Towards Scalable Real-Time Entity Resolution using a Similarity-Aware Inverted Index Approach

Peter Christen[1] and Ross Gayler[2]

[1] **Department of Computer Science,**
ANU College of Engineering and Computer Science,
The Australian National University,
Canberra ACT 0200, Australia

[2] **Veda Advantage,**
Melbourne VIC 3000, Australia

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

Outline

- Introduction to entity resolution
  - Applications and challenges
  - Entity resolution techniques
- Real-time entity resolution
- Indexing for real-time entity resolution
  1. Standard blocking
  2. Similarity-aware inverted index
  3. Materialised similarity-aware inverted index
- Experimental evaluation
- Conclusions and future work

What is entity resolution?

- The process of matching and aggregating records that represent the same entity (such as a patient, a customer, a business, an address, or an article)
  - Also called *data matching*, *record* or *data linkage*, *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*

Applications of entity resolution

- Health, biomedical and social sciences
- Census, taxation, social security
- Deduplication of (business mailing) lists
- Bibliographic databases and online libraries
- Geocode matching ('geocoding') of addresses for spatial analysis
- Crime and fraud detection, national security
- *Identity verification*
  - For example, credit card applications
  - Match applicant's details with large databases that contain existing identities

Entity resolution 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  (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)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 $O(n \times m)$
  - Some form of blocking, indexing or filtering is required
- Privacy and confidentiality
  (because personal information, like names and addresses, is commonly required for matching)
- No training data in many application areas
  (no record pairs with known true match status)

Entity resolution techniques

- Traditional approaches only consider attribute similarities (using various similarity functions)

```
Record A:         ['dr', 'peter', 'paul', 'miller']
Record B:         ['mr', 'john',  '',     'miller']
Matching weights: [0.2,  -3.2,    0.0,    2.4      ]
```

- Classify record pairs using *matching weights*
  (into *matches*, *non-matches*, and maybe *possible matches*, for which clerical review is needed)
- Recently, *collective* entity resolution techniques have been developed
  - Use relational information (connections between entities), rather than just attribute similarities

Real-time entity resolution (1)

- Traditionally, match two static databases
  (only one approach for *query-time* entity resolution: 31 sec for matching a query record with 831,000 records)
- Today, many applications require real-time matching
  - Identity verification during credit application, government services and benefits, e-Health, etc.
  - Crime detection and terrorism prevention systems
  - Health surveillance systems (disease outbreaks)
- A task similar to large-scale Web search (match a record to a large database, return most similar results)

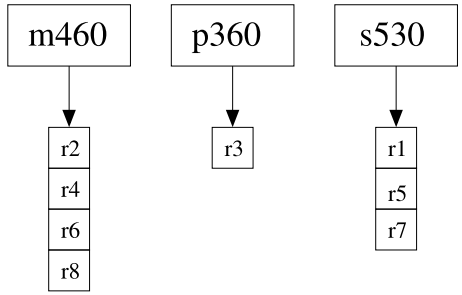
Real-time entity resolution (2)

- Objectives:
  - Process a stream of incoming query records with one or several large databases
  - Match these query records as quickly as possible
  - Generate a match-score (allows setting a threshold)
- Challenges:
  - Large databases with many million records
  - Dynamic database updates
  - User constraints (like *black-lists*, or known name variations of people who have changed names)
  - Multiple databases with different information content

Indexing for real-time entity resolution

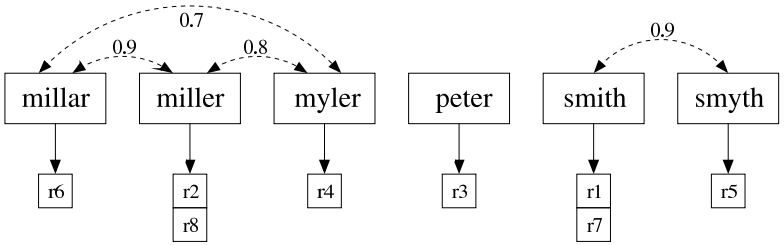
- Combine inverted index approach with similarity calculations (like approximate comparisons of names)
- Two phases of real-time entity resolution:
  1. Build index on database (insert all database records into index)
  2. Query index with incoming records (who's values might be in the index or not)
- We have implemented three index variations
  - Similarity functions return values from *0* (for total dissimilarity) to *1* (for exact similarity)
  - Use phonetic encoding (such as *Soundex*) to group record values into blocks

