Record Linkage

Introduction to Big Data for Social Science

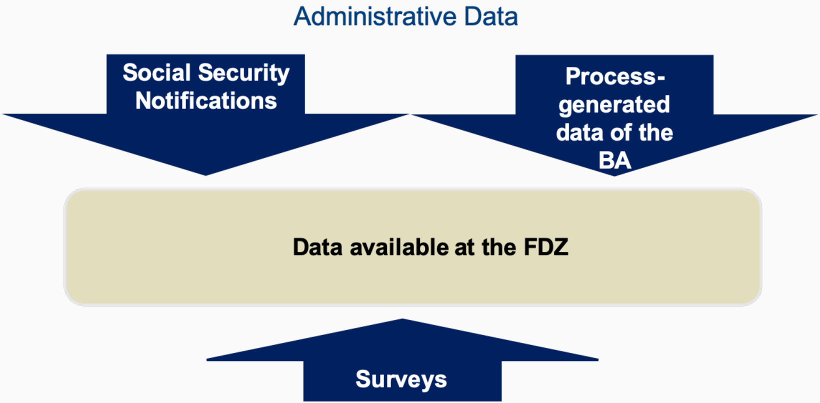
Frauke Kreuter[1] ...

June 3–4, 2019

[1] fkreuter@umd.edu

The linkage of Admin Data

- German Micro labor market data on individuals/households and establishments (internal data)


Administrative Data
Social Security Notifications
Process-generated data of the BA
Data available at the FDZ
Surveys

Linking Admin Data II

- Linking survey and administrative data is becoming increasingly common
- Patent data:
  - Name and address of inventors for all registered patents in from 1990-2012
- Bureau van Dijk (Company information)
- Geocoded data

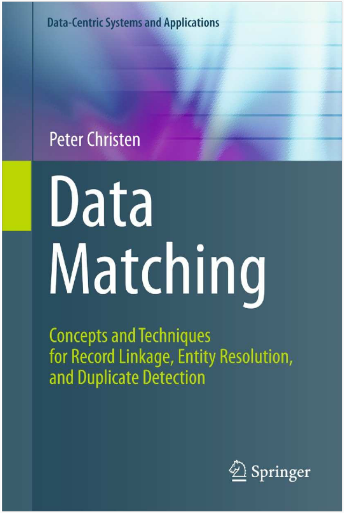
Summing Up (Christen 2015)

- Large amounts of data are being collected (big data).
- Analyzing such data can provide huge benefits.
- Data are from different sources (need for record linkage).
- Lack of unique entity identifiers: linking based on personal information.
- The linking of databases is challenged by data quality, database size, privacy and confidentiality concerns.

Take away

- Introduction to record linkage.
- Learn about potential pitfalls.
- (Practical examples.)
- Enable participants to assess the feasibility of, plan and manage record linkage projects as well as to perform each step along the actual linkage process.

Source for a Deeper Knowlege



Data-Centric Systems and Applications

Peter Christen

Data Matching

Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection

Springer

Table of contents

1. Introduction to Record Linkage (RL)

2. Identifiers

3. Preprocessing

4. Increasing the Efficiency of the Matching Step (Blocking)

5. Alternative Ways of Conducting the Matching Step (String Comparators)

6. Probabilistic RL

7. Evaluation

8. Advanced Classification Techniques

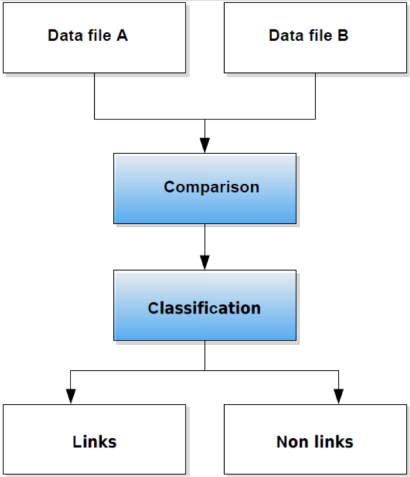
9. RL Software

Introduction to Record Linkage (RL)

Definition of Record Linkage and Major Challenges

- RL is finding records in different data sets that represent the same entity and link them.
- RL is also known as *data matching, entity resolution, object identification, duplicate detection, identity uncertainty, merge-purge.*
- Major challenge is that (clean) unique entity identifiers are not available in the databases to be linked.

The basic record linkage process



5 Main Applications of Record Linkage

1. Merging of two or more data files
2. Identifying the intersection of the two data sets
3. Updating of data files (with the data row of the other data files)
4. Impute missing data
5. Deduplicate a file (remove duplicates in one file)

1. Merging of two data files

- Merging of data files for microanalyses (e.g. survey- or registry data)
- Follow - up of cohorts (e.g. linkage with Cancer registry)
- Retrospective construction of panels
- Merging of panel waves
- Validation of answers in surveys: Comparing individual provided information’s with registry data.
- Bias ff detection in surveys: Conveyance of data for nonrespondents.
- Conveyance of external data to survey data for imputation of weighting.
- Adding contact information’s to survey-samples.

2. Identifying the intersection of the two data files

- Discovery of undercoverage within a census.
- Estimation of population size through capture-recapture.
- Discovery of overcoverage and undercoverage in sampling frames.
- Examination of the reidentification risk of micro data files.
- Discovery of underreporting in registries (e.g. linkage with mortality registry).
- Dropping of duplicates as part of data cleansing.

3. Update of a data file

- Update of sampling frames.
- Update of registries (e.g. new registrations in the cancer registry).

Record Linkage Technique (Christen 2015)

- **Deterministic matching**
  - Rule-based matching (complex to build and maintain)
- **Probabilistic record linkage** (*Fellegi and Sunter, 1969*)
  - Use available attributes for linking (often personal information, like names, addresses, dates of birth, etc.)
  - Calculate match weights for attributes
- **“Computer science” approaches**
  - Based on machine learning, data mining, database, or information retrieval techniques
  - Supervised: Requires training data (true matches)
  - Unsupervised: Clustering, collective, and graph based

Machine Learning, Information Retrieval and RL (Winkler 2000)

- Machine Learning: text classification is used for classifying documents into categories.
- Information Retrieval: the text might be queries related to subjects that are used for a library or an internet search.
- Record Linkage: the categories might simply be the determination that a pair of records from two lists represents the same entity (is a match) or is not the same entity (non-match).

