Data Linkage – An Overview and Research at the ANU

Peter Christen

Department (School) of Computer Science,
ANU College of Engineering and Computer Science,
The Australian National University,
Canberra, ACT 0200

Contact: **peter.christen@anu.edu.au**

Project Web site: **http://datamining.anu.edu.au/linkage.html**

Outline

- Short introduction to data linkage
  - Applications and challenges
  - The linkage process and linkage techniques
  - Recent developments in data linkage research
- Data linkage research at the ANU
  - The *Febrl* project
  - Linking historical census data (collaboration with the ANU Australian Demographic and Social Research Institute)
  - Linking bibliographic data (collaboration with the ANU Research Office)
- Outlook

Short introduction to data linkage

- The process of linking/matching records from one or more data sources that represent the same entity (such as a patient, customer, publication, etc.)
  - Also called *data matching*, *entity resolution*, *data scrubbing*, *object identification*, *merge-purge*, etc.
- Challenging if no unique entity identifiers available
  - For example, which of these three records refer to the same person?

Dr Smith, Peter	*42 Miller Street 2602 O'Connor*
Pete Smith	*42 Miller St, 2600 Canberra A.C.T.*
P. Smithers	*24 Mill Street; Canberra ACT 2600*

Recent interest in data linkage

- Traditionally, data linkage has been used in health (epidemiology) and statistics (census)
- In recent years, increased interest from businesses and governments
  - Increased computing power and storage capacities
  - A lot of data is being collected by many organisations
  - Data warehousing and distributed databases
  - Need for data sharing between organisations
  - Data mining of large data collections
  - E-Commerce and Web applications
  - Geocode matching and spatial data analysis

