Probabilistic Name and Address Cleaning and Standardisation

Peter Christen, Tim Churches and Justin Xi Zhu

Data Mining Group, Australian National University

Centre for Epidemiology and Research, New South Wales Department of Health

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

Project web page: **http://datamining.anu.edu.au/linkage.html**

Funded by the ANU, the NSW Department of Health and the Australian Partnership for Advanced Computing (APAC)

Outline

- Data cleaning and standardisation
- Record linkage and data integration
- Our approach
  - Cleaning
  - Tagging
  - Segmentation
- Hidden Markov models for data segmentation
- Experimental results
- *Febrl* – Freely extensible biomedical record linkage

Data cleaning and standardisation (I)

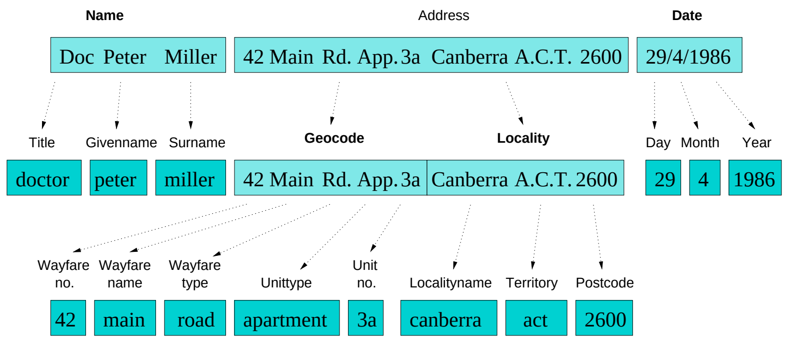
- Real world data is often *dirty*
  - Missing values
  - Typographical and other errors
  - Different coding schemes
  - Outdated data
- Names and addresses are especially prone to data entry errors
- Cleaned and standardised data is needed for
  - loading into databases and data warehouses
  - data mining and other data analysis studies
  - record linkage and data integration

Record linkage and data integration

- The task of linking together information from one or more data sources representing the same entity
- If no *unique identifier* is available, *probabilistic linkage techniques* have to be applied
- Applications of record linkage
  - Remove duplicates in a data set (internal linkage)
  - Merge new records into a larger master data set
  - Create customer or patient oriented statistics
  - Compile data for longitudinal studies

Data cleaning and standardisation is an important first step for successful record linkage

Data cleaning and standardisation (II)



- Remove unwanted characters and words
- Expand abbreviations and correct misspellings
- Segment data into well defined *output fields*

Our approach

1. Data cleaning
   - Remove unwanted characters and words
   - Correct various misspellings and abbreviations
2. Data tagging
   - Split into a list of words, numbers and separators
   - Assign one or more tags to each element of this list
3. Data segmentation
   - Assign list elements to *output fields*
   - Use *hidden Markov models* (HMMs) or rules

Data cleaning

- Assume the input *component* is one string
  (name or address – dates are processed differently)
- Convert all letters into lower case
- Use *correction lists* which contain pairs of `original:replacement` strings
- An empty `replacement` string results in removing the original string
- Correction lists are stored in text files and can be modified by the user
- Different correction lists for names and addresses

Data tagging

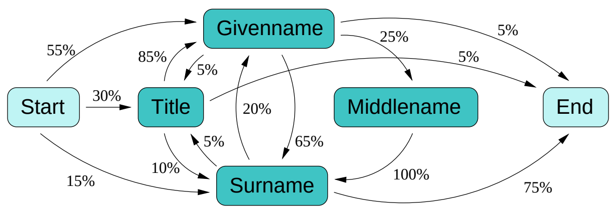
- Split cleaned string at whitespace boundaries into a list of words, numbers, characters, etc.
- Using *look-up tables* and some had-coded rules, each element is tagged with one or more *tags*
- Example:
  - Uncleaned input string: *“Doc. peter Paul MILLER”*
  - Cleaned string: *“dr peter paul miller”*
  - Word and tag lists:

```
['dr', 'peter', 'paul', 'miller']
['TI', 'GM/SN', 'GM',    'SN'   ]
```

Data segmentation

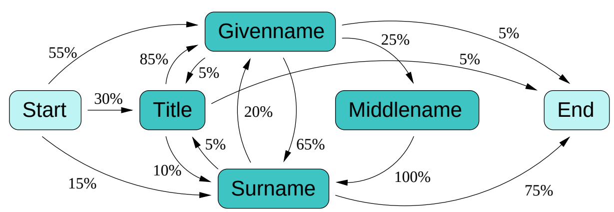
- Using the tag list, assign elements in the word list to the appropriate *output fields*
- Rules based approach (e.g. *AutoStan*)
  - Example: *“if an element has tag ‘TI’ then assign the corresponding word to the ‘title’ output field”*
  - Hard to develop and maintain rules
  - Different sets of rules needed for different data sets
- Hidden Markov model (HMM) approach
  - A machine learning technique (supervised learning)
  - Training data is needed to build HMMs