Machine Learning, Information Retrieval and RL (Winkler 2000)

- RL typically has more structured information. Name and address parsing and standardization software puts person names and addresses into specific locations.
- Because of the additional structure of knowing what words to compare (street with street) RL has not always needed training data. Guesses can sometimes yield suitable decision rules.
- Fellegi and Sunter (1969) provided a formal mathematical model for record linkage.

A short history of record linkage (Christen 2015)

- Computer assisted record linkage goes back as far as the 1950s (based on ad-hoc heuristic methods)
- Basic ideas of probabilistic linkage were introduced by *Newcombe & Kennedy (1962)*
- Theoretical foundation by *Fellegi & Sunter (1969)* Compare common record attributes (or fields)
    - Compute matching weights based on frequency ratios (global or value specific) and error estimates
    - Sum of the matching weights is used to classify a pair of records as a match, non-match, or potential match
    - Problems: Estimating errors and thresholds, assumption of independence, and clerical review

The Fellegi-Sunter Approach: General

Every pair of records is compared and represented using a vector of components that describe similarity between individual record fields

- E.g., ffname agreesff, ffname disagreesff, ffname missing on one or both recordsff

The Fellegi-Sunter Approach: CIA

- Conditional independence assumption (CIA): given a pair of records representing the same entity (true match), we assume that agreement in each field is independent of agreement in other fields.
- CIA is a mathematical convenience only.
- In reality, record fields wonfft be conditionally independent. Improve match discrimination by eliminating covarying fields (more fields not always better matching).
    - Area codes may covary with geography
    - First name may covary with sex

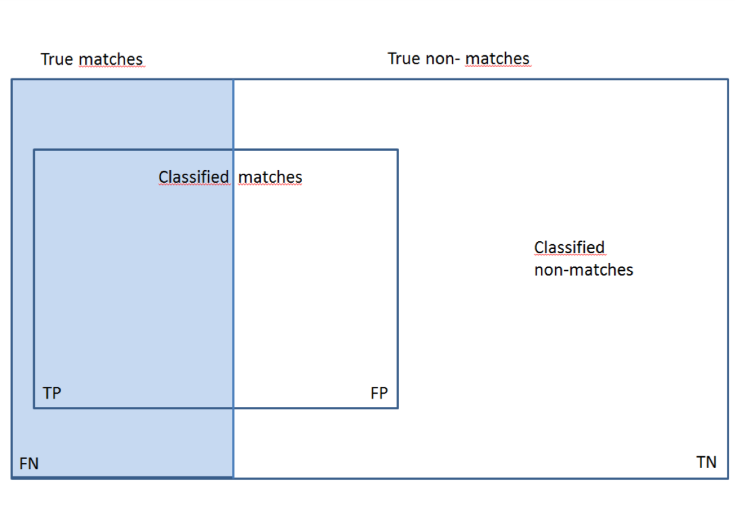
There is no perfect world

- In a perfect world

True Positive (TP)	
	True Negative (TN)

- But we do not live in a perfect world

True Positive (TP)	False Positive (FP)
False Positive (FP)	True Negative (TN)

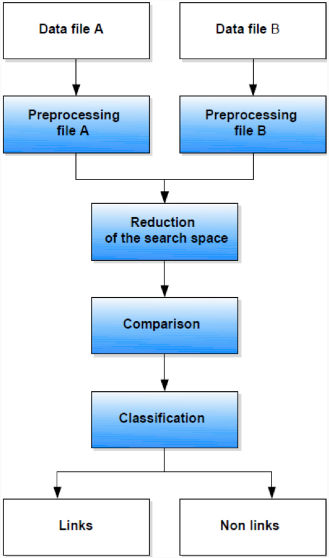


Source: Christen 2012

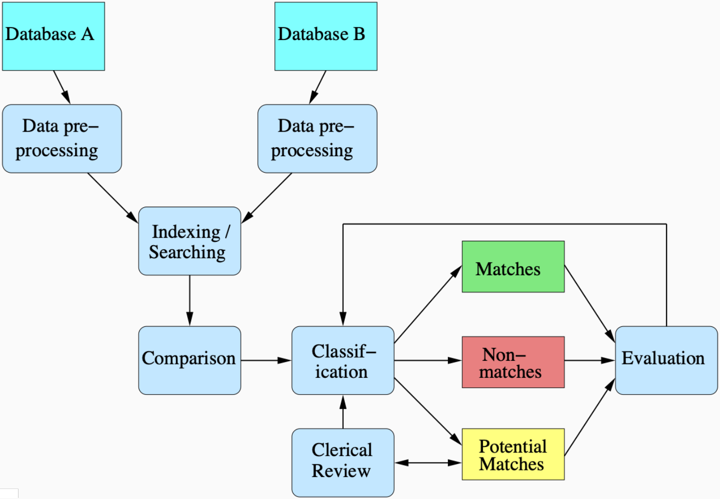
Record Linkage Challenges (Christen 2012)

- No unique entity identifiers available
- Real world data are dirty (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability
  - Naive comparison of all record pairs is quadratic
  - Remove likely no-matches as efficiently as possible
- No training data in many linkage applications
  - No record pairs with known true match status
- Privacy and confidentiality
  - (because personal information, like names and addresses, are commonly required for linking)

The extended record linkage process



The extended record linkage process



Source: Christen 2015

Caveats of Record Linkage

- Imperfect matching variables (like typos)
- Variables may be coded differently in both data sources
    - E.g., years of education vs. degrees received
- Data may require significant amounts of processing and data cleaning prior to linkage
- Not always a 1-to-1 match, but a 1-to-1 matched set can be extracted from Fellegi-Sunter output in a post-processing step (typically, by treating it as a linear sum assignment problem).
- (admin) record may not exist.

Identifiers

Identifiers

- Typical identifiers:
  - People: first and last name, address, birth date, sex
  - Establishments / firms: name, legal form, address
- The higher the number of different manifestations of an identifier, the better its suitability for a comparison.
- Complex identifiers should be parsed into its separate components
- Means of getting clean identifiers in the first place

Benefits of variation in identifiers

- Variations within a given unit possible in almost every variable
- Variation can arise almost everywhere
- A lot of reasons (like marriage with change of name, nickname)
- Every characteristic can be the one, which is also a part of the other data set.