Standard blocking (inverted) index



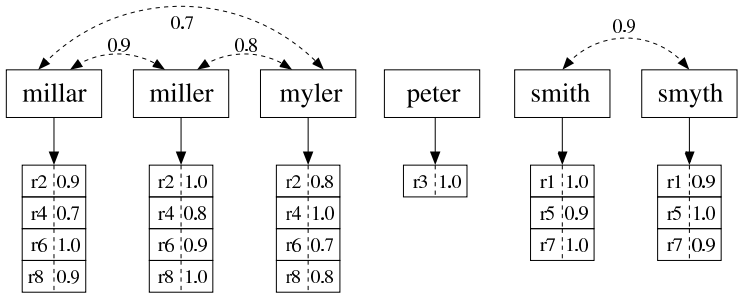
Record ID	Surname	Soundex encoding
r1	smith	s530
r2	miller	m460
r3	peter	p360
r4	myler	m460
r5	smyth	s530
r6	millar	m460
r7	smith	s530
r8	miller	m460

Similarity-aware inverted index



Record ID	Surname	Soundex encoding
r1	smith	s530
r2	miller	m460
r3	peter	p360
r4	myler	m460
r5	smyth	s530
r6	millar	m460
r7	smith	s530
r8	miller	m460

Materialised similarity-aware inverted index



Record ID	Surname	Soundex encoding
r1	smith	s530
r2	miller	m460
r3	peter	p360
r4	myler	m460
r5	smyth	s530
r6	millar	m460
r7	smith	s530
r8	miller	m460

Optimisations

- There is a large body of research on optimisation of inverted index techniques for search engines (not all of it published, most work commercial)
- Based on sorting or filtering of index elements
- We have implemented a *threshold* based filtering
  - In real applications, an index is built on several attributes (like in the following experiments)
  - Similarities are summed over attributes (for example: $sim_{name} = 0.6$, $sim_{suburb} = 0.3$, $sim_{postcode} = 0.9$)
  - Filter records that are guaranteed not to reach overall threshold (like with threshold $t = 2.2$, the above record can be removed after *suburb* similarity is calculated)

Experimental evaluation

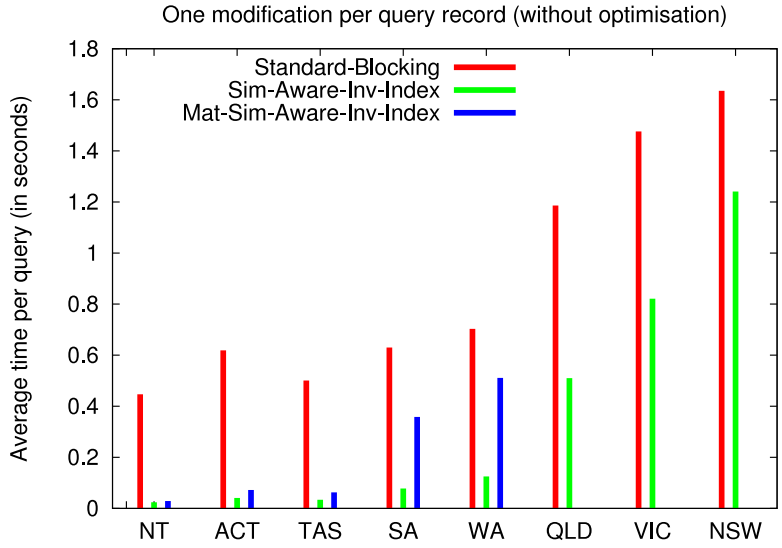
Australian state/territory	Number of records	Number of unique values		
		Postcodes	Suburbs	Surnames
NT	48,754	28	171	15,887
ACT	115,558	31	132	28,599
TAS	184,158	118	868	20,430
SA	544,562	342	1,304	63,288
WA	653,167	394	1,395	77,325
QLD	1,309,744	432	2,945	110,028
VIC	1,738,216	708	3,030	175,045
NSW	2,323,355	624	4,223	207,403

- Using *'Australia on Disk'* data set (November 2002)
- Randomly selected two times 100 records per data set (as query records)
  1. One single modification in one of the three attributes
  2. One or more modifications in all the three attributes