Hidden Markov model (HMM)



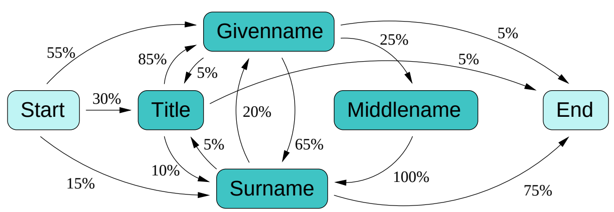
- A HMM is a *probabilistic* finite state machine
  - Made of a set of *states* and *transition probabilities* between these states
  - In each state an *observation* symbol is emitted with a certain probability distribution
  - In our approach, the observation symbols are *tags* and the states correspond to the *output fields*

HMM probability matrices



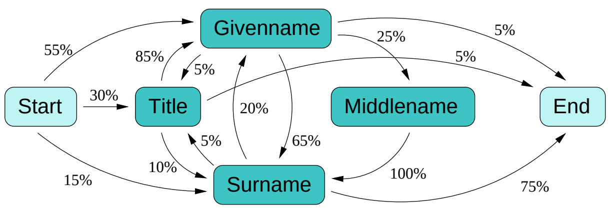
	State					
Observation	*Start*	Title	Givenname	Middlename	Surname	*End*
TI	–	96%	1%	1%	1%	–
GM	–	1%	35%	33%	15%	–
GF	–	1%	35%	27%	14%	–
SN	–	1%	9%	14%	45%	–
UN	–	1%	20%	25%	25%	–

HMM data segmentation



- For an observation sequence we are interested in the most probable path through a given HMM
  (in our case an observation sequence is a *tag list*)
- The *Viterbi* algorithm is used for this task
  (a dynamic programming approach)
- *Smoothing* is applied to account for unseen data
  (assign small probabilities for unseen observation symbols)

HMM segmentation example



- Input word and tag list

  `['dr', 'peter', 'paul', 'miller']`

  `['TI', 'GM/SN', 'GM', 'SN' ]`

- Two example paths through HMM

```
Start -> Title (TI) -> Givenname (GM) ->
        Middlename (GM) -> Surname (SN) -> End
Start -> Title (TI) -> Surname (SN) ->
        Givenname (GM) -> Surname (SN) -> End
```


HMM training (I)

- Both transition and observation probabilities need to be trained using *training data*
  (maximum likelihood estimates (MLE) are derived by accumulating frequency counts for transitions and observations)
- Training data consists of records, each being a sequence of `tag:hmm_state` pairs
- Example (2 training records):

```
# '2 richard street lewisham 2049 new_south_wales'
NU:wfnu,UN:wfna1,WT:wfty,LN:loc1,PC:pc,TR:ter1

# '42 / 131 miller place manly 2095 new_south_wales'
NU:unnu,SL:sla,NU:wfnu,UN:wfna1,WT:wfty,LN:loc1,PC:pc,TR:ter1
```


HMM training (II)

- A *bootstrapping* approach is applied for semi-automatic training
  1. Manually edit a small number of training records and train a first rough HMM
  2. Use this first HMM to segment and tag a larger number of training records
  3. Manually check a second set of training records, then train improved HMM
- Only a few person days are needed to get a HMM that results in an accurate standardisation
  (instead of weeks or even month to develop rules)

Address standardisation results

- Various NSW Health data sets (millions of records)
  - *HMM1* trained on 1,450 Death Certificate records
  - *HMM2* contains *HMM1* plus 1,000 Midwifes Data Collection training records
  - *HMM3* is *HMM2* plus 60 unusual training records
- *AutoStan* rules (for ISC) developed over years

	HMM/Method			
Test Data Set (1,000 records each)	HMM 1	HMM 2	HMM 3	Auto Stan
Death Certificates	95.7%	96.8%	97.6%	91.5%
Inpatient Statistics Collection	95.7%	95.9%	97.4%	95.3%

Name standardisation results

- NSW Midwifes Data Collection (1990 - 2000)
  (around 963,000 records, no medical information)
- 10-fold cross-validation study with 10,000 random records (9,000 training and 1,000 test records)
- Both rule based and HMM data cleaning and standardisation
  - Rules were better because most names were simple (not much structure to learn for HMM)

	Min	Max	Average	StdDev
HMM	83.1%	97.0%	92.0%	±4.7%
Rules	97.1%	99.7%	98.2%	±0.7%

Outlook - Febrl

- HMM approach is comparable with traditional rule based approach (but easier to develop and maintain)
- Implemented in *Febrl* **http://febrl.sourceforge.net**
  (Freely extensible biomedical record linkage)
  - Open-source, *Python*, multi-platform
- Currently under development are
  - probabilistic record linkage routines
  - new *fuzzy indexed look-up* mechanisms
  - parallel techniques for standardisation and linkage
  - predictive modelling for increased linkage quality