→ Always keep all available variations and apply them!

Preprocessing

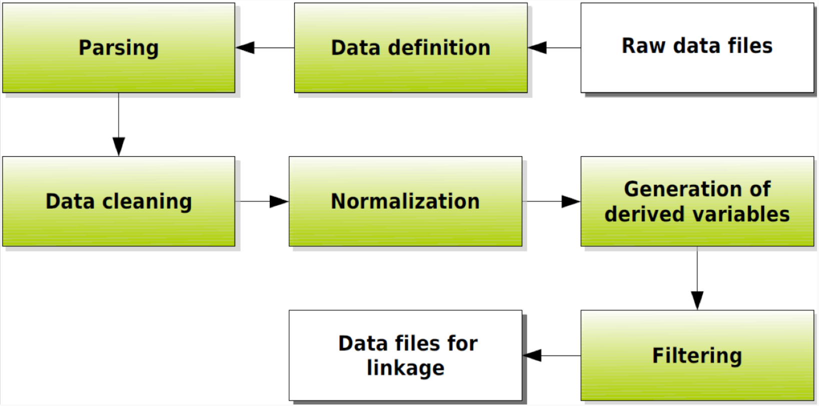
Importance of Preprocessing

- “In situations of reasonably high-quality data, preprocessing can yield a greater improvement in matching efficiency than string comparators and ffoptimized parametersff. In some situations, 90% of the improvement in matching efficiency may be due to preprocessing.” (Winkler 2009, p. 370)
- Inability or lack of time and resources for cleaning up files in preparation of matching are often the main reasons that matching projects fail.” (Winkler 2009, p. 366)

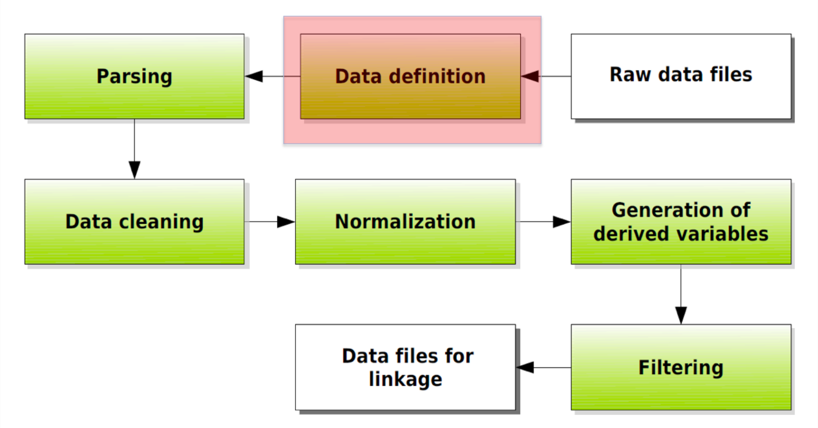
Shares of effort within linkage process

- 5% matching and linking efforts
- 20% checking that the computer matching is correct
- 75% cleaning and parsing the two input files
- (see Gill 2001, p. 31)

Preprocessing: Workflow



Preprocessing: Workflow



Creating a data definition

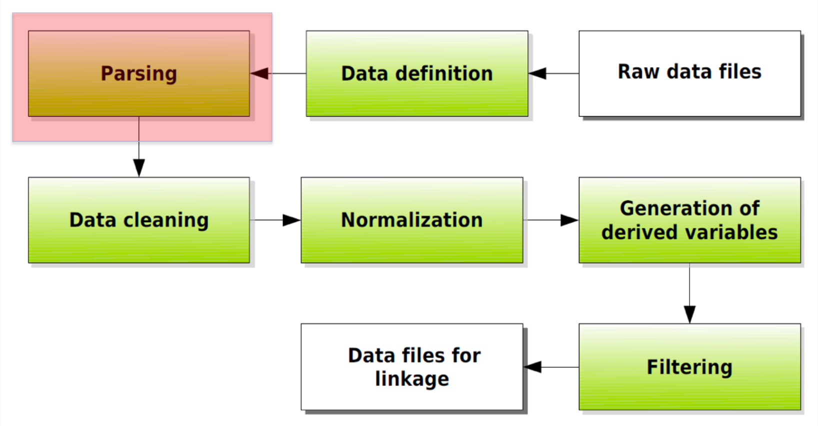
A data definition records attributes for each identifier that are assigned to them conceptionally. These attributes should encompass:

1. In general: Variable name, variable type, data type, missing code
2. For continuous variables: Allowed range
3. For coded categorical variables: Code list
4. For uncoded variables (respectively name variables)
    - regular length
    - regular pattern
    - allowed character set
    - excluded rules
    - is a list of permissible values available?

Example for a data definition 1: sex

1. Variable name: sex
2. Variable type: categorical, coded
3. Data type: byte
4. Code list: 1 male 2 female 3 not determinable 9 missing

Preprocessing: Workflow



Raw data files

Data definition

Parsing

Data cleaning

Normalization

Generation of derived variables

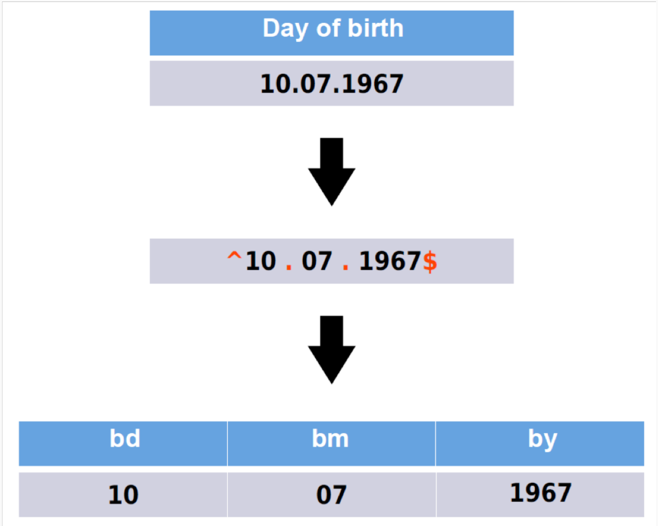
Filtering

Data files for linkage

Parsing

- Parsing is the decomposition of a complex variable into single components.
- Subsequently, the single components can be composed to a standard form or can be used as single match variables.
- In simple cases the decomposition takes place through delimiter or through simple regular expressions.
    - Example: field with zip code and place name
- For more complex fields or fields with heterogeneous representations of their values, specific parsing routines are necessary.

