Data Matching of Bibliographic Data: Recent Advances and an Open Source Matching System

Peter Christen

Department 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 matching
  - Applications and challenges
  - The matching process and matching techniques
- Data matching for bibliographic data
  - Recent research developments
- ANU Research Office matching pilot project
  - Example chemistry article titles
  - Application of an advanced matching system
- Overview and demonstration of *Febrl*
  (Freely Extensible Biomedical Record Linkage)

Short introduction to data matching

- The process of matching records from one or more data sources that represent the same entity (such as a patient, customer, business, or a *publication*)
  - Also called *record* or *data linkage*, *entity resolution*, *data scrubbing*, *object identification*, *merge-purge*, etc.
- Challenging if no unique entity identifiers available

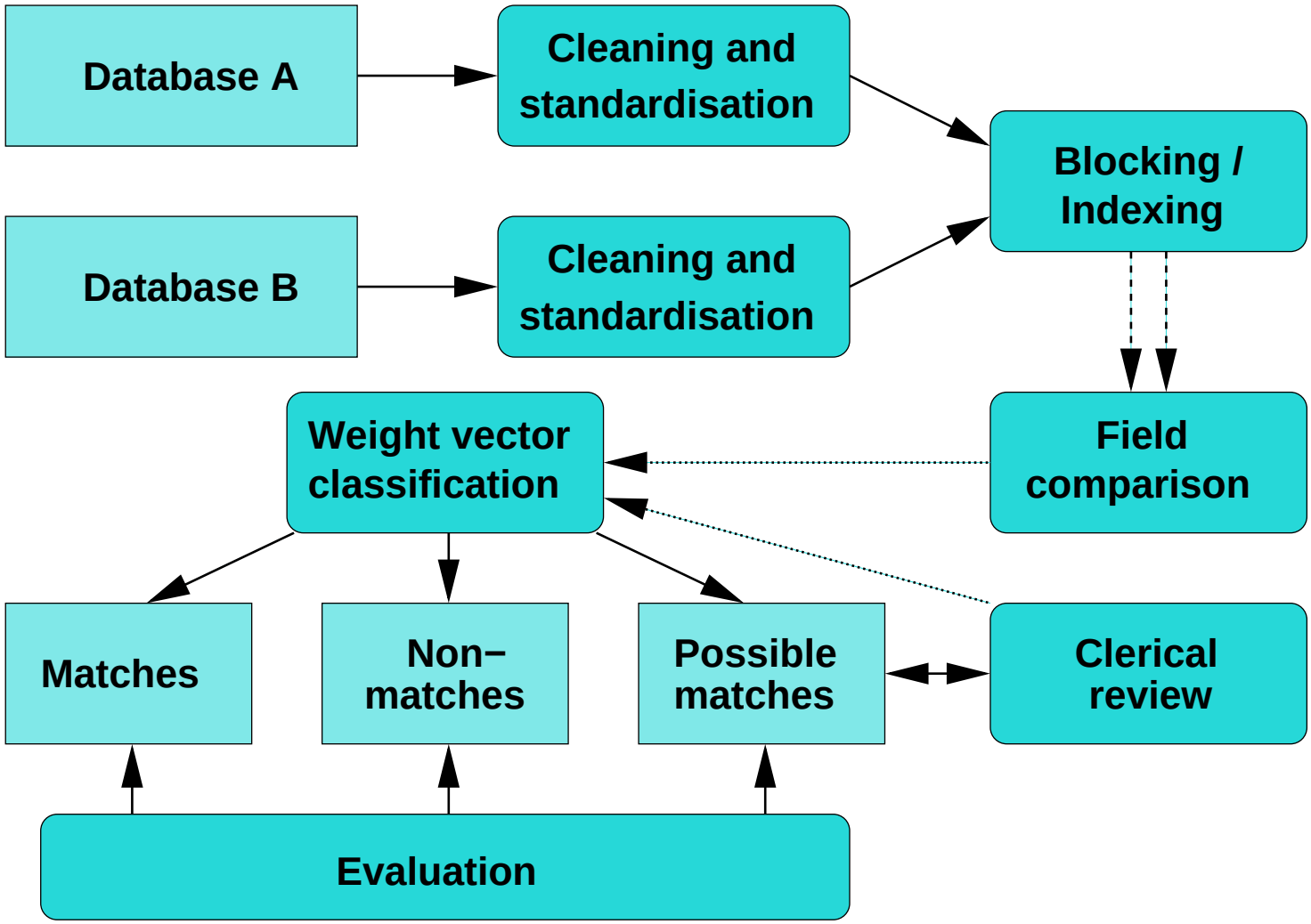
  For example, which of these three records refer to the same person?