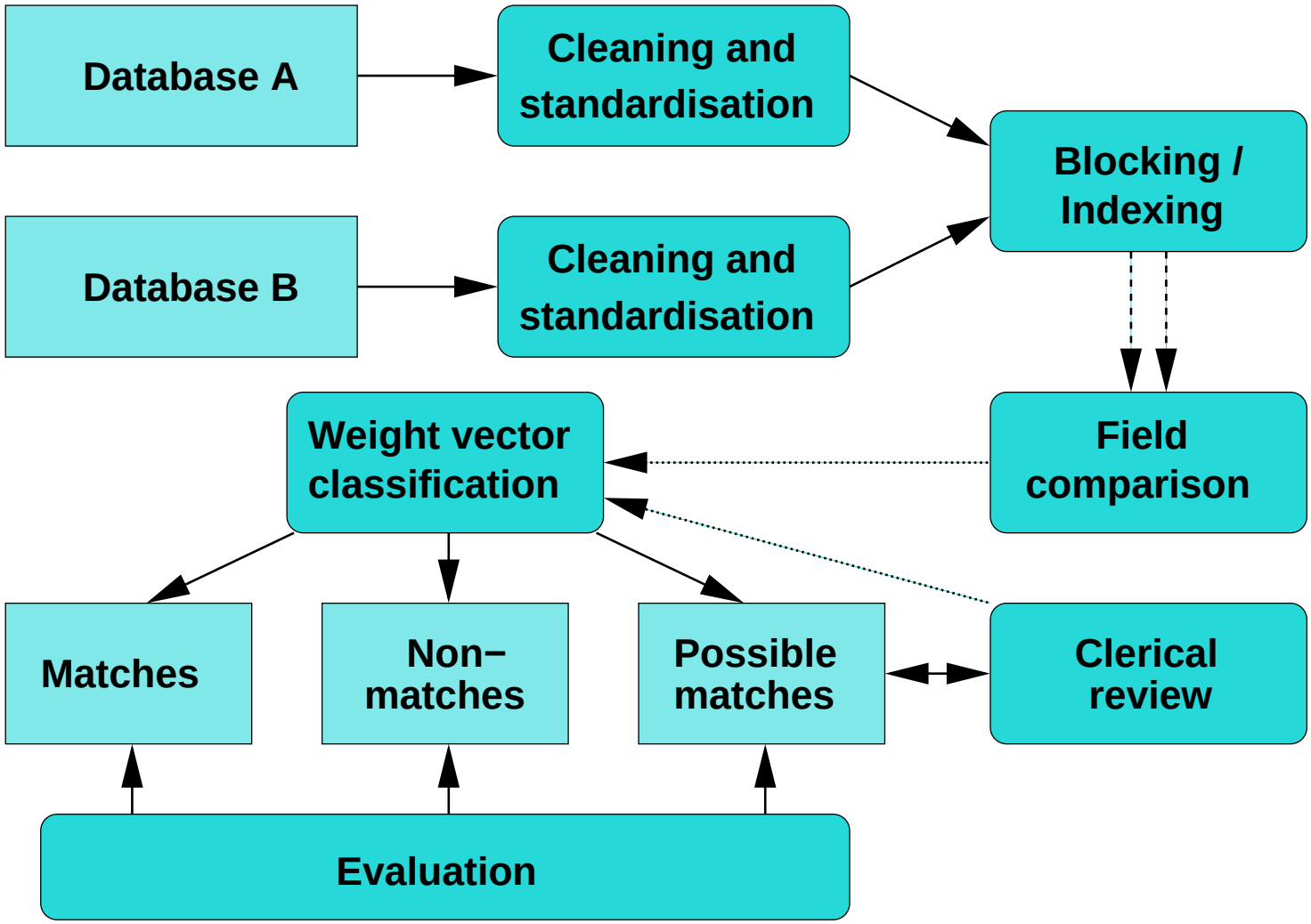
Applications of data linkage

- Remove duplicates in one data set (internal linkage)
- Merge new records into a larger master data set
- Create patient or customer oriented statistics
  (for example for longitudinal studies)
- Clean and enrich data for analysis and mining
- Geocode matching (with reference address data)
- Widespread use of data linkage
  - Immigration, taxation, social security, census
  - Fraud, crime and terrorism intelligence
  - Business mailing lists, exchange of customer data
  - Social, health and biomedical research

Data linkage challenges

- Often no unique entity identifiers are available
- Real world data is dirty
  (typographical errors and variations, missing and
  out-of-date values, different coding schemes, etc.)
- Scalability
  - Naïve comparison of all record pairs is $O(|A| \times |B|)$
  - Some form of blocking, indexing or filtering is required
- Privacy and confidentiality
  (because personal information, like names and addresses,
  are commonly required for linking)
- No training data in many linkage applications
  (no record pairs with known true match status)

The data linkage process



Data linkage techniques

- Deterministic linkage
  - Exact linkage (if a *unique identifier* of high quality is available: precise, robust, stable over time)
    Examples: *Medicare*, *ABN* or *Tax file* number (?)
  - Rules based linkage (complex to build and maintain)
- Probabilistic linkage (*Fellegi and Sunter*, 1969)
  - Use common attributes for linkage (often personal information, like names, addresses, dates of birth, etc.)
  - Can be wrong, missing, coded differently, or out-of-date
- Modern approaches

  (based on machine learning, data mining, AI, database, or information retrieval techniques)

Probabilistic data linkage

- Compare record pairs using the common attributes, calculate a *weight vector* of similarities

