Automatic Record Linkage using Seeded Nearest Neighbour and SVM Classification

Peter Christen

Department of Computer Science,
ANU College of Engineering and Computer Science,
The Australian National University,
Canberra, Australia

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

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

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

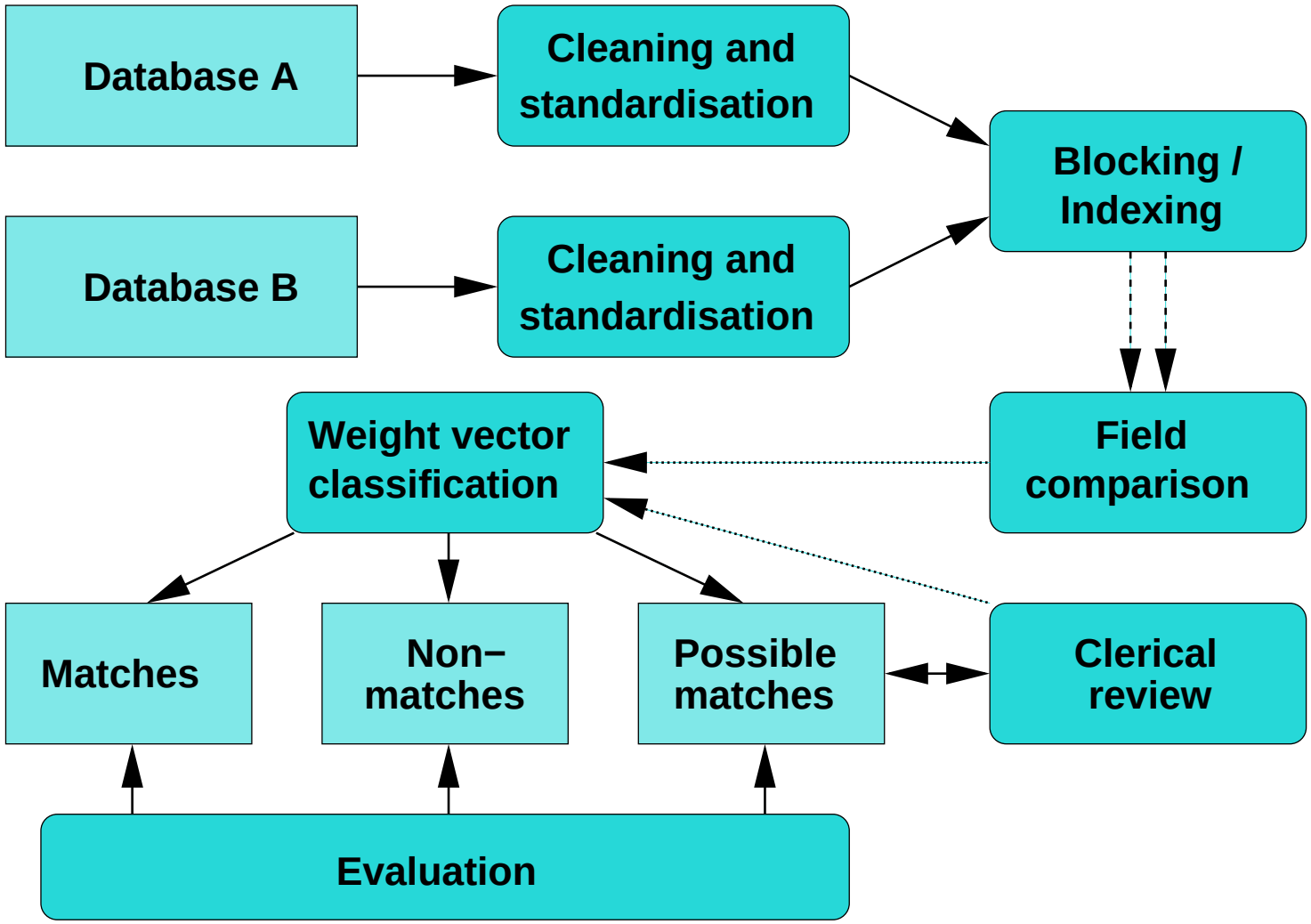
Outline

- Record linkage and its challenges
- The record linkage process
- Record pair comparison and classification
  - Records and weight vectors example
- Two-step classification approach
- Experimental results
- Outlook and future work

Record linkage and its challenges

- The process of linking and aggregating records that represent the same entity (such as a patient, a customer, a business, etc.)
  - Also called *data matching*, *data scrubbing*, *entity resolution*, *object identification*, *merge-purge*, etc.
- Has several major challenges
  - Real world data is dirty (typographical errors and variations, missing and out-of-date values, etc.)
  - Scalability (naïve comparison of all record pairs is $\boldsymbol{O}(n^2)$, so some form of blocking or indexing is required)
  - No training data available in many application areas (no data sets with known true match status)

The record linkage process



Record pair comparison and classification

- Pairs of records are compared field (attribute) wise using various field comparison functions
  - Such as exact or approximate string (edit-distance, *q*-gram, Winkler), numeric, age, date, time, etc.
  - Return 1.0 for exact similarity, 0.0 for total dissimilarity
- For each compared record pair, a *weight vector* containing *matching weights* is calculated
- Record pairs are then classified into *matches*, *non-matches* (and *possible matches*)
  - Various techniques have been explored: Summing and threshold based, decision trees, SVM, clustering, etc.