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*

Data matching challenges

- Real world data is dirty
  (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability
  - Comparison of all record pairs has quadratic complexity (however, the maximum number of matches is in the order of the number of records in the databases)
  - Some form of blocking, indexing or filtering required
- No training data in many matching applications
  - No record pairs with known true match status
  - Possible to manually prepare training data (but, how accurate will manual classification be?)

The data matching process



Data matching techniques

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  - Can be wrong, missing, coded differently, or out-of-date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

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

Probabilistic data matching

- Computer assisted data matching goes back as far as the 1950s (based on ad-hoc heuristic methods)
- Basic ideas of probabilistic matching 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 ratios) and error estimates
  - Sum of the matching weights is used to classify a pair of records as *match*, *non-match*, or *possible match*
  - Problems: Estimating errors and threshold values, assumption of independence, and *clerical review*

Fellegi and Sunter classification

- For each compared record pair a vector with *matching weights* is calculated

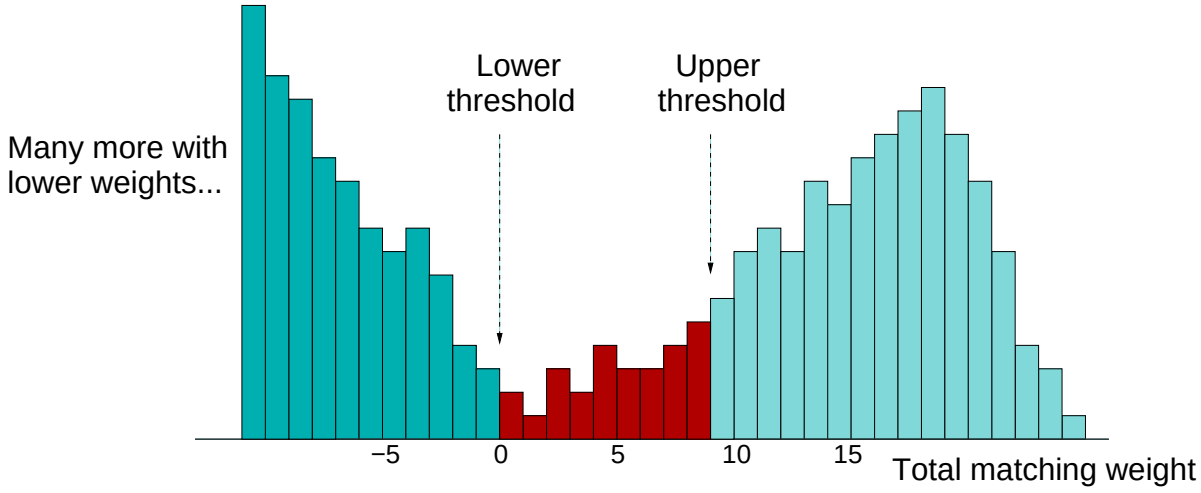
  Record A: `['dr', 'peter', 'paul', 'miller']`

