# Accurate Synthetic Generation of Realistic Personal Information

Peter Christen[1] and Agus Pudjijono[2]

[1]School of Computer Science,
ANU College of Engineering and Computer Science,
The Australian National University
Canberra, Australia

[2]Data Center,
Ministry of Public Works of Republic of Indonesia
Jakarta, Indonesia

Contact: peter.christen@anu.edu.au

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



## Outline

- Why synthetic data generation?
- Advantages and challenges of synthetic data
- Modelling of variations and errors
- The new *Febrl* data generator
  - The data generation process
  - Generate family and household data
  - Duplicate record modification
- Example of generated data
- Outlook and future work



## Why synthetic data generation?

- A large portion of data collected today is about people (such as customers, clients, patients, tax payers, students, travellers, employees, etc.)
- Analysis, mining and sharing of such data can result in privacy and confidentiality issues (especially when data needs to be matched or exchanged between organisations)
- Privacy issues prohibit publication of real data (that contains personal information)
- It is therefore difficult for researchers to efficiently conduct their work if they rely upon such data (for example for research in deduplication, data linkage, data mining, or information retrieval and extraction)



## Synthetic data – Advantages

- Privacy issues prohibit publication of real data (for example of names, addresses, dates of birth, etc.)
- De-identified or encrypted data cannot be used (as real name and address values are required, for example for data linkage or deduplication research)
- Several advantages of synthetic data
  - Volume and characteristics can be controlled (errors and variations in records, number of duplicates, etc.)
  - It is known which records are duplicates of each other, and so matching quality can be calculated
  - Data and the data generator program can be published (allowing others to repeat experiments)

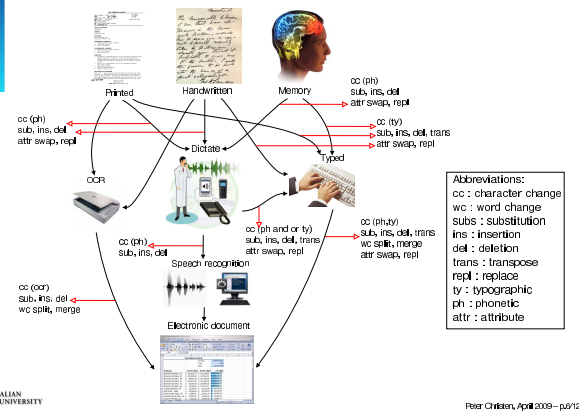


## Synthetic data – Challenges

- Modelling the content and characteristics of real data (frequencies of values; variations and errors)
- Modelling dependencies between attributes (for example, given names often depend on gender)
- Earlier data generators were much simpler
  - *Hernandez and Stolfo* (mid 1990s): Only based on value tables, no frequencies, simple typographic errors
  - *Bertolazzi et al.* (2003): Added frequency tables, allowed missing values, still simple error generation
  - *Christen* (2005): First version of *Febrl* generator, added look-up tables with misspellings. nicknames, etc.



## Modelling of variations and errors

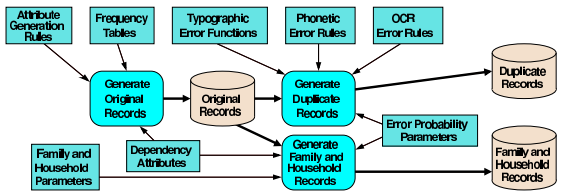


## The new Febrl data generator

- Can generate different types of modifications
  - Typographic (insert, delete, substitute, transpose)
  - Phonetic (based on transformation rules – more later)
  - Optical character recognition (OCR) (single or groups of characters that look similar)
- Can generate family and household data
  (groups of records with same address but different given names and ages – more later)
- Can model dependencies between attributes
  - Using look-up tables with dependency information
  - With a certain probability (set by user), a dependency is not followed



## The data generation process




- Step 1: Generate original records
- Step 2: Generate duplicates of these originals, or generate family and household records



## Family and household generation

- For a family, select an original record at random, then determine its *role* according to its values
  (possible roles are *wife*, *husband*, *daughter*, or *son*)
  - Then randomly choose the number of members to be generated for this family
- Copy the original record and change age, given name and gender values (and with small probability also address, assuming a child has left home)
  - Similar approach for households, but also change surnames and keep all ages above 18
- Family and household data generation involves many parameters to be set by the user



## Phonetic modifications for duplicates

- Based on phonetic encoding rules that are used in *Soundex*, *Phonix*, *Double-Metaphone*, etc.
  (methods to group together strings that sound similar)
- Currently, around 350 phonetic modification rules (each made of *position*, *original pattern*, *substitute pattern*, and four *conditions*)
- Example phonetic rules
  - ALL, 'h' → '@' No condition ('@' refers to the empty string)
    *(mustapha → mustapa)*
  - END, 'le' → 'ile' Condition: Only after a consonant
    *(brambl<u>e</u> → brambil<u>e</u>)*
  - MIDDLE, 'ge' → 'ke' Condition: Start with 'van', 'von', or 'sch'
    *(van g<u>e</u>raldus → van k<u>e</u>raldus)*



## Example of generated data

| rec_id, | age, | given_name, | surname, | street, | suburb |
|---|---|---|---|---|---|
| rec-1-org, | ***33***, | ***Madison***, | Solomon, | Tazewell ***Circuit***, | ***Beechboro*** |
| rec-1-dup-0, | 33, | Madisoi, | Solomon, | Tazewell Circ, | Beech Boro |
| rec-1-dup-1, | , | Madison, | Solomon, | Tazewell Crct, | Bechboro |
| rec-2-org, | 39, | ***Desirae***, | ***Contreras***, | Maltby Street, | ***Burrawang*** |
| rec-2-dup-0, | 39, | Desirae, | Kontreras, | Maltby Street, | Burawang |
| rec-2-dup-1, | 39, | Desire, | Contreras, | Maltby Street, | Buahrawang |
| rec-3-org, | ***81***, | ***Madisyn***, | Sergeant, | ***Howitt*** Street, | ***Nangiloc*** |
| rec-3-dup-0, | 87, | Madisvn, | Sergeant, | Hovvitt Street, | Nanqiloc |

- Typographic (rec-1), phonetic (rec-2) and OCR (rec-3) modifications



## Outlook and future work

- We have presented a novel data generator that can create realistic personal information
- Much improved compared to similar earlier data generators
- Part of the *Febrl* data linkage system
  (Freely extensible biomedical record linkage)
- Various avenues for future work
  - Extend family roles (nieces, cousins, aunts, uncles, etc.)
  - Enable Unicode to allow generation of international data
  - Develop a GUI to facilitate setting of parameters
- Freely available at:
  **https://sourceforge.net/projects/febrl/**