Records and weight vectors example

R1:	Christine	Smith	42	Main	Street
R2:	Christina	Smith	42	Main	St
R3:	Bob	O'Brian	11	Smith	Rd
R4:	Robert	Bryce	12	Smythe	Road

WV(R1,R2): [0.9, 1.0, 1.0, 1.0, 0.9]
WV(R1,R3): [0.0, 0.0, 0.0, 0.0, 0.0]
WV(R1,R4): [0.0, 0.0, 0.5, 0.0, 0.0]
WV(R2,R3): [0.0, 0.0, 0.0, 0.0, 0.0]
WV(R2,R4): [0.0, 0.0, 0.5, 0.0, 0.0]
WV(R3,R4): [0.7, 0.3, 0.5, 0.7, 0.9]

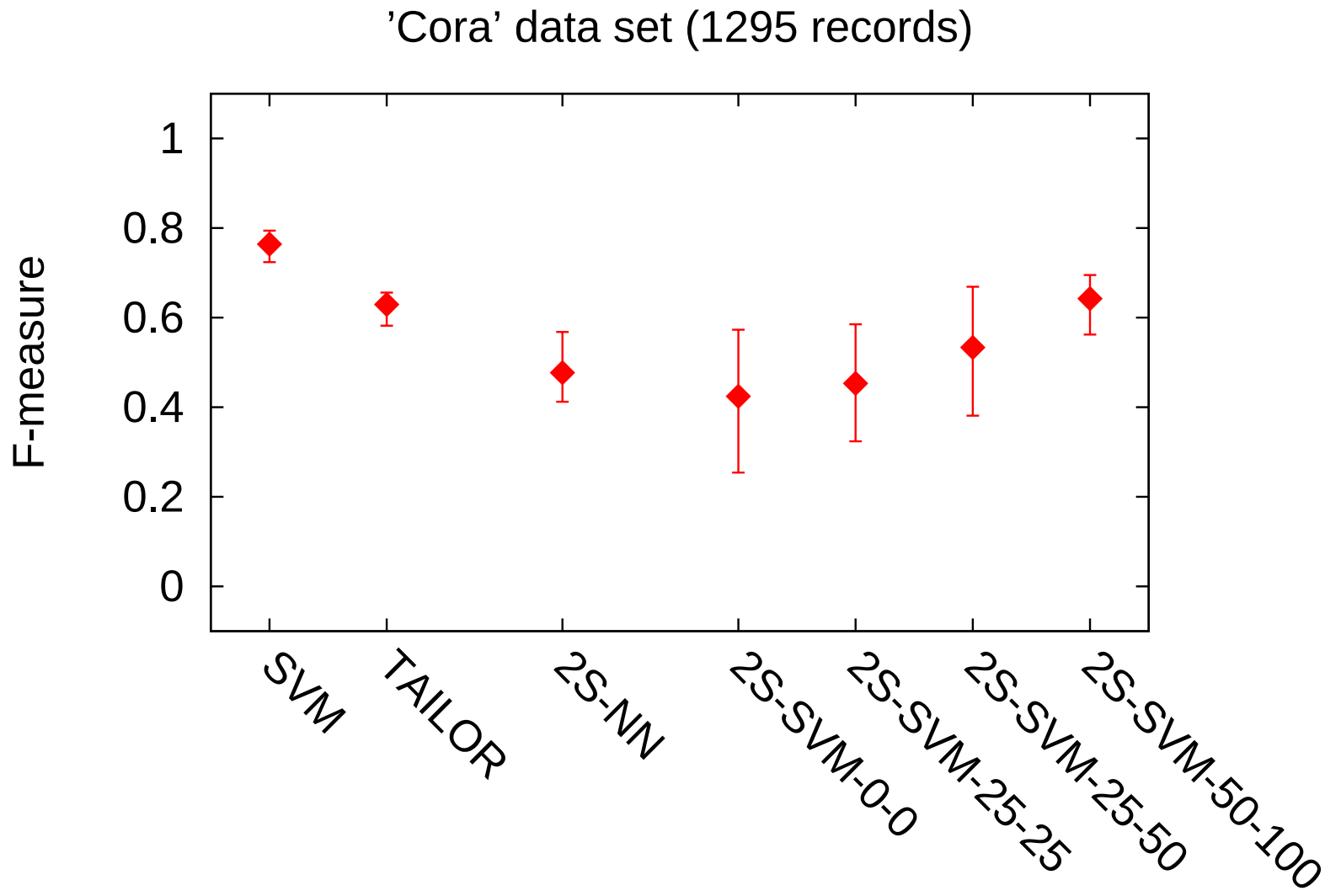
Two-step classification approach

1. Select weight vectors into seed training sets
   - Weight vectors closest to the exact match vector into the *match seed training set*
   - Weight vectors closest to the total dissimilarity weight vector into the *non-match seed training set*
2. Start binary classification using seed training sets