  Record B: `['mr', 'john', '', 'miller']`

  Matching weights: `[0.2, -3.2, 0.0, 2.4 ]`

- *Fellegi and Sunter* approach sums all weights

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



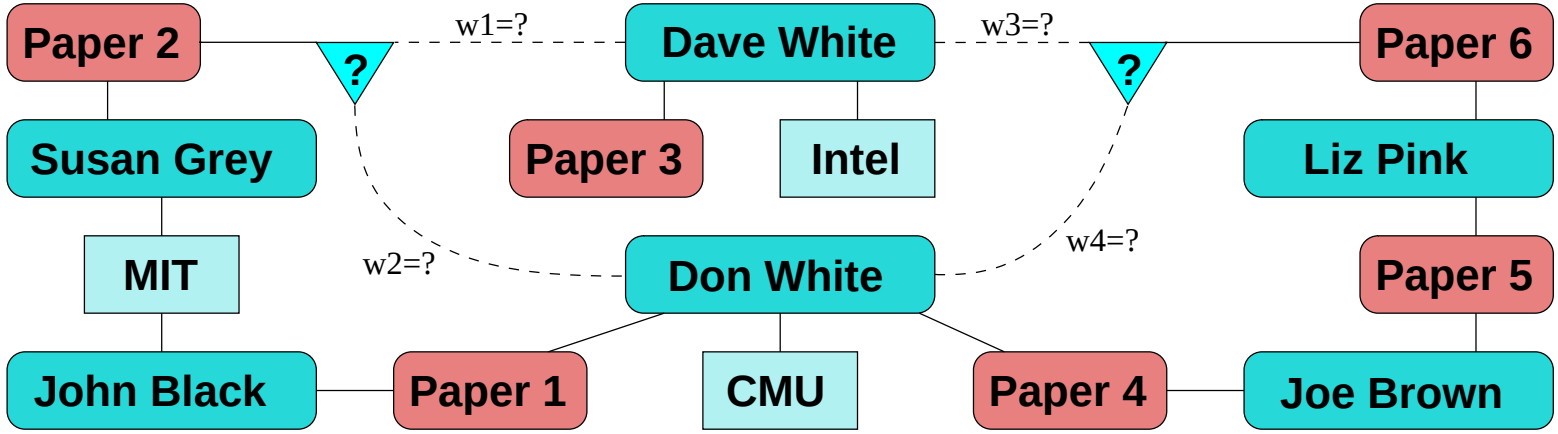
Modern matching approaches

- Summing of weights results in loss of information
  (like *same name but different address*, or *different address but same name*)
- View record pair classification as a *multi-dimensional binary classification* problem
  (use weight vector to classify record pairs as *matches* or *non-matches*, but not *possible matches*)
- Many machine learning techniques can be used
  - Supervised: *Decision trees*, *neural networks*, *learnable string comparisons*, *active learning*, etc.
  - Un-supervised: Various *clustering* algorithms
- Major issue: Lack of training data

Matching bibliographic data

- Most computer science research in data matching uses bibliographic data for experiments
  - Publicly available (*'Cora'*, a small machine learning publication data set)
  - No privacy and confidentiality issues (compared to personal data, such as patient records)
- Complex domain with different entity types (authors, articles, venues, institutions)
- Most research has focussed on matching quality
  - Collective matching of a complete database
  - Use relational information (connections between entities), rather than just attribute similarities

Collective matching 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

Collective matching issues

- Several approaches have been developed (by machine learning, data mining and database communities)
- Combine graph and clustering based techniques (iteratively refine connection weights)
- Generally achieve much improved matching quality (compared to traditional matching based only on attribute similarities between two records)
- However, the computational complexity of these approaches is generally high
  - For matching two databases with n records each, $n \times n$ calculation steps (or more) are required
  - Not scalable to large databases

ANU Research Office matching

- For *ERA*, match *Thompson ISI* with *ANU ARIES* database (*ISI*: 1,420,083 authors, 414,897 publications; *ARIES*: 116,142 authors, 15,166 publications; 2,569 *ARIES* publications are in non-ISI journals)
- ANU RO has conducted SQL based matching
  - Different matching criteria ('rule based')
  - Author names so far not considered
  - Successfully matched 9,232 *ARIES* publications (74%)
- Apply more sophisticated matching
  - 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-triiridiummolybdenum 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 matching 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 matching using *Febrl* found all but 7 of the RO matches (and many thousand more new potential matches, including many false positives)