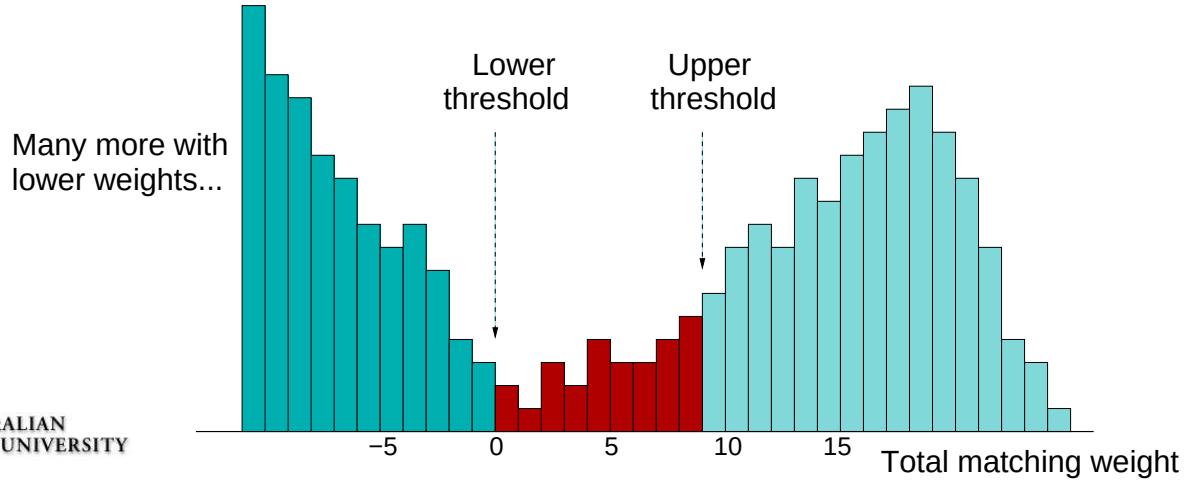
  Record A: `['dr', 'peter', 'paul', 'miller']`

  Record B: `['mr', 'pete', 'j.', 'miller']`

  Matching weights: `[0.2, 0.9, 0.0, 1.0 ]`

- *Fellegi and Sunter* approach sums all weights

  (then uses two thresholds to classify record pairs as *matches*, *non-matches*, or *possible matches*)



Blocking / indexing / filtering

- Number of record pair comparisons equals the product of the sizes of the two data sets
  (linking two data sets containing 1 and 5 million records will result in $1{,}000{,}000 \times 5{,}000{,}000 = 5 \times 10^{12}$ record pairs)
- Performance bottleneck in a data linkage system is usually the (expensive) detailed comparison of field values between record pairs
  (such as approximate string comparison functions)
- Blocking / indexing / filtering techniques are used to reduce the large amount of comparisons
- Aim of blocking: Cheaply remove candidate record pairs which are obviously not matches

Traditional blocking

- Traditional blocking works by only comparing record pairs that have the same value for a *blocking variable* (for example, only compare records that have the 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 be uniformly distributed (as the most frequent values determine the size of the largest blocks)
    Example: Frequency of *'Smith'* in NSW: *25,425*

Recent indexing approaches (1)