   - Nearest neighbour: Iteratively add not yet classified weight vector closest to a training set into it
   - Iterative SVM: Train an SVM, then add the weight vectors furthest away from the decision boundary into the training sets, then train a new SVM

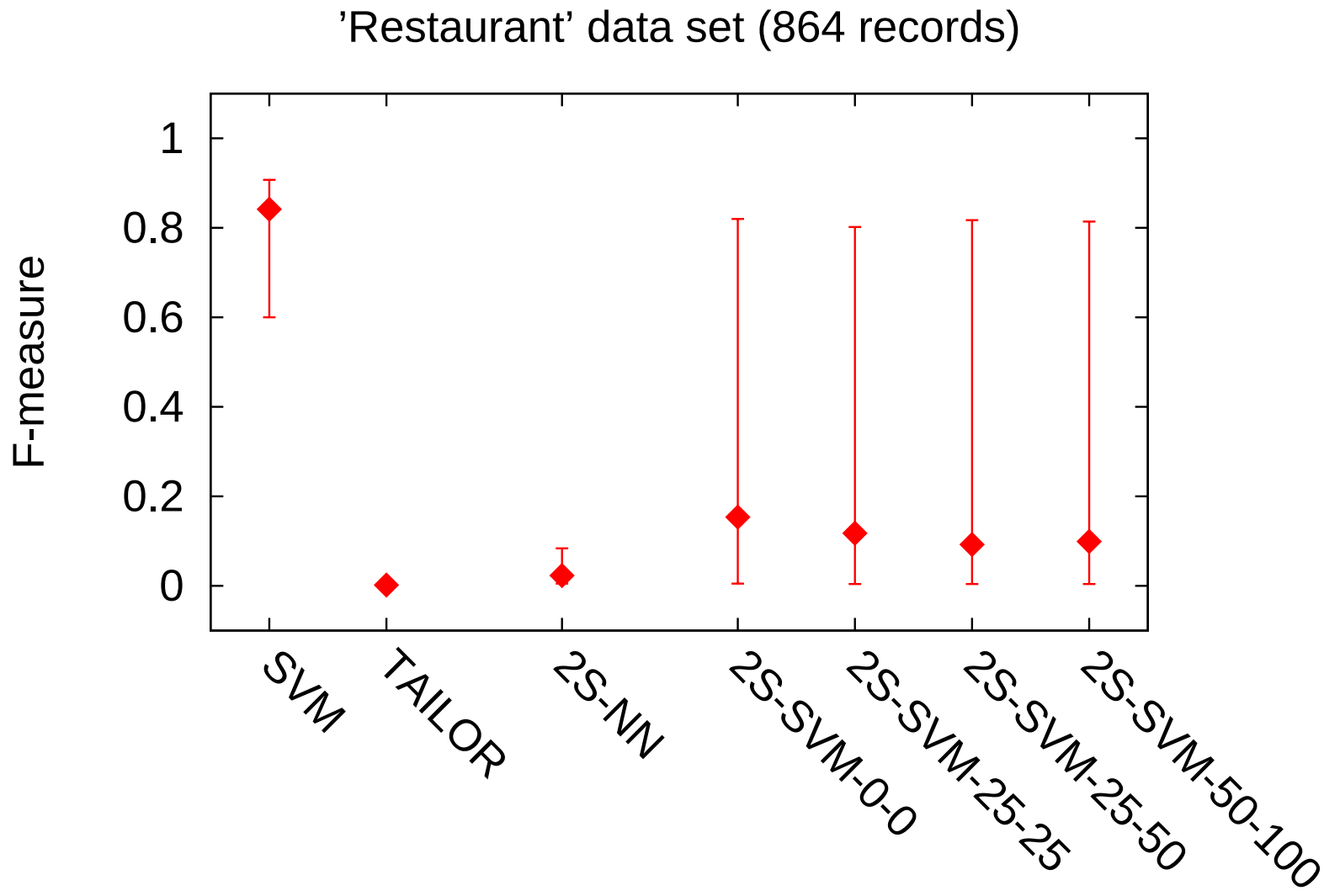
Experimental results

- All techniques are implemented in the *Febrl* open source record linkage system
  (available from: **https://sourceforge.net/projects/febrl/**)
- Experiments using both real and synthetic data
  (*Secondstring* repository and *Febrl* data set generator)
- The proposed two-step approach is compared with two other classifiers
  - Support vector machine (SVM) (supervised)
  - Hybrid TAILOR approach (k-means followed by SVM)
- *F*-measure used to evaluate classifier results
  (minimum, average and maximum values shown in graphs)