Simple parsing



Day of birth
10.07.1967

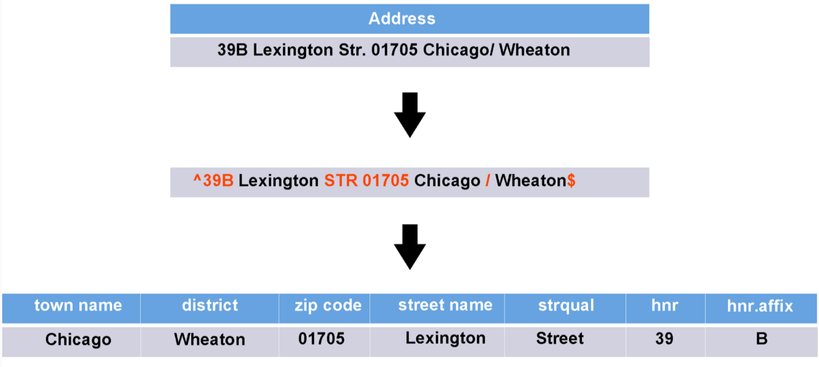
^10 . 07 . 1967$

bd	bm	by
10	07	1967

Complex parsing

- Decomposition into predefined components using predefined rules
- Typical procedure:
    1. Splitting fields into tokens (words) on basis of delimiters
    2. Standardization of tokens by lookup tables and substitution by a standard form.
    3. Categorization of tokens
    4. Identification of pattern of anchors, tokens and delimiters
    5. Calling subroutines according to the identified pattern, therein mapping of tokens to the predefined components.

Example: Parsing of addresses



Address
39B Lexington Str. 01705 Chicago/ Wheaton

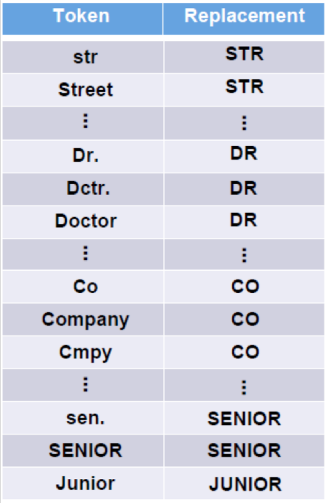
^39B Lexington STR 01705 Chicago / Wheaton$

town name	district	zip code	street name	strqual	hnr	hnr.affix
Chicago	Wheaton	01705	Lexington	Street	39	B

Parsing freeform addresses is a hard problem! One approach uses hidden Markov models.

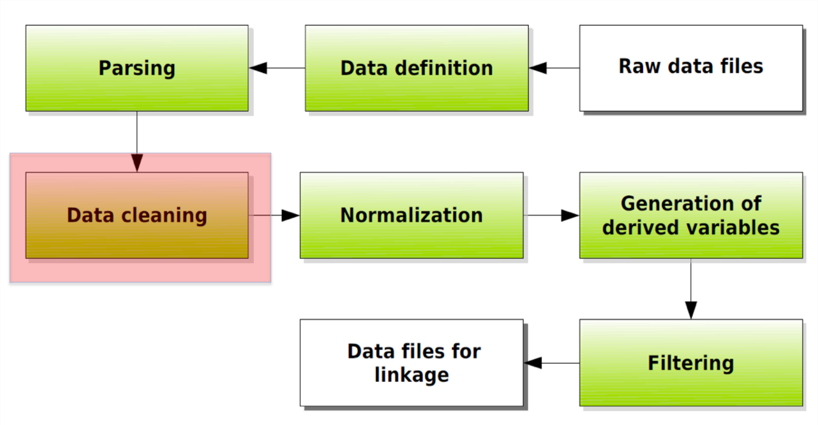
Lookup tables for standardization

Typical are tables for tokens in establishment names, personal names and addresses.



Token	Replacement
str	STR
Street	STR
⋮	⋮
Dr.	DR
Dctr.	DR
Doctor	DR
⋮	⋮
Co	CO
Company	CO
Cmpy	CO
⋮	⋮
sen.	SENIOR
SENIOR	SENIOR
Junior	JUNIOR

Preprocessing: Workflow



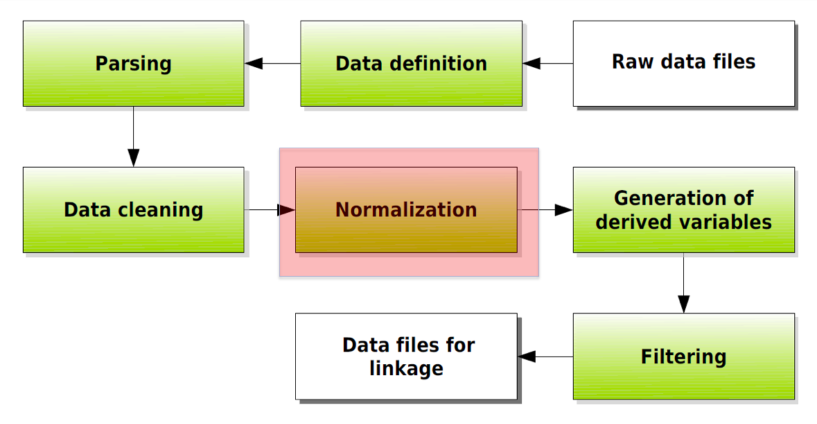
Data cleaning: Overview

1. Evaluation of identifiers against data definition
2. Checking plausibility of variable values
3. Checking records for consistency
4. Standardization
5. Deduplication

Example (Christen 2015)