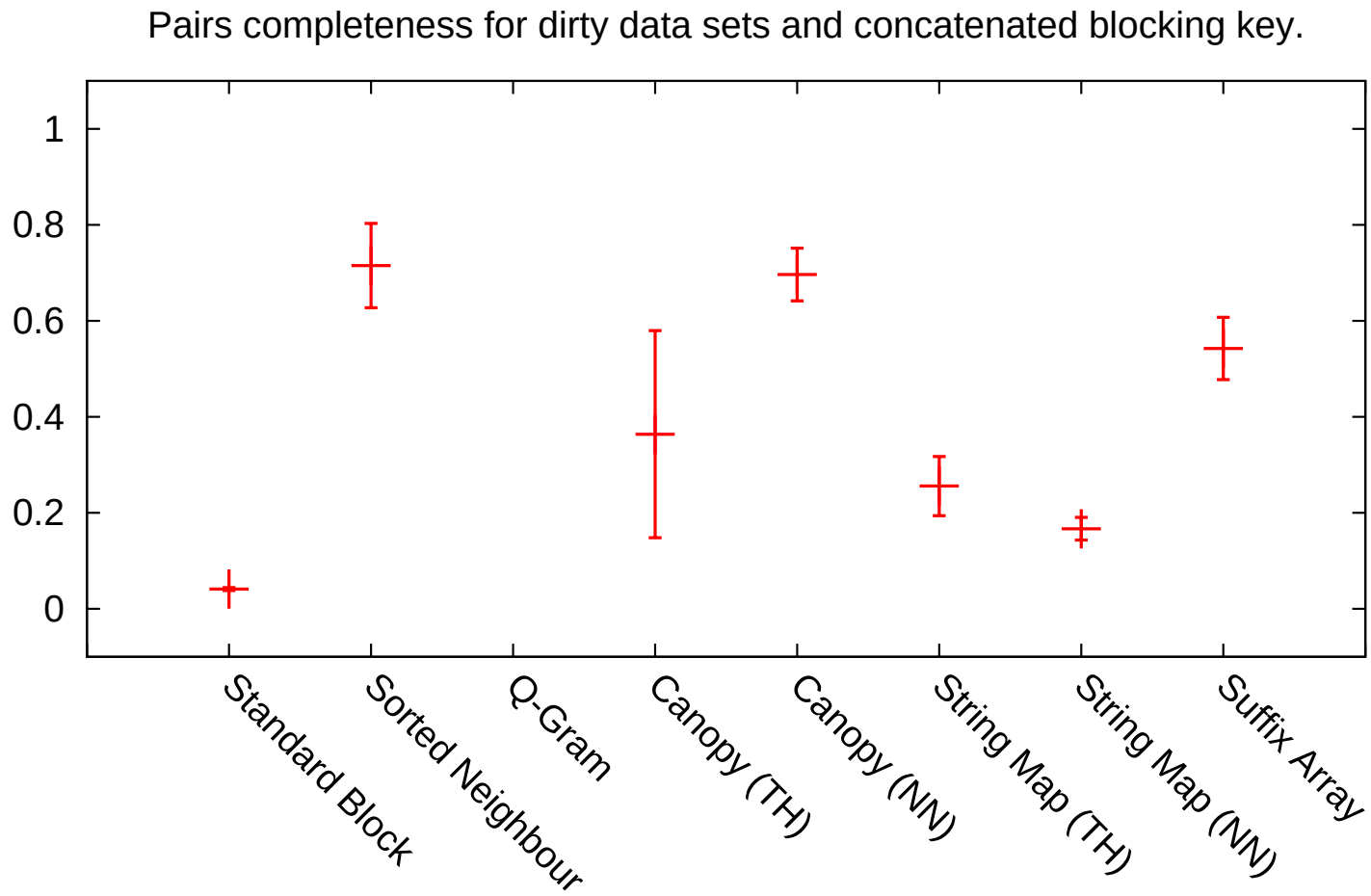
- Sorted neighbourhood approach
  - Sliding window over sorted blocking variable
  - Use several passes with different blocking variables
- *Q*-gram based blocking (e.g. *2-grams / bigrams*)
  - Convert values into *q*-gram lists, then generate sub-lists
    'peter' $\rightarrow$ *['pe','et','te','er']*, ***['pe','et','te']***, *['pe','et','er']*, ...
    'pete' $\rightarrow$ ***['pe','et','te']***, *['pe','et']*, *['pe','te']*, *['et','te']*, ...
  - Each record will be inserted into several blocks
- Overlapping *canopy* clustering
  - Based on *q*-grams and a 'cheap' similarity measure, such as Jaccard or TF-IDF/cosine

Recent indexing approaches (2)

- *StringMap* based blocking
  - Map strings into a multi-dimensional space such that distances between pairs of strings are preserved
  - Use similarity join to find similar pairs
- Suffix array based blocking
  - Generate suffix array based inverted index
    (suffix array: *'peter'* $\rightarrow$ *'eter'*, *'ter'*, *'er'*, *'r'*)
- Post-blocking filtering
  (for example, string length or *q*-grams count differences)
- US Census Bureau: *BigMatch*
  (pre-process 'smaller' data set so its values can be directly accessed; with all blocking passes in one go)

How good are recent approaches?

