Framework for Evaluating Clustering Algorithms in Duplicate Detection

Oktie Hassanzadeh,
Fei Chiang,
Hyun Chul Lee,
Renée J. Miller
Database Group
University of Toronto

August 2009

Presentation at VLDB 2009 Conference - Lyon, France

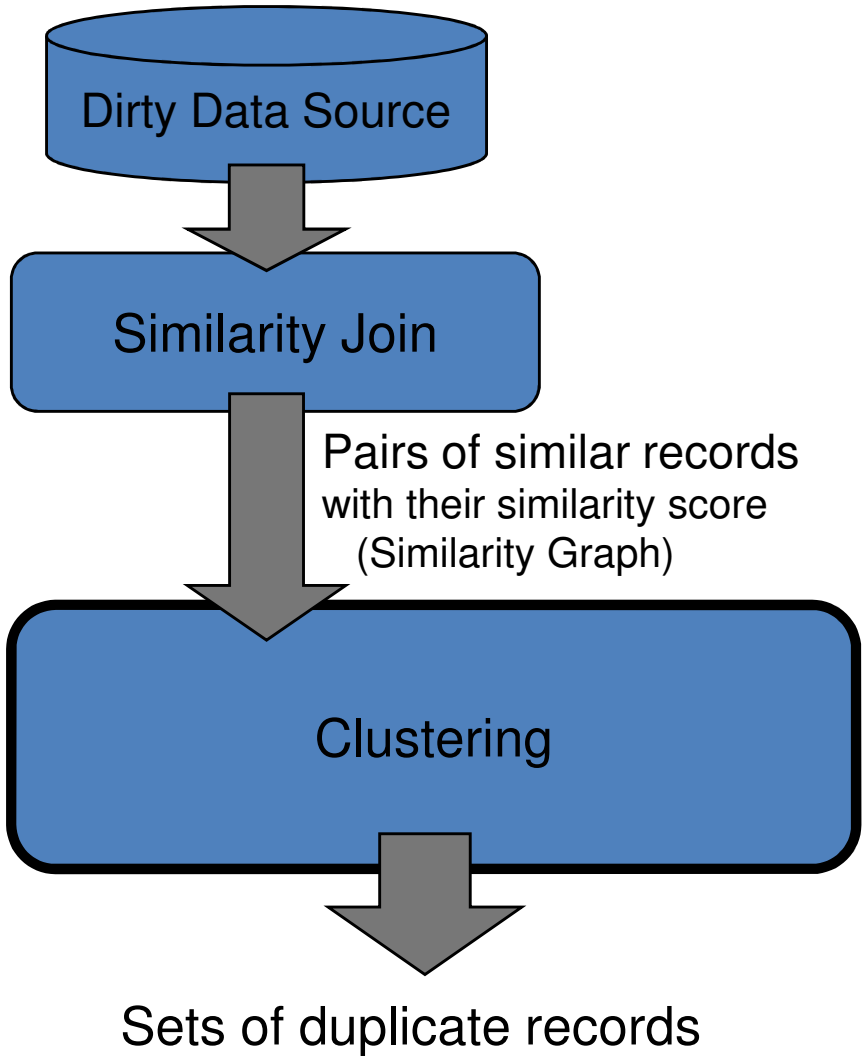
Outline

- Stringer Duplicate Detection Framework
- Overview of the Clustering Algorithms
- Evaluation Framework
- Summary and Conclusion