- Raw input address: '42 meyer Rd COOMA 2371'
- Cleaned into: ff42 meyer road cooma 2371ff
- Tagged: (both look-up tables and feature tags)
- (ff42ff,ffmeyerff,ffroadff, ffcoomaff, ff2371ff)
- (ffN2ff,ffSN/L5ff,ffST/L4ff,ffLN/SN/L5ff,ffPC/N4ff)
- Segmented into *output fields:*
    - number_first : ff42ff
    - street_name : ffmeyerff
    - street_type : ffroadff
    - locality_name : ffcoomaff
    - postcode : ff2371ff

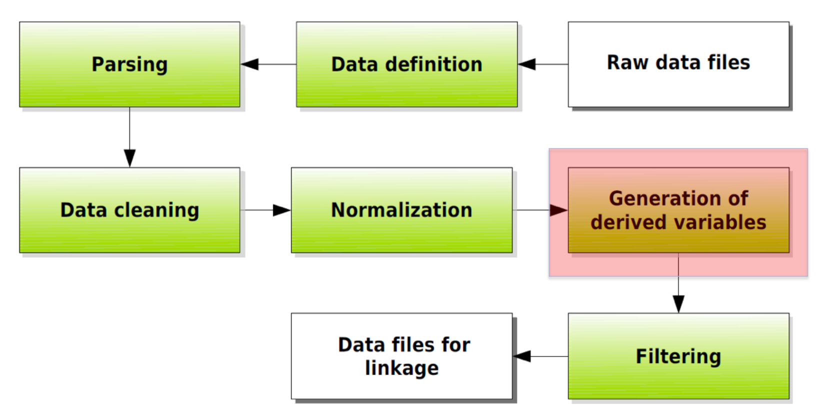
Preprocessing: Workflow



Normalization

- Normalization refers to the harmonization of input files for the actual linkage.
- Examine and harmonize variable pairs that will be compared afterwards.
- Common checks: from data types to standardization.

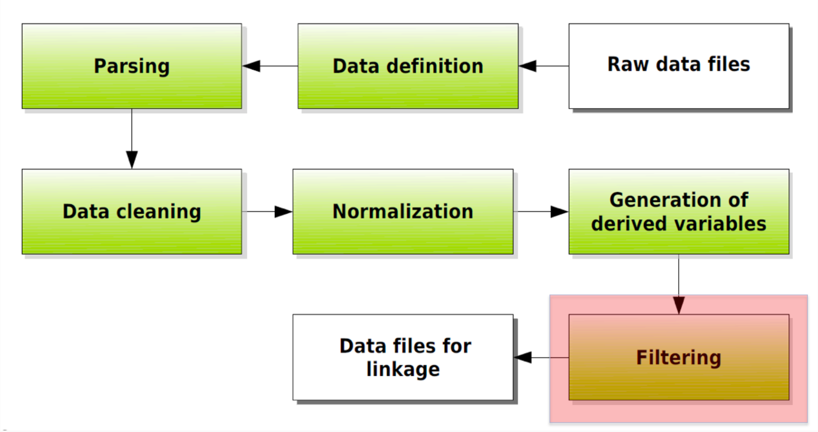
Preprocessing: Workflow



Generating of derived variables

- Usually to get appropriate blocking variables
- Typically over-standardized variants of existing identifiers
- Examples:
  - Phonetic codes of first and surname
  - Initial letters of first and surname
  - Truncation of zip codes to 3 or 4 digits

Preprocessing: Workflow



Filtering

- Removal of data rows that have no or irrelevant counterparts in the other input file.
- They would lead to unnecessary comparisons.

Statistics New Zealand: Standard preprocessing of surname

- Take surname
- Capitalize
- Remove spaces
- Set to missing if surname contains ffunknownff
- Remove any characters other than alphabetic characters
- Name the resulting field surname1
- Define new variable initial_surname = first character of surname1
- Define new variable soundex_surname = Soundex code of surname1 (See Statistics of New Zealand 2006, p.50)

Preprocessing: main results (I)

- Preprocessing is always specific for the concrete application.
    - Example: Establishment vs. individual data
- Expenditure of time for preprocessing often exceeds efforts of the record linkage (comparison, classification).
- Especially with bad data quality preprocessing is the most important factor for the success of linkage projects.
- Budget enough resources for preprocessing.

Preprocessing: main results (II)

- Neither is there a universally suitable software for this, nor is there a comprehensive textbook.
- In practice a program for data analysis or script languages like AWK, Perl or Python are used.
- The software has to be capable to do searches with regular expressions.
- Common statistics software can be alienated:
  - R and SAS allow the functionality of Perl
  - Stata offers his own implementations, but strongly limited range of functions towards real script languages (see http://www.stata.com/support/faqs/data-management/regular-expressions)

Increasing the Efficiency of the Matching Step (Blocking)

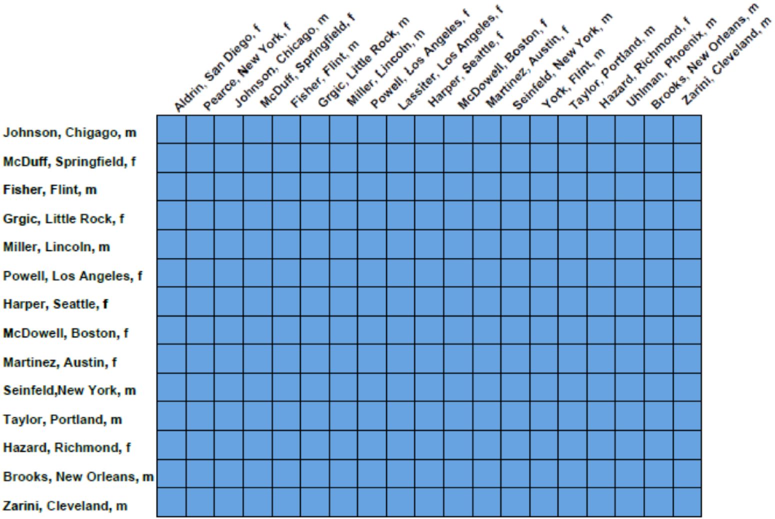
The Efficiency Problem

- With n records in file A and m records in file B, n x m pairs have to be compared.
- 100 000 x 100 000 = 10 000 000 000 (10 billion) comparisons
- With 10 000 comparisons per second this takes 278 hours or 11.6 days