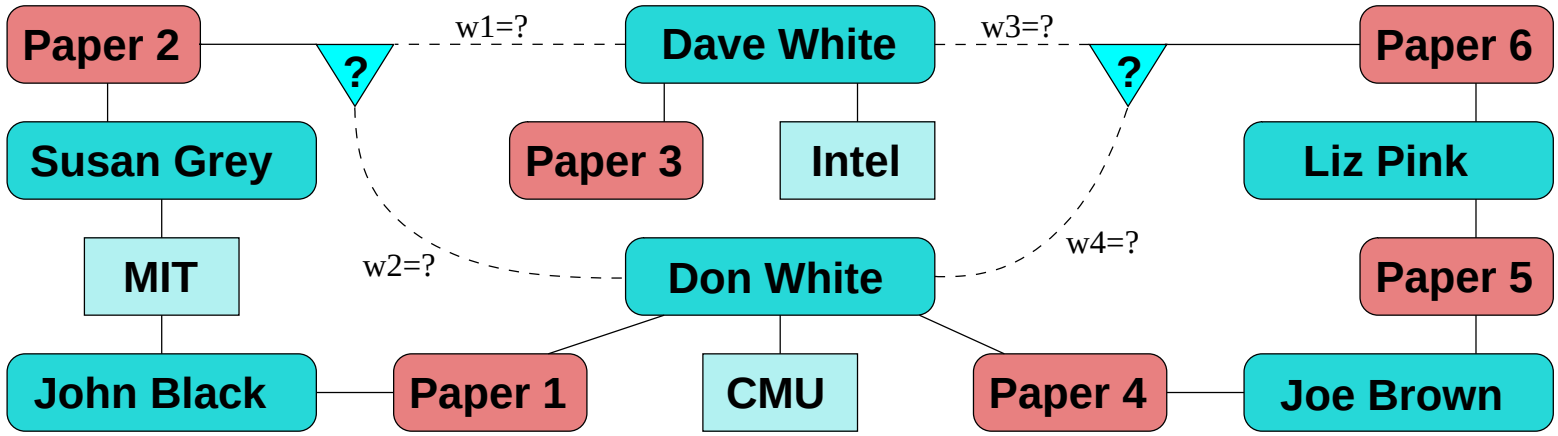
- No experimental comparisons of recent indexing techniques have so far been published



Improved record pair classification

- *Fellegi and Sunter* summing of weights results in loss of information
- View record pair classification as a *multi-dimensional binary classification* problem
  (use weight vectors to classify record pairs as *matches* or *non-matches*, but not *possible matches*)
- Many machine learning techniques can be used
  - Supervised: Decision trees, SVMs, neural networks, learnable string comparisons, active learning, etc.
  - Un-supervised: Various clustering algorithms
- Recently, *collective* entity resolution techniques have been investigated (rather than classifying each record pair independently)

Collective linkage example



(A1, Dave White, Intel)
(A2, Don White, CMU)
(A3, Susan Grey, MIT)
(A4, John Black, MIT)
(A5, Joe Brown, unknown)
(A6, Liz Pink, unknown)

(P1, John Black / Don White)
(P2, Sue Grey / **D. White**)
(P3, Dave White)
(P4, Don White / Joe Brown)
(P5, Joe Brown / Liz Pink)
(P6, Liz Pink / **D. White**)

Adapted from Kalashnikov and Mehrotra, ACM TODS, 31(2), 2006

Classification challenges

- In many cases there is no training data available
  - Possible to use results of earlier linkage projects?
    Or from manual *clerical review* process?
  - How confident can we be about correct manual classification of *possible links*?
- Often there is no *gold standard* available
  (no data sets with true known linkage status)
- No large test data set collection available
  (like in information retrieval or machine learning)
- Recent small repository: *RIDDLE*

  http://www.cs.utexas.edu/users/ml/riddle/

  (Repository of Information on Duplicate Detection, Record Linkage, and Identity Uncertainty)

Outline

- Short introduction to data linkage
  - Applications and challenges
  - The linkage process and linkage techniques
  - Recent developments in data linkage research
- Data linkage research at the ANU
  - The *Febrl* project
  - Linking historical census data (collaboration with the ANU Australian Demographic and Social Research Institute)
  - Linking bibliographic data (collaboration with the ANU Research Office)
- Outlook