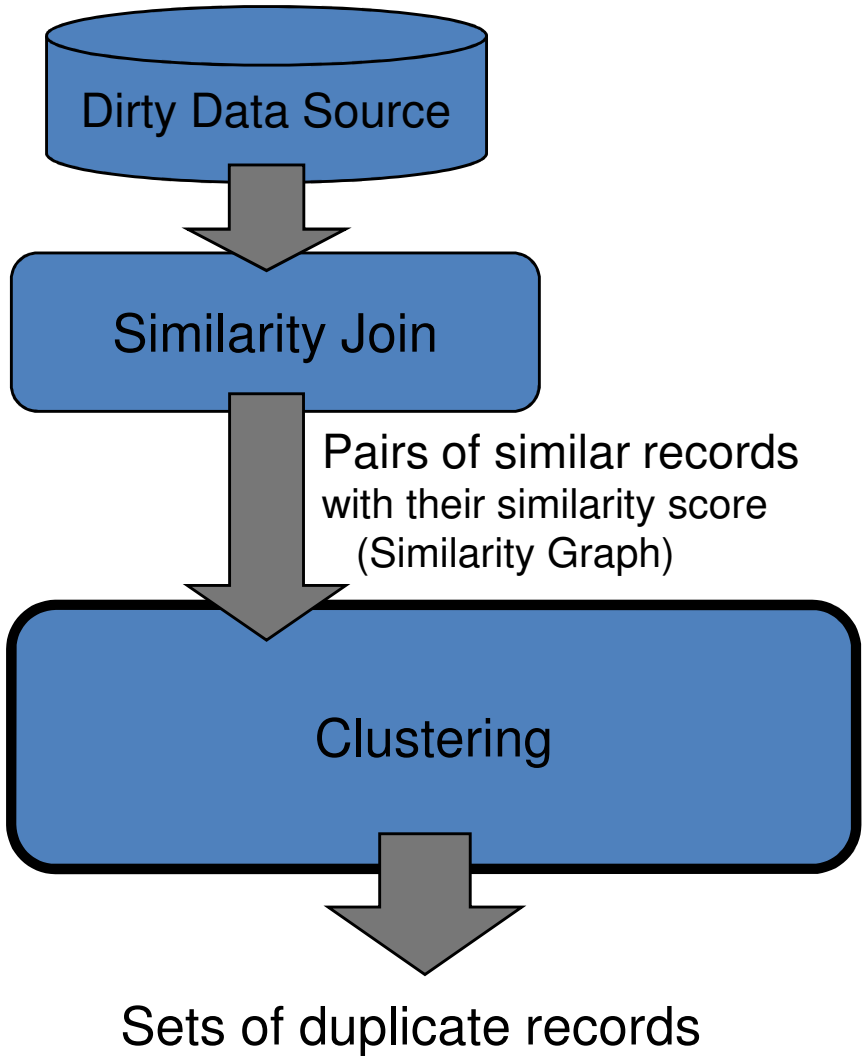
Outline

- **Stringer Duplicate Detection Framework**
- Overview of the Clustering Algorithms
- Evaluation Framework
- Summary and Conclusion