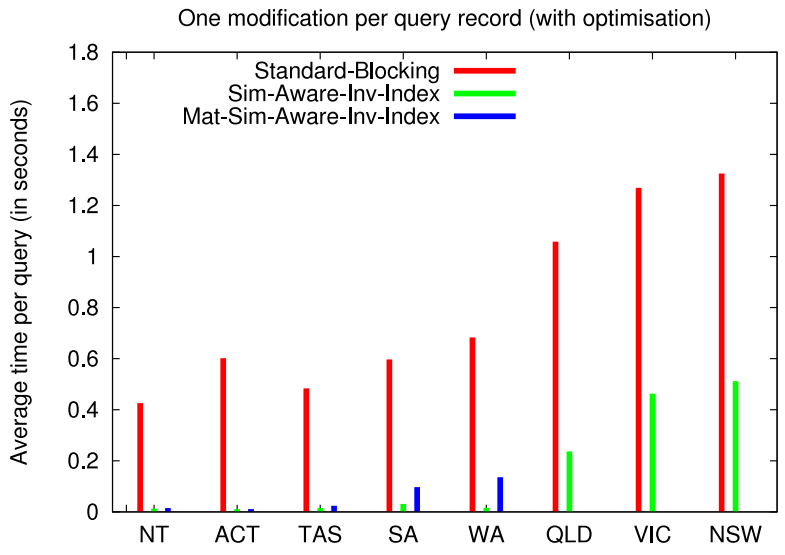
Matching accuracy *(as percentages)*

Australian state/territory	Standard-blocking	Sim-Aware-Inv-Index	Mat-Sim-Aware-Inv-Index
One modification only per record			
NT	97 / 97	99 / 99	97 / 99
ACT	92 / 92	95 / 95	95 / 95
TAS	94 / 94	93 / 93	93 / 93
SA	95 / 95	97 / 97	97 / 97
WA	96 / 96	95 / 95	95 / 95
QLD	98 / 98	94 / 94	–
VIC	95 / 95	92 / 92	–
NSW	91 / 91	87 / 87	–
Three modifications per record			
NT	85 / 85	67 / 66	67 / 66
ACT	78 / 78	60 / 65	60 / 65
TAS	75 / 75	55 / 54	55 / 54
SA	78 / 78	39 / 52	39 / 52
WA	73 / 73	48 / 54	48 / 54
QLD	69 / 69	30 / 41	–
VIC	72 / 72	36 / 56	–
NSW	79 / 79	45 / 65	–

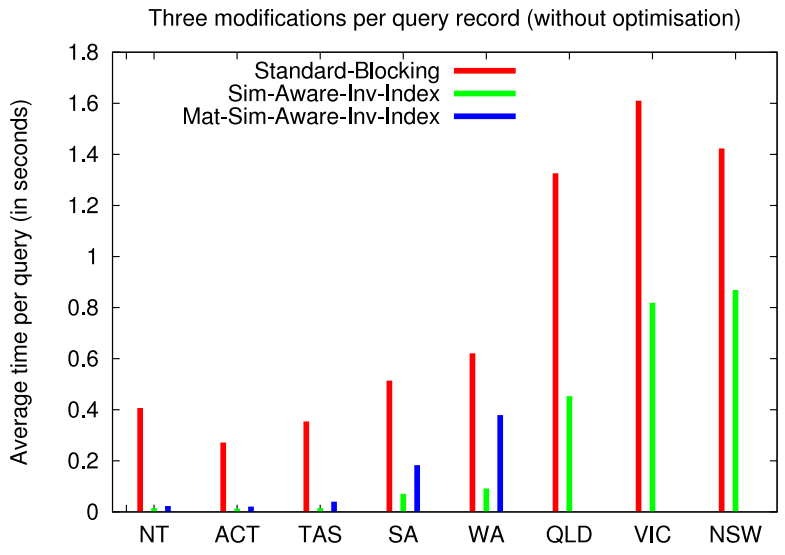
Timing results (1)



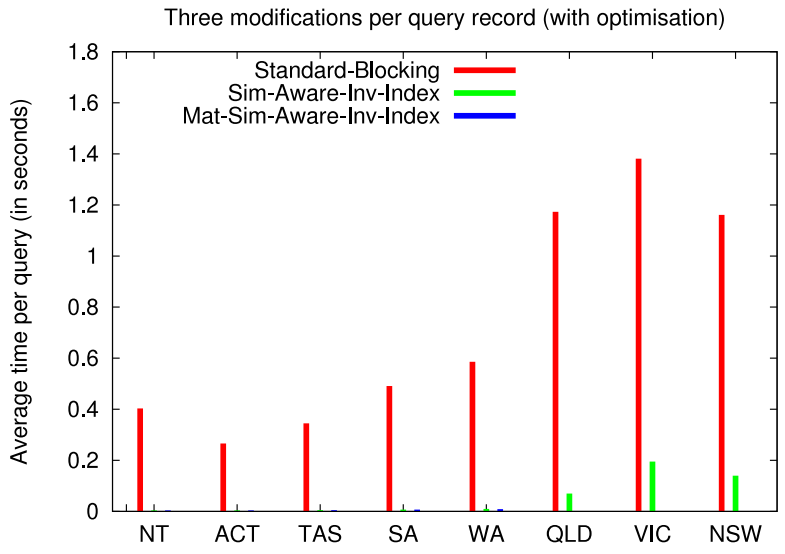
Timing results (2)



Timing results (3)



Timing results (4)



Conclusions

- Real-time entity resolution is of significance in many applications (but not much work done so far)
- We have combined inverted index approaches with similarity calculations
  - Our approach is between 1.3 and 27 (no optimisation) and 2.6 to 100 (with optimisation) times faster than standard blocking
  - However, accuracy is suffering with our approach
- More work needed on optimisation, as well as combining real-time indexing with advanced classification for entity resolution