Standard Technique: Traditional Blocking

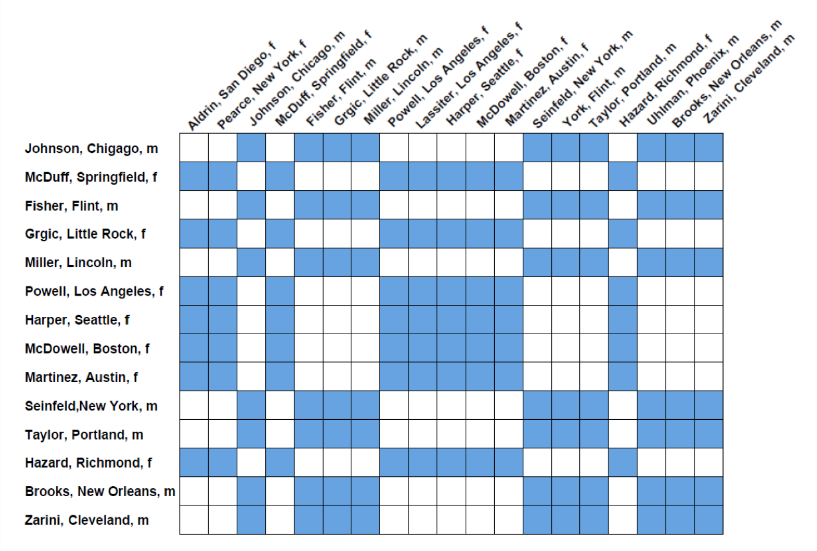
- According to its values, a variable partitions both data files into subsets, called blocks or pockets.
- The A- and the B-file are partitioned using the same (blocking) variable.
- Only pairs of records belonging to the same block within a certain file are compared.
- Advanced methods are error-tolerant.
    - Challenge: Blocking saves resources, but could potentially cost true positives.

Example: No blocking



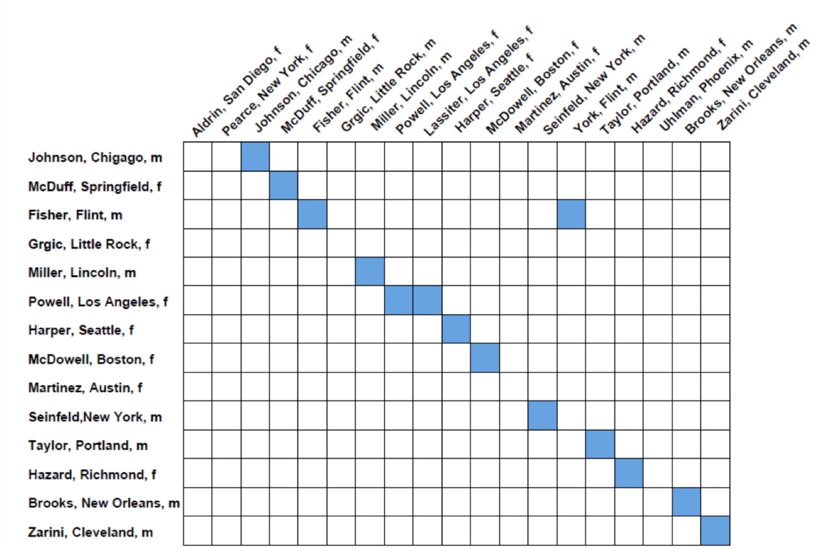
	Aldrin, San Diego, f	Pearce, New York, f	Johnson, Chicago, m	McDuff, Springfield, f	Fisher, Flint, m	Grgic, Little Rock, m	Miller, Lincoln, m	Powell, Los Angeles, f	Lassiter, Los Angeles, f	Harper, Seattle, f	McDowell, Boston, f	Martinez, Austin, f	Seinfeld, New York, m	York, Flint, m	Taylor, Portland, m	Hazard, Richmond, f	Uhlman, Phoenix, m	Brooks, New Orleans, m	Zarini, Cleveland, m
Johnson, Chigago, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
McDuff, Springfield, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Fisher, Flint, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Grgic, Little Rock, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Miller, Lincoln, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Powell, Los Angeles, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Harper, Seattle, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
McDowell, Boston, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Martinez, Austin, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Seinfeld,New York, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Taylor, Portland, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Hazard, Richmond, f	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Brooks, New Orleans, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
Zarini, Cleveland, m	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Blocking by sex



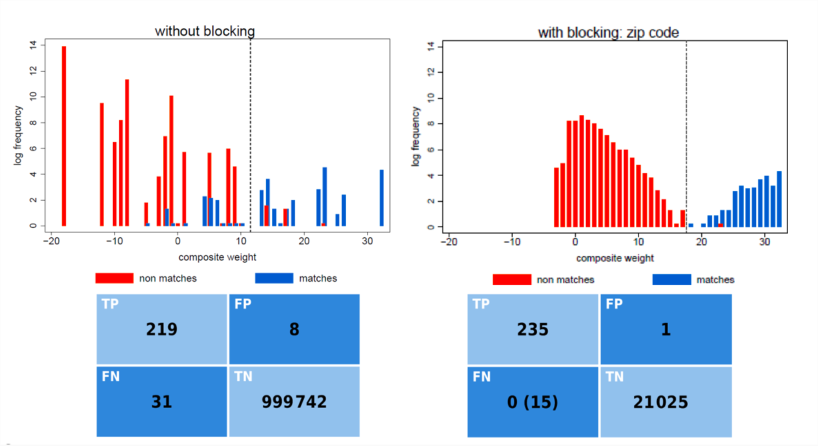
	Aldrin, San Diego, f	Pearce, New York, f	Johnson, Chicago, m	McDuff, Springfield, f	Fisher, Flint, m	Grgic, Little Rock, m	Miller, Lincoln, m	Powell, Los Angeles, f	Lassiter, Los Angeles, f	Harper, Seattle, f	McDowell, Boston, f	Martinez, Austin, f	Seinfeld, New York, m	York, Flint, m	Taylor, Portland, m	Hazard, Richmond, f	Uhlman, Phoenix, m	Brooks, New Orleans, m	Zarini, Cleveland, m
Johnson, Chigago, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
McDuff, Springfield, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Fisher, Flint, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
Grgic, Little Rock, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Miller, Lincoln, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
Powell, Los Angeles, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Harper, Seattle, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
McDowell, Boston, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Martinez, Austin, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Seinfeld,New York, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
Taylor, Portland, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
Hazard, Richmond, f	🟦	🟦		🟦				🟦	🟦	🟦	🟦	🟦				🟦			
Brooks, New Orleans, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦
Zarini, Cleveland, m			🟦		🟦	🟦	🟦						🟦	🟦	🟦		🟦	🟦	🟦