Duplicate Detection Framework



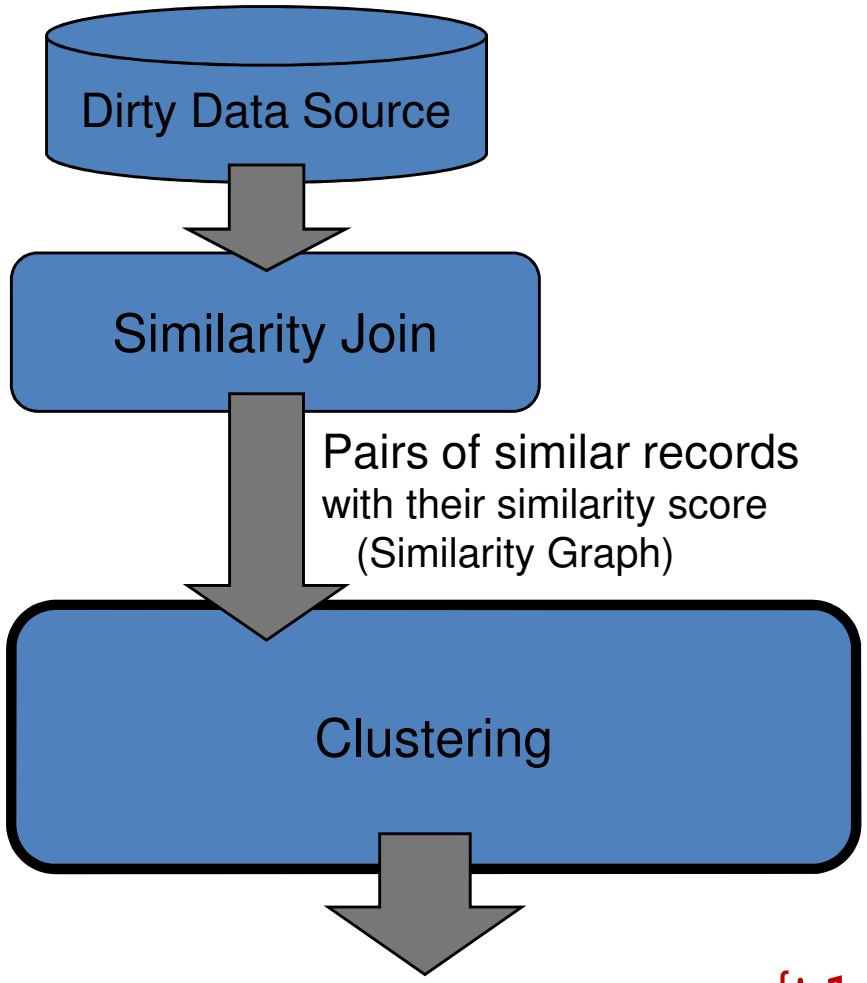
Duplicate Detection Framework



Dirty Data Source

tid	string
t1	Microsoft Corp.
t2	Macrosoft Corp.
t3	Microsoft Corporation
t4	Macromedia Inc.
t5	Macromedia Corp.
t6	IBM Inc.
t7	IBM Corporation
t8	IBM Corp.