Overview of Febrl

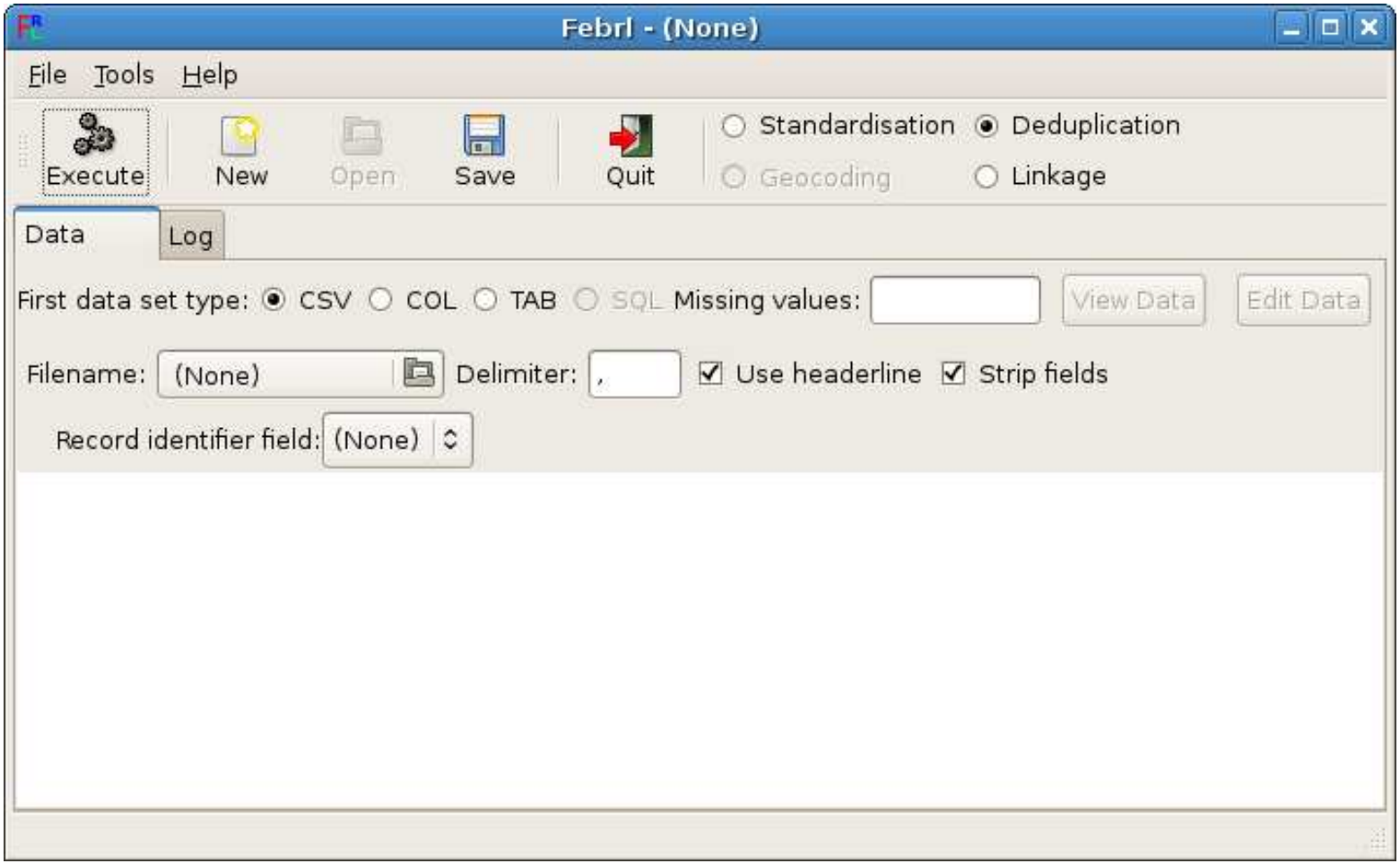
- Has been developed since 2002 (as part of a project between the ANU and the *NSW Department of Health*)
- Is implemented in *Python* (open source, object oriented, good for rapid prototype development)
- Source code is available (easy to extend and modify)
- Includes many recently developed data matching algorithms and techniques
- A tool to experiment with and learn about data matching
- **Is a prototype tool, not production software!**
- Freely available at:

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

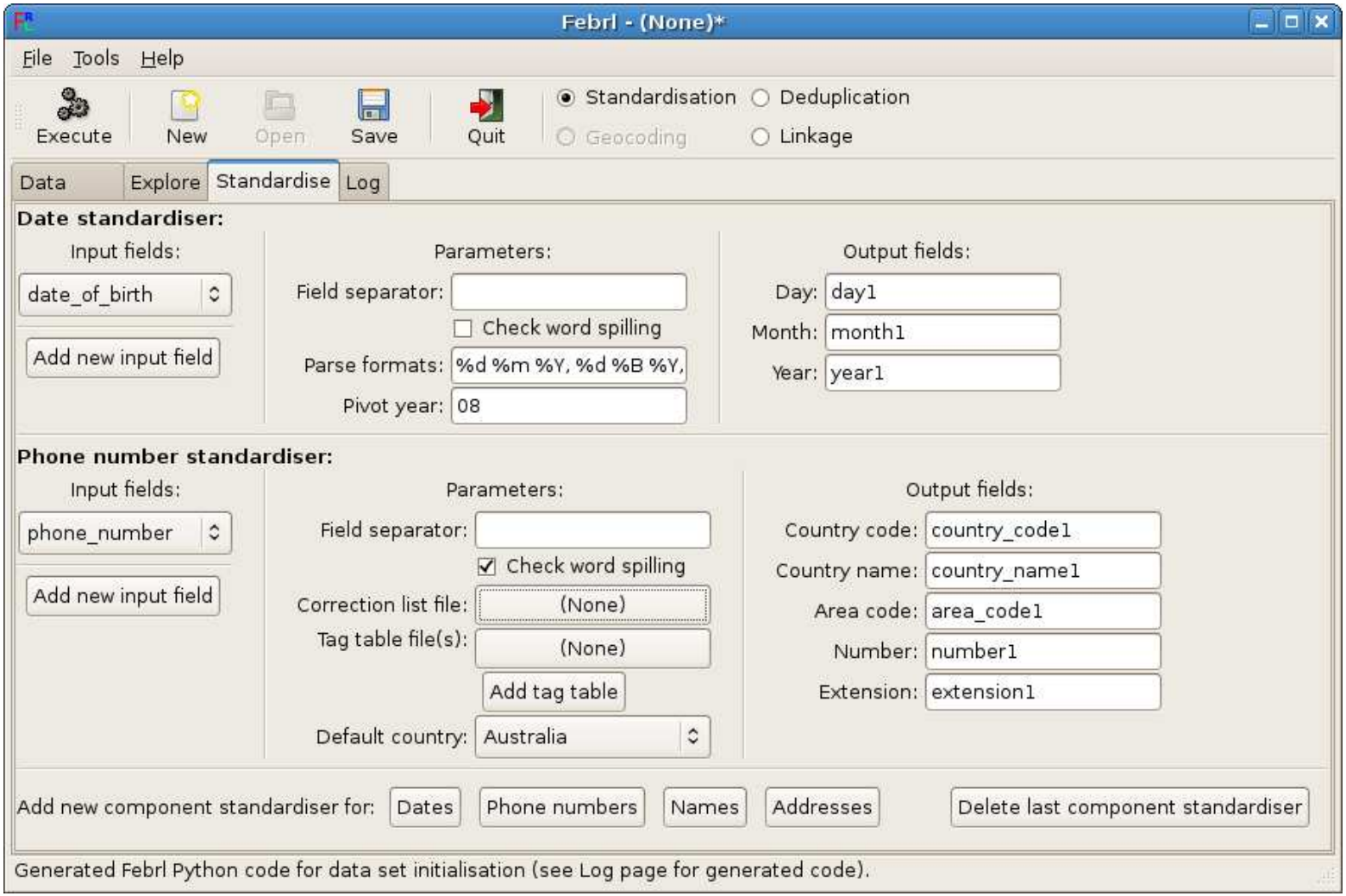
Main Febrl features

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

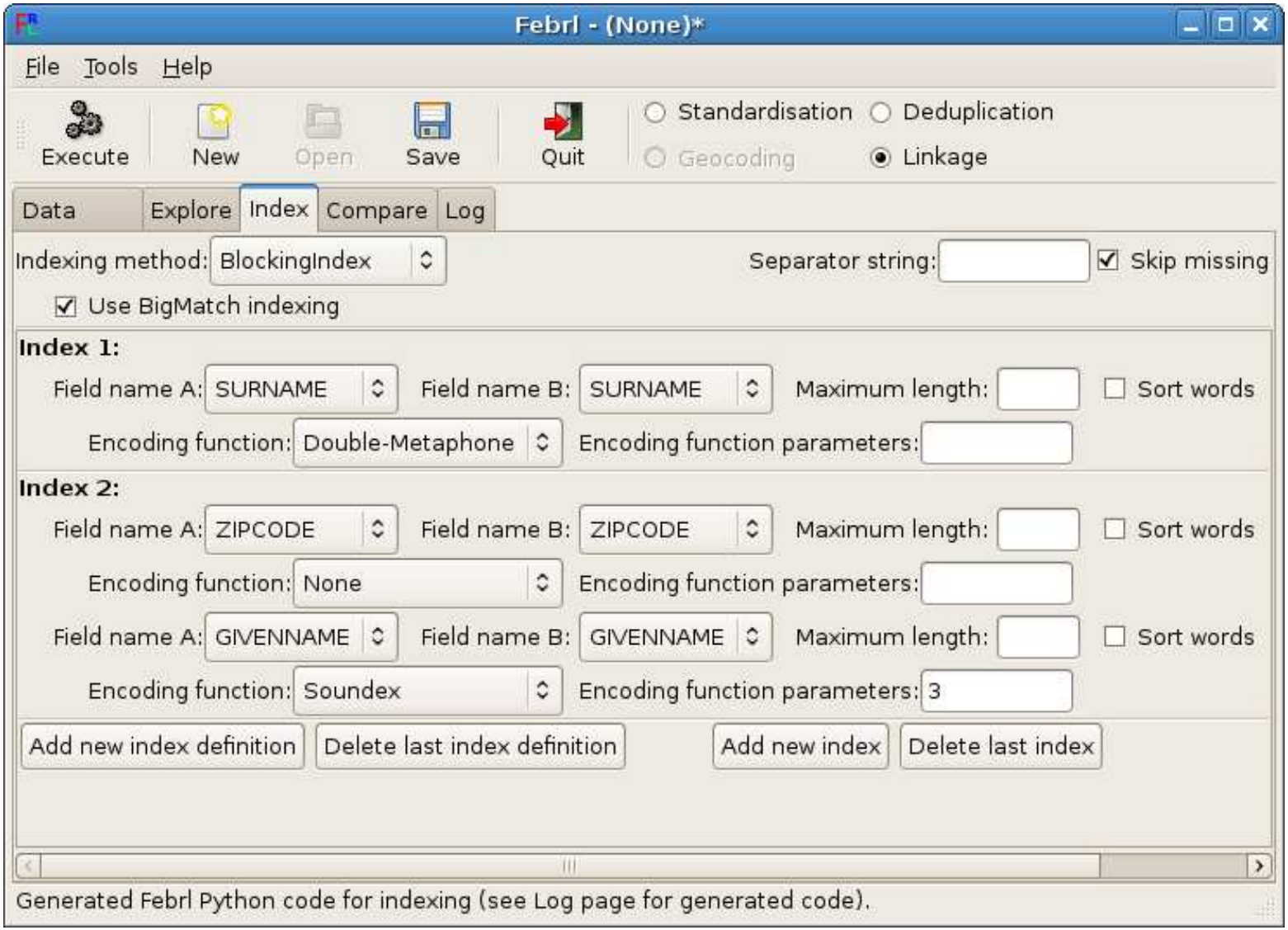
Initial Febrl graphical user interface