Blocking by sex and location



	Aldrin, San Diego, f	Pearce, New York, f	Johnson, Chicago, m	McDuff, Springfield, f	Fisher, Flint, m	Grgic, Little Rock, m	Miller, Lincoln, m	Powell, Los Angeles, f	Lassiter, Los Angeles, f	Harper, Seattle, f	McDowell, Boston, f	Martinez, Austin, f	Seinfeld, New York, m	York, Flint, m	Taylor, Portland, m	Hazard, Richmond, f	Uhlman, Phoenix, m	Brooks, New Orleans, m	Zarini, Cleveland, m
Johnson, Chigago, m			■																
McDuff, Springfield, f				■															
Fisher, Flint, m					■									■					
Grgic, Little Rock, f																			
Miller, Lincoln, m							■												
Powell, Los Angeles, f								■	■										
Harper, Seattle, f										■									
McDowell, Boston, f											■								
Martinez, Austin, f																			
Seinfeld,New York, m													■						
Taylor, Portland, m															■				
Hazard, Richmond, f																■			
Brooks, New Orleans, m																		■	
Zarini, Cleveland, m																			■

Effect of blocking



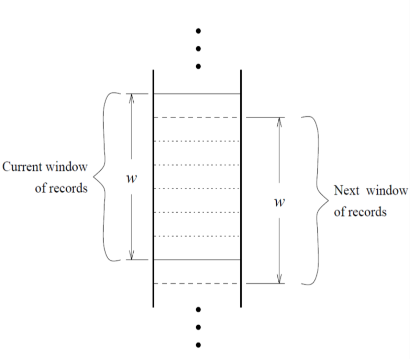
TP 219	FP 8
FN 31	TN 999742

TP 235	FP 1
FN 0 (15)	TN 21025

Traditional Blocking (Christen 2015)

- Traditional blocking works by only comparing record pairs that have the same value for a *blocking* variable (same *postcode* value)
- Problems with traditional blocking
  - An erroneous value in a blocking variable results in a record being inserted into the wrong block (several *passes* with different blocking variables can solve this)
  - Values of blocking variable should have uniform frequencies (as the most frequent values determine the size of the largest blocks) Example: Frequency of *ffSmithff vs. ffBenderff*

Error-tolerant methods 1: Sorted Neighbourhood



- All records of both files written in one list.
- Records are sorted in accordance to a sorting key (e.g. surname).
- A window of the width w is moved one record further over the sorted records.
- A comparison only takes place for those records within the same window.
- As in traditional blocking, multiple runs using different sorting keys should be carried out.
- (see Hernndez, Stolfo 1998)

Alternative Ways of Conducting the Matching Step (String Comparators)

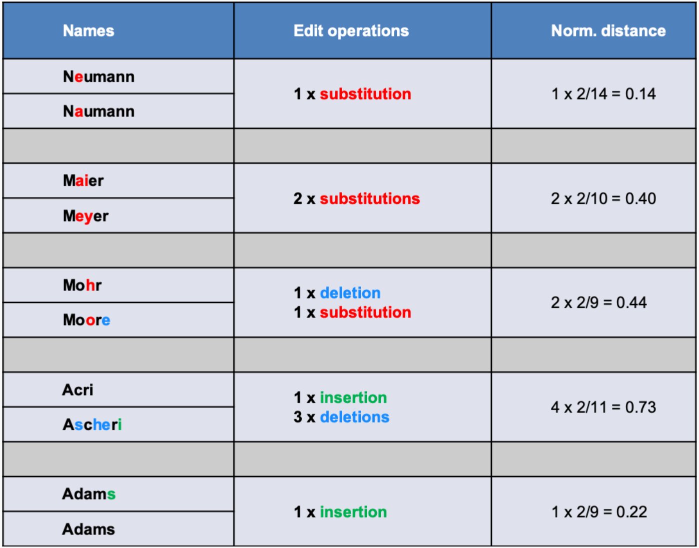
String similarities

- Function of a pair of character strings with similarity as function value.
- Common: Standardization of the function value to the interval [0-1] (0: no agreement; 1: complete agreement).
- Variations of the following classifications of string similarity functions are commonly used:
  - Phonetics
  - Edit-distances
  - n-grams
  - Jaro’s string comparator

Edit-distances: Principle

- An edit-distance between two strings a and b is the lowest number of permitted edit-operations needed to transfer a to b
- A certain edit-distance variant is defined by the set of permitted operations.
- For the Levenshtein-distance, for example insertions, deletions and substitutions are allowed
- Common: Normalization using the sum of the length of the strings
- Similarities are obtained by 1 - LDnorm

Levenshtein-distance



Names	Edit operations	Norm. distance
Neumann	1 x substitution	1 x 2/14 = 0.14
Naumann		
Maier	2 x substitutions	2 x 2/10 = 0.40
Meyer		
Mohr	1 x deletion 1 x substitution	2 x 2/9 = 0.44
Moore		
Acri	1 x insertion 3 x deletions	4 x 2/11 = 0.73
Ascheri		
Adams	1 x insertion	1 x 2/9 = 0.22
Adams		

Probabilistic RL

Principles (I)

- Simple summing up of similarities cannot be optimal.
- Different identifiers differ in how strongly an agreement is indicative for a link.

Name	Sex	Residence	Date of birth
Tom McDonalds	m	Albuquerque	12/06/1966

Principles (II)

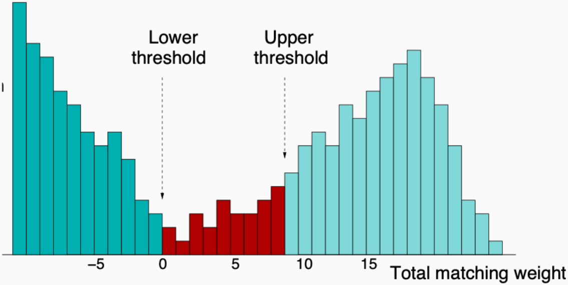
- Assigning appropriate weights to identifiers before summing up would be a better method.
- In order to weight identifiers it must be quantified for each identifier how strongly an agreement indicates a link.
- How likely is an agreement within the matches compared to within the non-matches?