Duplicate Detection Framework



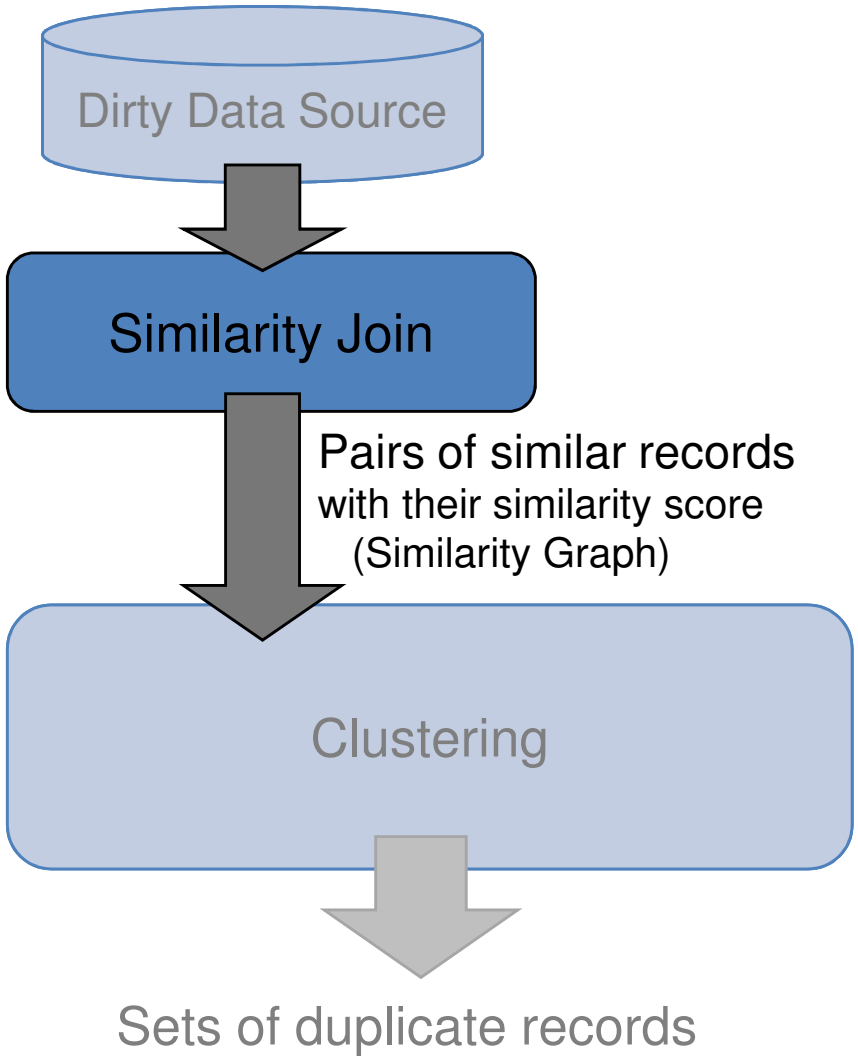
Dirty Data Source

tid	string
t1	Microsoft Corp.
t2	Macrosoft Corp.
t3	Microsoft Corporation
t4	Macromedia Inc.
t5	Macromedia Corp.
t6	IBM Inc.
t7	IBM Corporation
t8	IBM Corp.

{t1, t2, t3}, {t4, t5}, {t6, t7, t8}

UofT:DB Group

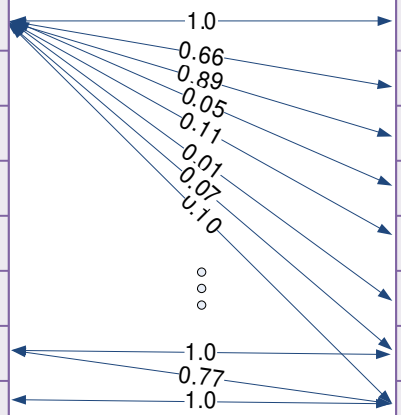
Duplicate Detection Framework



Similarity Join

- Similarity (Approximate) Join
  - Input: Two relations of string records $R = \{r_i : 1 \le i \le N_1\}$ and $S = \{s_i : 1 \le j \le N_2\}$
