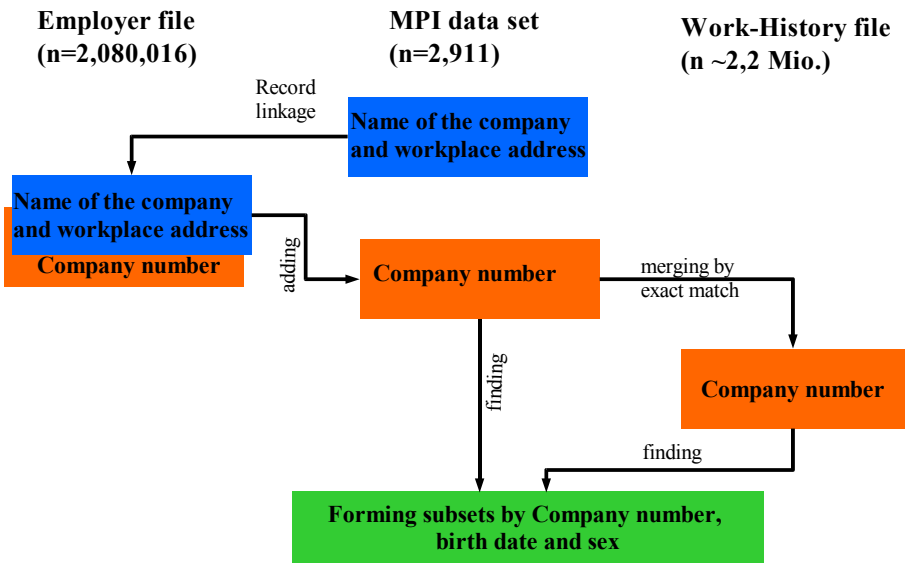

Figure 3: linkage process

matching keys only. This yields 81 persons with at least one combination of employer number (data set I) and company name (data set III). After removing statistical twins, 71 persons remained. 44 persons were correctly linked, 22 were wrong links. In sum, using Jaro-Winkler results in a 10-fold increase of linkage compared to exact matches in our data set (3.0% to 32.1% or 3.6% to 38.1%).[3]

5. Conclusion

The project results regarding linkage of survey respondent data to official employer data bases can be summed up with four statements:

1. People who don't know the employer name will provide one anyway, whether that name is correct or not.

2. Discrepancies between official employer data and employee self-report seem to be due to differences in the workplace definition of respondents.

3. Using the name of the firm and the name of the workplace municipality is a much more promising way for linkage than using a lot of different numerical variables.

4. Work on pre-processing of potential keys seems to have resulted in more correct links than work on string comparators or similarity threshold selection.

Nethertheless, our further work will concentrate on implementation of a probabilistic record linkage module, algorithms to process company abbreviations, implementation of further string similarity measures and a comparison with neural nets for pattern recognition of names[4].

References

Bacher, J., Brand, R., & Bender, S. (2002). Re-identifiying register data by survey data using cluster analysis: An empirical study. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(5), 589-608.

Bender, S., Haas, A., & Klose, C. (2000). *The iab-employment subsample: Opportunities for analysis provided by the anonymised subsample* (Discussion Paper No. 117). Bonn: IZA.

Borgman, C. L., & Siegfried, S. L. (1992). Getty's synoname and its cousins: A survey of applications of personal name-matching algorithms. *Journal for the American Society of Information Science*, *43*(7), 459-476.

[3] Taking the same data sets with 13 different variables, but only a fraction of 2% of data set II (Bacher, Brand, & Bender, 2002) found below 10% of correct links by using simple cluster analysis techniques.

[4] Details about program status and availability can be found on the project homepage: `http://www.uni-konstanz.de/FuF/Verwiss/Schnell/recordli.html`

Brückner, E., & Mayer, K.-U. (1998). Collecting life history data: Experiences from the german life history study. In J. Giele & G. E. Jr. (Eds.), *Methods of life course research: Qualitative and quantitative approaches* (p. 152-181). Thousand Oaks: Sage.

Guth, G. J. A. (1976). Surname spellings and computerized record linkage. *Historical Methods Newsletter*, *10*(1), 10-19.

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the Association for Computing Machinery*, *24*(4), 664-675.

Knuth, D. E. (1998). Sorting and searching. In *The art of computer programming* (Vol. 3, 2. ed., p. 394-395). Reading/Mass.: Addison-Wesley.

Lait, A., & Randell, B. (1996). *An assessment of name matching algorithms* (Tech. Rep. No. 550). Department of Computing Science, University of Newcastle upon Tyne.

Philips, L. (1990). Hanging on the metaphone. *Computer Language*, *7*(12), 39-43.

Philips, L. (2000). The double metaphone search algorithm. *C/C++ Users Journal*, *18*(6).

Porter, E. H., & Winkler, W. E. (1997). Approximate string comparison and its effect on an advanced record linkage system. In W. Alvey & B. Jamerson (Eds.), *Record linkage techniques: Proceedings of an international workshop and exposition.* (p. 190-199). Arlington, VA.: Office of Management and Budget.

Taft, R. L. (1970). *Name searching techniques.* Albany, N.Y.: Bureau of Systems Development.

Ukkonen, E. (1985). Algorithms for approximate string matching. *Information and Control*, *64*(1-3), 100-118.

Ukkonen, E. (1992). Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science*, *92*(1), 191-211.