Probabilistic Record Linkage (Christen 2015)

- Theoretical foundation by *Fellegi & Sunter, 1969*
- Compare common record attributes (or fields) using approximate (string) comparison functions
- Compute matching weights based on frequency ratios (global or value specific ratios) and error estimates
- Sum of the matching weights is used to classify a pair of records as *match, non-match*, or *potential match*
- Problems: Estimating errors, find optimal thresholds, assumption of independence, and manual clerical review

- For each compared record pair a vector containing *matching weights* is calculated
  - Record A: [ffdrff, ffpeterff, ffpaulff, ffmillerff]
  - Record B: [ffmrff, ffjohnff, ff ff , ffmillerff]
  - Matching weights: [0.2, -3.2, 0.0, 2.4]
- Sum weights in vector, and use two thresholds to classify record pairs as *matches*, *non-matches*, or *potential matches*

Many more with lower weights...



Evaluation

Evaluation (Christen 2015)

- To measure linkage quality, we need the true matches (gold standard, ground truth data)
    - Two types of errors:
        - A missed true match (false non-match, *false negative*)
        - A wrong match (false match, *false positive*)
- Record linkage is a very *imbalanced* problem
    - Most records pairs (even after blocking) are true non-matches
- Calculating *accuracy* (percentage of false matches and false non-matches) is not meaningful
  (classifying all record pairs as non-matches can give very high accuracy)

Advanced Classification Techniques

“Computer Science” Approaches

- Based on machine learning, data mining, database, or information retrieval techniques
  - Supervised: Requires training data (true matches)
  - Unsupervised: Clustering, collective, and graph based

Advanced Classification Techniques (Christen 2015)

- View record pair classification as a *multi- dimensional binary classification problem*
  (use *attribute similarities* to classify record pairs as *matches* or *non-matches* – donfft sum into one similarity)
- Many machine learning techniques can be used
  - Supervised: Requires training data (record pairs with known true match and non-match status)
  - Different supervised techniques have been used: Decision trees, support vector machines, neural networks, learnable string comparisons, etc.
  - Active and semi-supervised learning
  - Unsupervised: Clustering

Classification Challenges (Christen 2015)

- In many cases there are no training data available
  - Possible to use results of earlier matching projects?
  - Or from manual *clerical review* process?
  - How confident can we be about correct manual classification of *potential matches*?
- Often there is no *gold standard* available (no data sets with true known match status)
- No large test data set collection available (like in information retrieval or machine learning)
- Recently, *collective* classification techniques have been investigated (also take *relational similarities* into account)

RL Software

Software overview

- Other (free) programs (see Appendix):
    - Big Match
    - GRLS
    - The Link King
    - Link Plus
    - FRIL
    - Open Refine
    - Relais
    - R-Paket RecordLinkage
    - TDGen

Freely Extensible Biomedical Record Linkage (FEBRL)

- Project “Parallel Large Scale Techniques for High-Performance record linkage”
- Australian National University (ANU), Department of Computer Science
- Peter Christen
- Project: datamining.anu.edu.au/projects/linkage.html
- Version 0.4.2, 2013
- Download: sourceforge.net/projects/febrl

FEBRL: Features

- Freely available and expandable (open source license): Python
- Preprocessing module
- Probabilistic record linkage
- Further classification techniques
- Different blocking techniques
- Many string similarity functions
- Geocoding
- Blindfolded/Privacy Preserving Record Linkage
- Frequency weights

MTB: Basics

- Merge ToolBox (MTB) is a Java application developed by the German RLC
- Current version: 0.742, November 2012
- Free use for academic purposes
- To be found at: http://record-linkage.de/-Downloads–software.htm
- Counseling by the German RLC

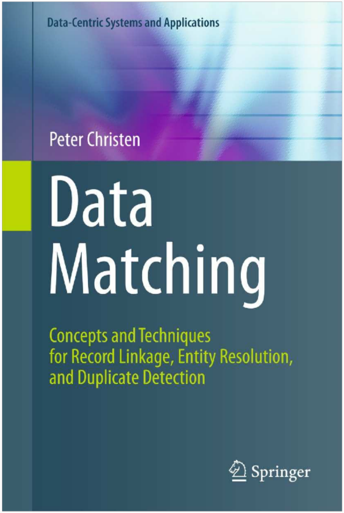
MTB: Features

- Probabilistic record linkage
- Many string similarity functions
- Several blocking techniques implemented
- Frequency weights
- 1-1 matching
- Parameter estimation using EM-algorithm
- Array-matching
- Privacy Preserving Record Linkage using Bloom Filters

MTB: Configuration via XML-file

- XML-configuration files allow replicable MTB runs.
- Particularly helpful during testing or if data files have to be divided for size-related reasons
- In the batch-mode configurations can be run successively and automatically.
- After initially creating a configuration, (copies of) the XML-file can be adapted with external editors (e.g. Emacs, Notepad++, WinEdt)

Source for a Deeper Knowlege



Data-Centric Systems and Applications

Peter Christen

Data Matching

Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection

Springer

References

Christen, P. (2012). Data Matching Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.

Christen, P. (2015). ...

Fellegi, van P. & Sunter, Alan B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, 64:328, 1183-1210.

Gill, ... (2001). ...

Hernndez, M. A. & Stolfo, S. J. (1998). Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *Data Mining and Knowledge Discovery* (2) 9-37.

Newcombe, H. B. & Kennedy, J. M. (1962). Record linkage: making maximum use of the discriminating power of identifying information. *Communications of the ACM* (Volume 5 Issue 11, Nov. 1962) 563-566.

References

Winkler, W. E. (2000). Machine learning, information retrieval, and record linkage. *Proc Section on Survey Research Methods, American Statistical Association*, 20-29.

Winkler, W. E. (2009). Record Linkage. in: Pfeffermann. D. & Rao, C. R. (eds.) *Sample Surveys: Theory, Methods and Inference* 351-380.