The Febrl project

- A collaboration with the NSW Department of Health (ARC Linkage Project 2004–2008)
- Aim was to develop new and improved techniques for parallel large scale data linkage
- Main research areas
  - Probabilistic techniques for automated data cleaning and standardisation (mainly on addresses, using *G-NAF*)
  - New and improved blocking and indexing techniques
  - Improved record pair classification using un-supervised machine learning techniques (reduce clerical review)
  - Improved performance (scalability and parallelism)

Overview of Febrl software

- Is implemented in *Python* (open source, object oriented, good for rapid prototype development)
- Source code is available (easy to extend and modify)
- Includes many recently developed data linkage algorithms and techniques
- A tool to experiment with and learn about data linkage (facilitated by a graphical user interface)
- **Is a prototype tool, not production software!**
- Freely available at:

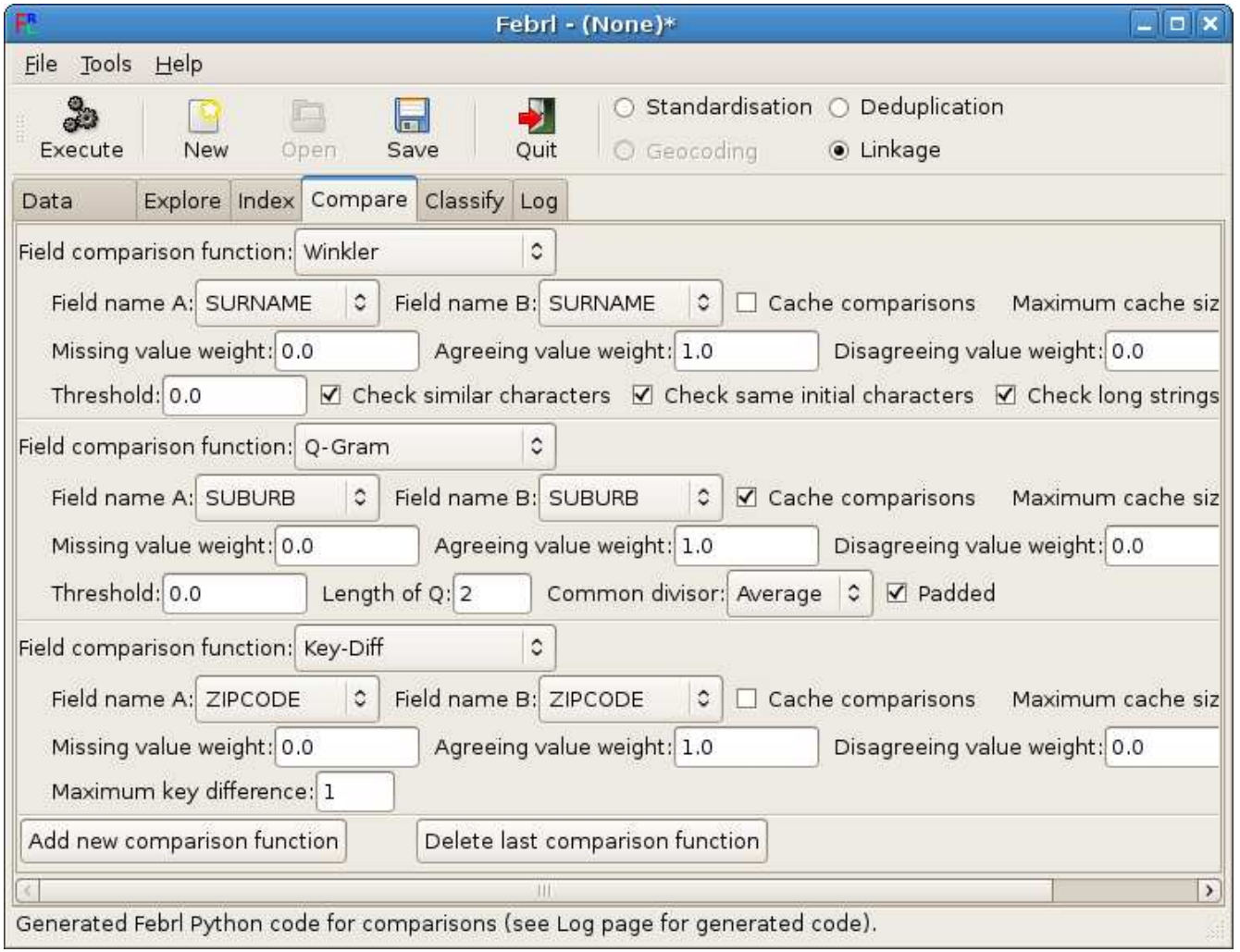
  https://sourceforge.net/projects/febrl/