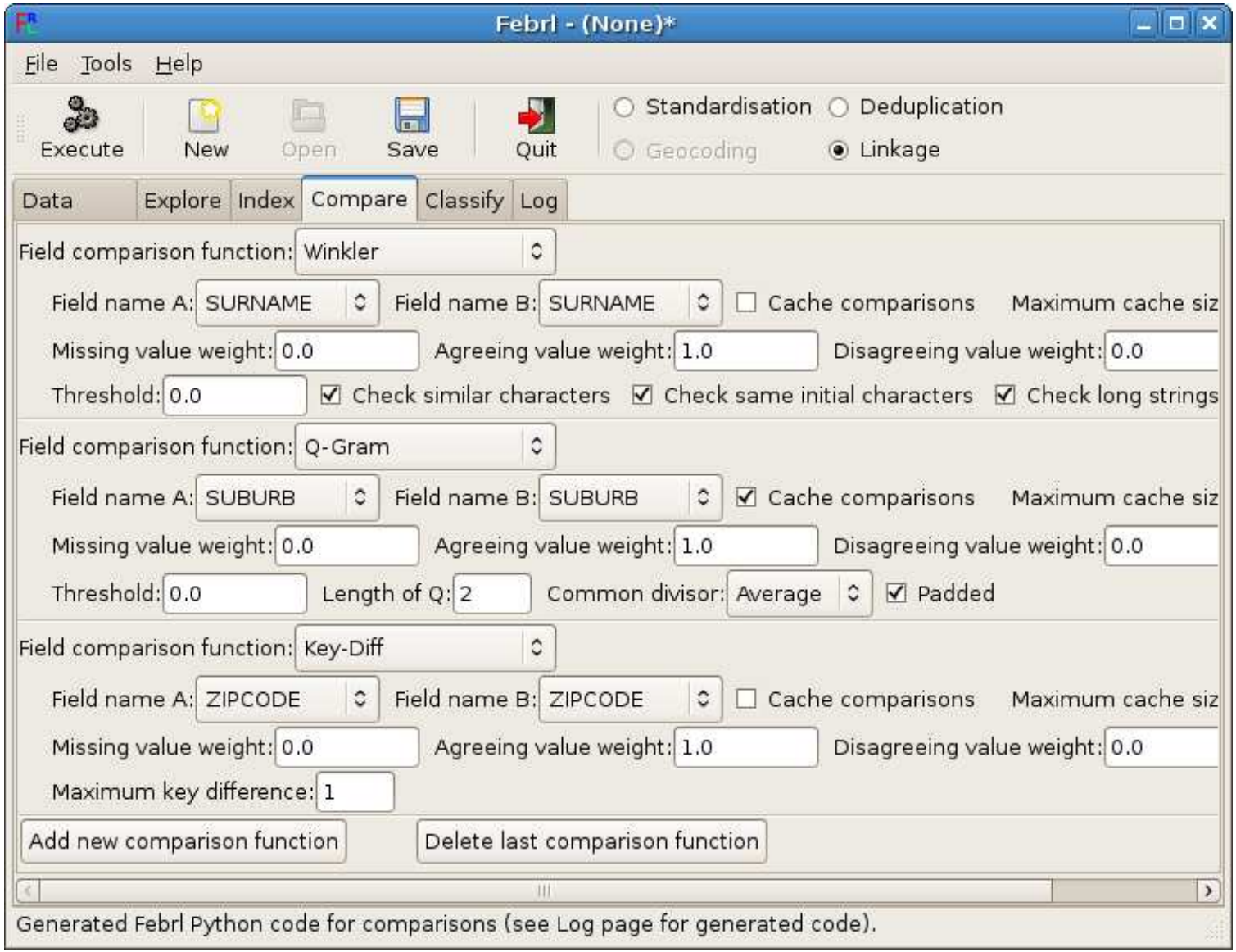
Date and phone standardisers



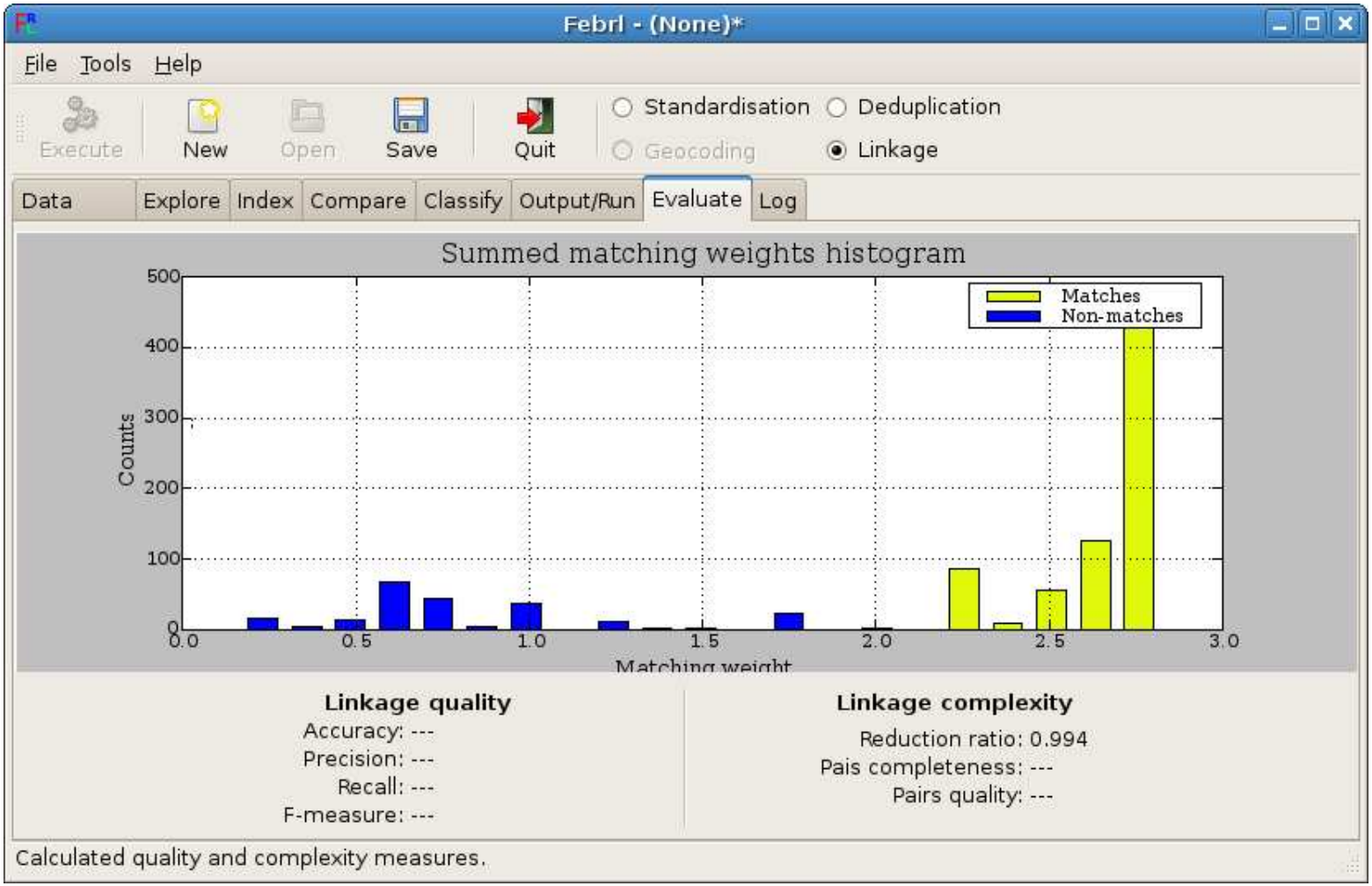
Indexing (blocking) definition



Comparison functions



Matching weights histogram



Conclusions

- Recent advances in matching bibliographic databases using collective matching approaches (however, currently not scalable to very large databases)
- Data matching is domain and data dependent
  - Requires domain knowledge
  - Requires knowledge about data matching techniques
  - Requires manual intervention
- Matching for *ERA* will likely require specific matching approaches and tools (possibly domain dependent approaches, such as for physics, medicine, engineering, humanities, etc.)