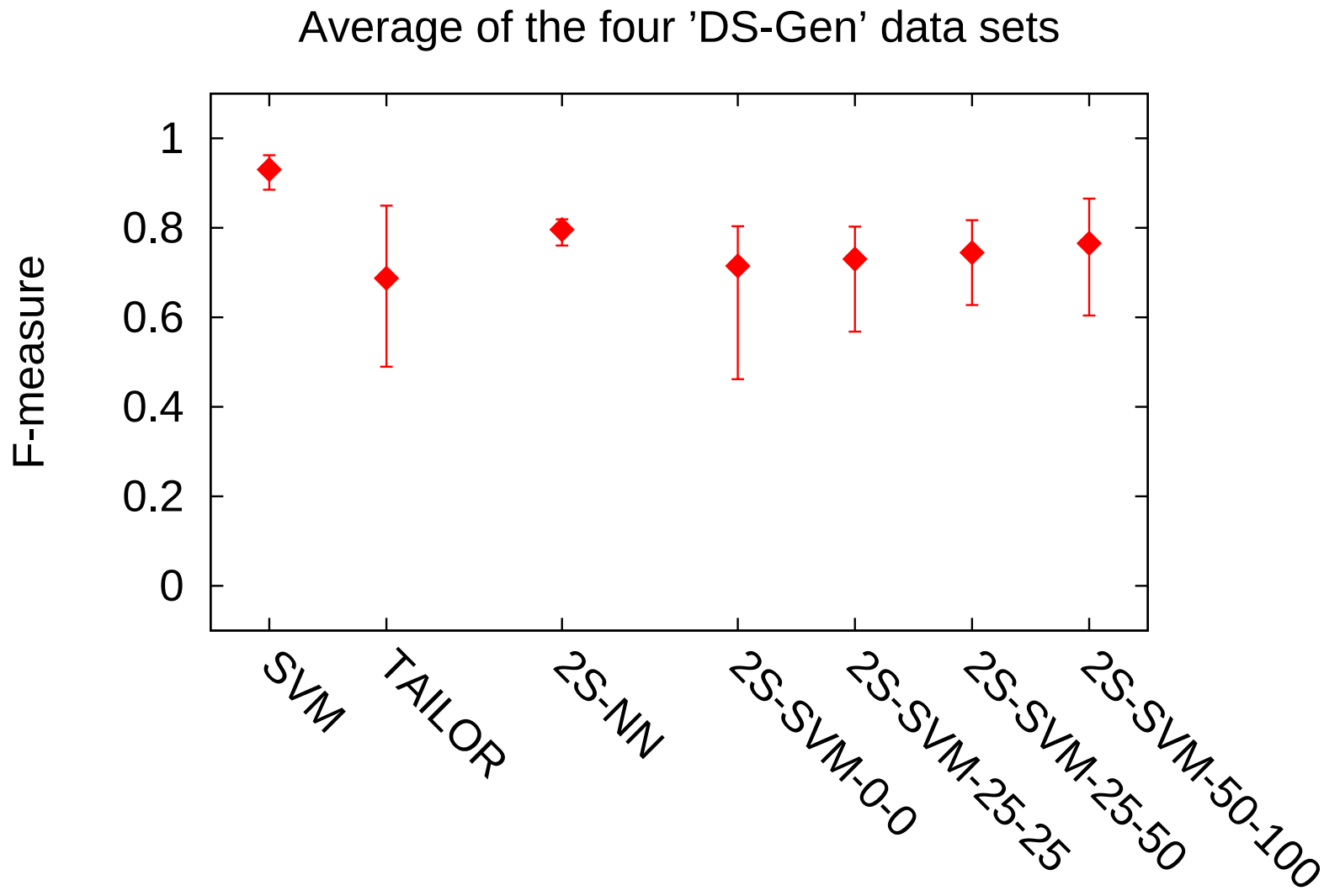
Classification results for 'Cora'



Classification results for ‘Restaurant’



Results for synthetic data sets



Outlook and future work

- The proposed two-step record pair classification approach shows promising results
  - Can automatically select good quality training examples
  - Can achieve better results than other unsupervised classification techniques
- Improvements for second step (classification)
  - Implement data reduction and fast indexing techniques to improve performance and scalability
  - Investigate how this approach can be combined with active learning
- Conduct more experiments on larger data sets