  - Output: pairs $(r_i, s_j) \in R \times S$ where r_i and s_j are *similar* records
  - Two records are *similar* if $sim(r_i, s_j) \ge \theta$ for some string similarity function sim() and a threshold θ

tid	string
t1	Microsoft Corp.
t2	Macrosoft Corp.
t3	Microsoft Corporation
t4	Macromedia Inc.
t5	Macromedia Corp.
t6	IBM Inc.
t7	IBM Corporation
t8	IBM Corp.



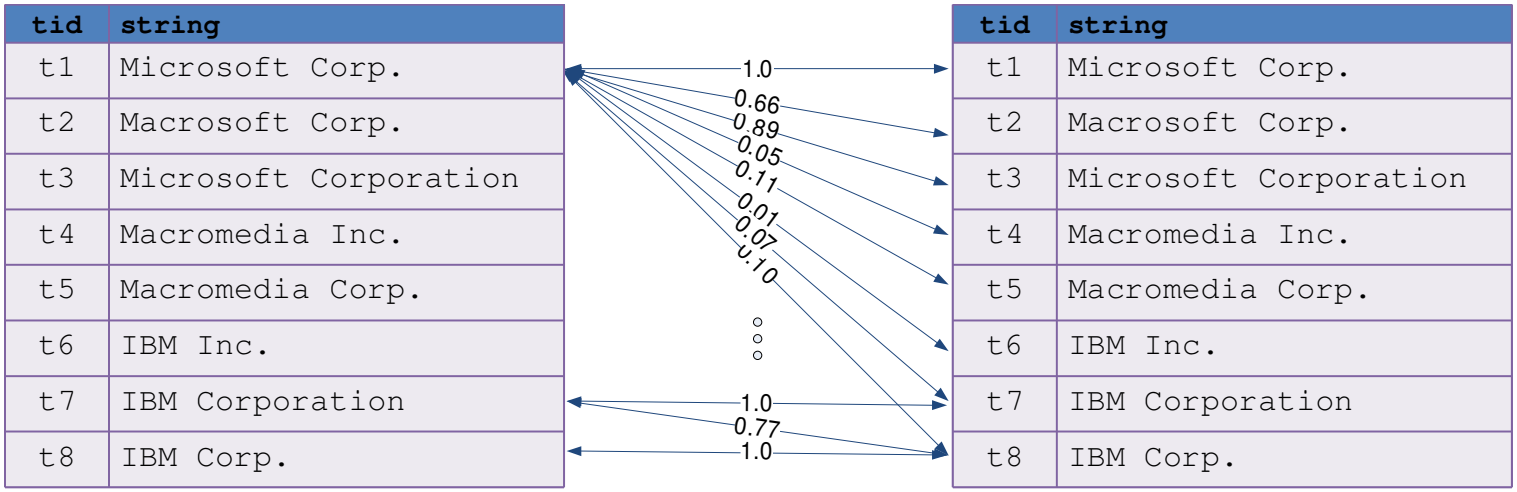
tid	string
t1	Microsoft Corp.
t2	Macrosoft Corp.
t3	Microsoft Corporation
t4	Macromedia Inc.
t5	Macromedia Corp.
t6	IBM Inc.
t7	IBM Corporation
t8	IBM Corp.

