A Comparison of Personal Name Matching: Techniques and Practical Issues

Peter Christen

Department of Computer Science,
Faculty of Engineering and Information Technology,
ANU College of Engineering and Computer Science,
The Australian National University

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

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

Funded by the Australian National University, the NSW Department of Health,
and the Australian Research Council (ARC) under Linkage Project 0453463.

Outline

- Why is personal name matching important?
  - Some application areas
- Personal name characteristics
- Sources of variations and errors
- Matching techniques
  - Phonetic encoding
  - Pattern matching
  - Combined techniques
- Some experimental results
- Discussion and outlook

Why is name matching important?

- A lot of data collected and processed contains information about people (for example patients, customers, authors, students, politicians, film/music and sport stars, work colleagues, friends and family)
- Personal names are often used as *identifiers* to access data or when searching for people (for example Web or bibliographic searches)
- Three main application areas for name matching
  - Text data mining
  - Information retrieval
  - Data linkage and deduplication

Personal name characteristics

- Personal names can have several valid variations (for example *Gale*, *Gail* and *Gayle*)
  - Make use of dictionary based spelling correction hard
- People often use nicknames (like *Liz*, *Bill* or *Bob*)
- Personal names change over time (most commonly when somebody gets married)
- Names are influenced by language and culture
  - Several transliterations from Asian to Roman alphabet
  - Compound names in French and German (for example *Jean-Pierre* and *Hans-Peter*)
  - Arabic name often have several components and contain various affixes

Types of errors in names

- Damerau (1964) found that 80% of spelling errors were single character errors (inserts, deletes, or substitutions) (other studies reported similar results)
- A study (Friedman et al. 1992) on hospital patient names reported almost 40% of errors were insertion of an additional name word, initial or title (only around 40% of all errors were single character errors)
- Kukich (1991) classifies character level errors as:
  - Typographical errors (correct spelling known)
  - Cognitive errors (lack of knowledge or misconceptions)
  - Phonetic errors (similar sounding spelling)

Sources of variations and errors (1)

- Scanning of handwritten forms (optical character recognition, transpositions of similar looking characters)
- Manual keyboard entry (wrongly typed neighbouring keys, like $e \leftrightarrow r$ or $k \leftrightarrow j$)
- Data entry over telephone (a confounding factor to manual keyboard entry, sometimes a default spelling is assumed)
- Limitations in length of input fields (forces people to omit name parts, or use abbreviations and initials only)
- People themselves sometimes provide different name variations (depending upon the organisation they are in contact with)

Sources of variations and errors (2)

- Different characteristics of variations if names come from different sources (challenging in distributed text data mining and data linkage systems)
- Recent development of *adaptive* name matching systems need training data (they can only deal with variations and errors as found in the training data)
- When matching names one has to deal with
  - Legitimate name variations (that should be preserved and matched)
  - Errors introduced during data entry and recording (that should be corrected)

Matching techniques

- Two main approaches
  - Phonetic encoding (followed by exact matching)
  - Pattern matching (approximate string matching)
- Combined approaches aim to improve the matching quality
- Many different approximate string matching techniques have been developed
  - Generally normalised into a *similarity* measure
  - Two strings are the same → sim = 1.0
  - Two strings are totally different → sim = 0.0
  - Two strings are somewhat similar → 0.0 > sim < 1.0

Phonetic encoding

- Are language dependent (pronunciations)
- Soundex (using an encoding table to convert names into a one-character-three-digit code, e.g. *Peter* → *P360*)
- Phonex (improves on Soundex by pre-processing names according to English pronunciations)
- Phonix (more than 100 transformations on letter groups)
- NYSIIS (New York State Identification and Intelligence System, similar to Phonex, code only contains letters)
- Double-Metaphone (aims to better account for non-English names, can return two codes)
- Fuzzy-Soundex (based on q-gram substitutions, combines elements from other phonetic encodings)

Pattern matching (1)

- Levenshtein or Edit-distance (smallest number of inserts, deletes or substitutions needed to transform one string into another)
- Damerau-Levenshtein distance (counts a trans-position as one edit operation rather than two)
- Bag distance (cheap approximate to edit-distance, counts common characters)
- Smith-Waterman distance (accounts for gaps, often used in biological sequence comparisons)
- Longest common sub-string (applied repeatedly until a minimum length is reached)
- Q-grams (counts sub-strings of lengths q in common)

Pattern matching (2)

- Positional q-grams (take position into account, only match within a maximum distance)
- Skip-grams (based on the idea of forming q-grams also of characters not adjacent to each other, accounts for inserts and deletes; has been used in multi-lingual IR)
- Compression (apply a standard compressor (*gzip* or *bz2*) to compress strings independently and concatenated, then use compression lengths to calculate similarity)
- Jaro (similarity is calculated counting common and transposed characters; commonly used in data linkage)
- Jaro-Winkler (increase similarity if beginning of names is the same (up to 4 characters), or strings are long, or characters are similar)

Combined techniques

- Editex (combines edit-distance methods with Soundex letter-groupings, edit cost is 0 if two letters are the same, 1 if in the same letter group, 2 otherwise; has been used in IR)
- Syllable alignment distance (idea is to match names syllable by syllable rather character by character, applies rules to get syllables, then uses edit-distance based method for matching)
- Authors of both techniques claim to achieve better matching performance than other methods [Zobel and Dart, 1996; Gong and Chan, 2006]

Comparison experiments

	Pairs	Singles
Midwives given names	15,233	49,380
Midwives surnames	14,180	79,007
Midwives full names	36,614	339,915
COMPLETE surnames	8,942	13,941

- Test data sets based on real world names
  - **Midwives** [New South Wales Health, 2001]
  - **COMPLETE** [Pfeifer, Poersch, Fuhr, 1996]
- Matching implemented in Python using *Febrl* (Freely Extensible Biomedical Record Linkage)
- Evaluated using average f-measure (varying threshold from 0.0 to 1.0)

Matching results

- We ran a total of 123 tests on each data set (many matching methods have different parameter settings)
- Main results
  - No technique performs better than all others
  - Pattern matching methods clearly outperform phonetic encoding methods
  - Simple phonetic encoding methods perform better than more complex ones
  - Combined techniques do not perform as good as expected
  - Surnames are harder to match than given names (due to complete name changes)

Discussion and outlook

- Personal names have characteristics that are different from general text
- Many different name matching techniques have been develop
  - Pattern matching techniques outperform phonetic encoding techniques
  - No technique performs better than all others
  - Practical issues (like setting parameters) make finding best matching method challenging
- For more information see our project Web site (publications, talks, *Febrl* data linkage software)
  http://datamining.anu.edu.au/linkage.html