Main Febrl features

- Three main functionalities
  - Cleaning and standardisation (of names, addresses, dates, and phone numbers)
  - Deduplication of one data set
  - Linkage of two data sets
- A variety of data linkage techniques
  - Seven blocking / indexing methods
  - Twenty-six similarity functions (mainly for strings)
  - Six record pair classifiers
- Includes a data generator and various test data sets

Example Febrl GUI screen-shot

- Showing comparison function definitions



Linking historical census data

- Work done with the ANU Australian Demographic and Social Research Institute (CASS)
- Aim: Reconstruct families and households across historical census data sets that were collected at different points in time
- We have access to a data collection from the UK made of six data sets from 1851 to 1901 (around 30,000 records each)
- Basic idea is to a apply novel backwards-forwards linkage across time (starting with individual records, then families and households)
- We submitted an ARC Discovery Project grant earlier this year

ANU Research Office data linkage

- For *ERA*, match *Thompson ISI* / *Elsevier Scopus* with *ANU ARIES* database
- ANU RO has conducted SQL based linkage
  - Different linkage criteria (‘rule based’)
  - Author names so far not considered
  - Successfully matched around 74% of *ARIES* publications with *ISI*
- Apply more sophisticated data linkage
  - Deal with cases that have typographical errors and variations in authors, journals and articles
  - Combine article and author matches

Example chemistry article titles

- ‘Undecacarbonyl(methylcyclopentadienyl)-tetrahedro-triiridiummolybdenum, undecacarbonyl(tetramethylcyclopentadienyl)-tetrahedro-tr iiridiummolybdenum and undecacarbonyl(pentamethylcyclopentadienyl)-tetrahedro-triiridiummolybdenum’
- ‘Fused supracyclopentadienyl ligand precursors. Synthesis, structure, and some reactions of 1,3-diphenylcyclopenta[l]phenanthrene-2-one, 1,2,3-triphenylcyclopenta[l]phenanthrene-2-ol, 1-chloro-1,2,3-triphenylcyclopenta[l]phenanthrene, 1-bromo-1,2,3-triphenylcyclopenta[l]phenanthrene, and 1,2,3-triphenyl-1H-cyclopenta[l] phenanthrene’

ANU RO data linkage challenges

- Only author surnames and initials in both *ARIES* and *ISI* (many records with 'M Smith' or 'J Williams')
- Journal abbreviations and name changes
- Domain specific article titles (very similar when seen as text strings – such as examples on previous slide)
- What relative matching weights to give to journals, articles and authors?
- Different number of authors (have to normalise number of matched authors by number of listed authors)
- Initial linkage using *Febrl* found all but 7 of the RO matches (and many thousand more new potential matches, including many false positives)

Outlook

- Recent interest in data linkage
  - Data mining and data warehousing, e-Commerce and Web applications
  - Health, census, crime/fraud detection, social security, immigration, intelligence/surveillance
- Main future challenges
  - Automated and accurate linkage (reduce manual efforts)
  - Higher performance (linking very large data sets)
  - Secure and privacy-preserving data linkage
- For more information see our project Web site (publications, talks, software, Web resources / links)

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