if θ=0.6 => (t1,t1), (t1,t2), (t1,t3), … will be in the output

UofT:DB Group

Similarity Join

- Similarity (Self) Join
  - Input: A relation of string records $R = \{r_i : 1 \leq i \leq N_1\}$
  - Output: pairs $(r_i, r_j) \in R \times R$ where r_i and s_j are *similar* records and $i<j$
  - Two records are *similar* if $sim(r_i, r_j) \geq \theta$ for some string similarity function sim() and a threshold θ

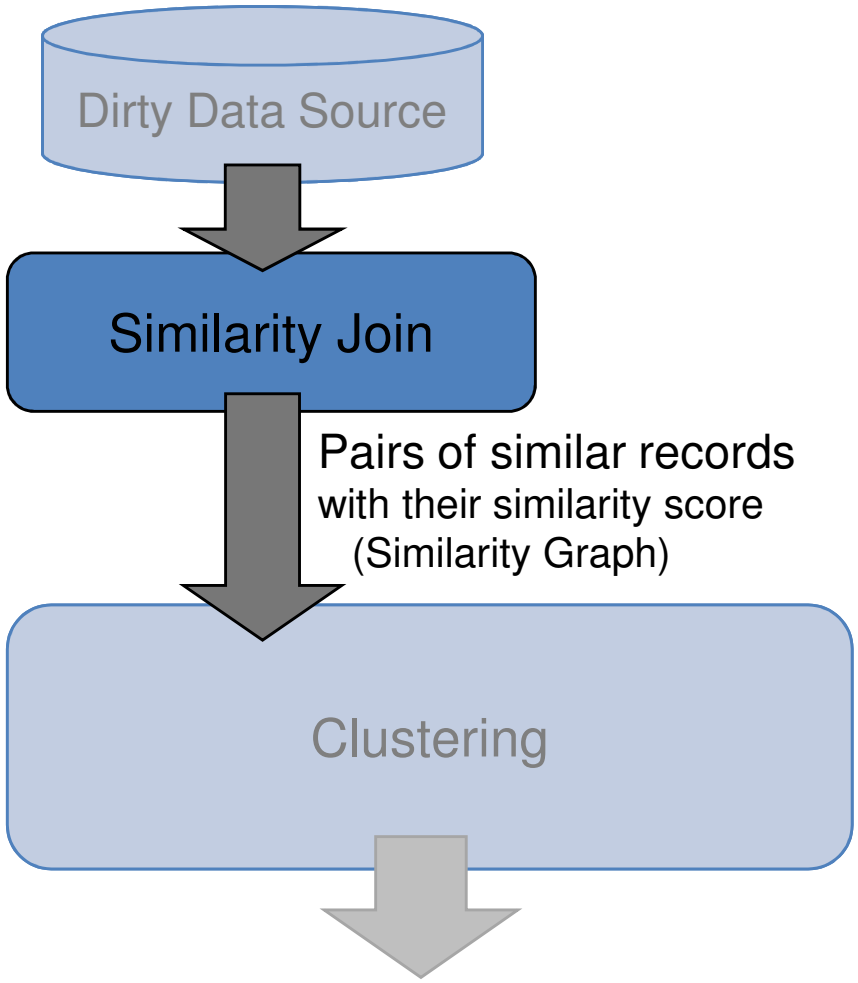


if θ=0.6 => (t1,t2), (t1,t3), … will be in the output

Similarity Join – Related Work

- Work on efficiency
  - Blocking, indexing and hashing techniques
    - Efficient Set-Similarity Joins [SK-SIGMOD'04], [AGK-VLDB'06]
    - All-Pairs Similarity Search [BMS-WWW07]
    - Cluster Pruning [CPR+-PODS07]
    - Variable-length Grams [LWY-VLDB'07]
    - …
- Work on accuracy
  - Benchmarking string similarity predicates
    - For approximate selection [CHK+-SIGMOD'07]
    - Extension to similarity join [HSM-QDB'07]

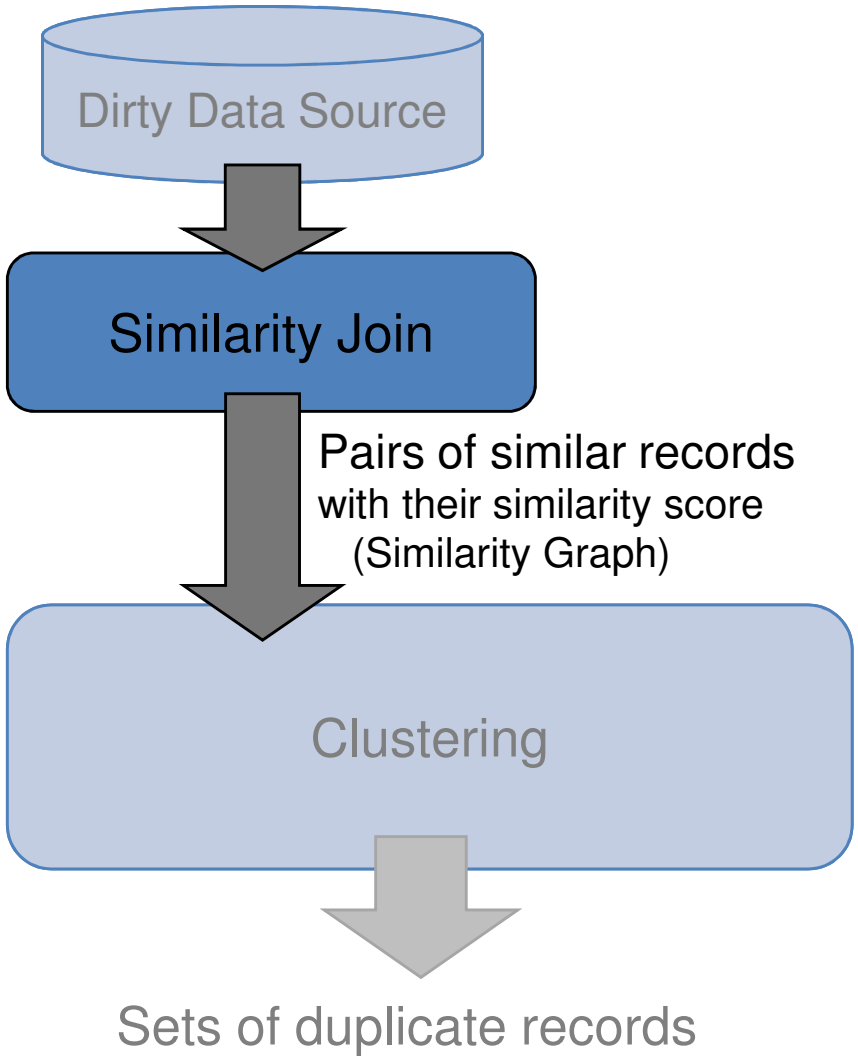
Duplicate Detection Framework



Dirty Data Source

tid	string
t1	Microsoft Corp.
t2	Macrosoft Corp.
t3	Microsoft Corporation
t4	Macromedia Inc.
t5	Macromedia Corp.
t6	IBM Inc.
t7	IBM Corporation
t8	IBM Corp.

Duplicate Detection Framework



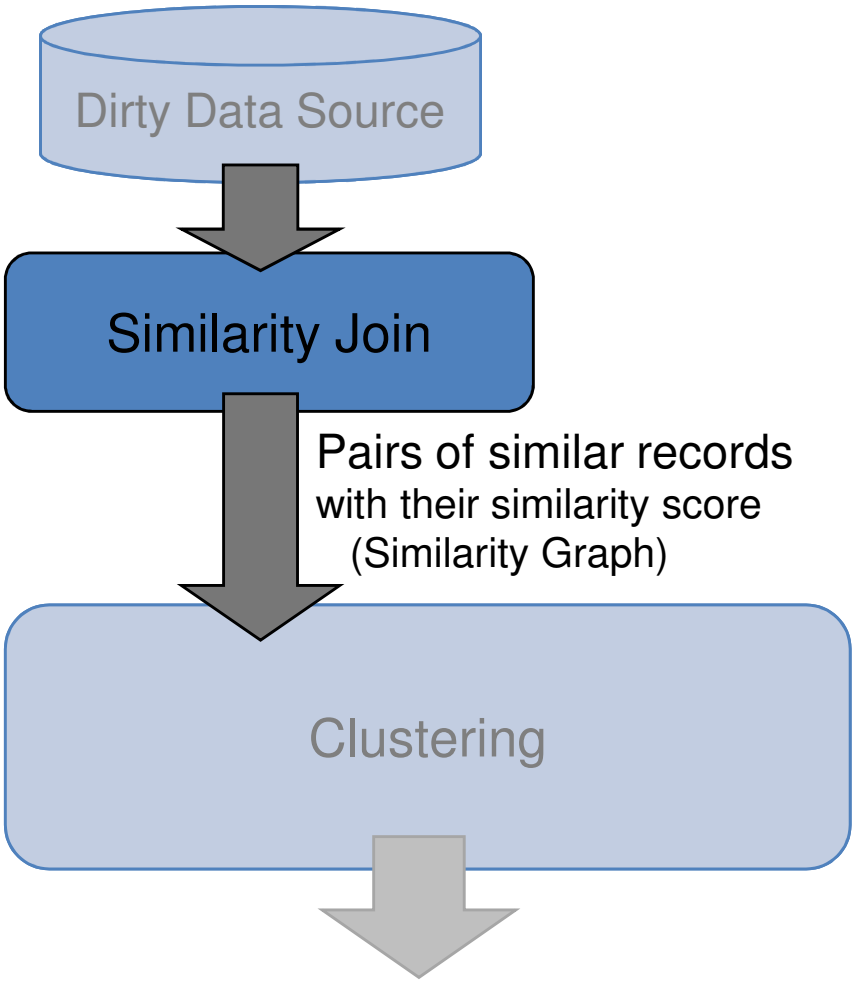
Similarity Scores

tid1	tid2	sim(tid1,tid2)
t1	t2	0.66
t1	t3	0.89
t1	t5	0.11
t2	t3	0.59
t2	t4	0.16
t2	t5	0.23
t3	t7	0.14
t4	t5	0.65
t4	t6	0.10
t6	t7	0.26
t6	t8	0.30
t7	t8	0.77

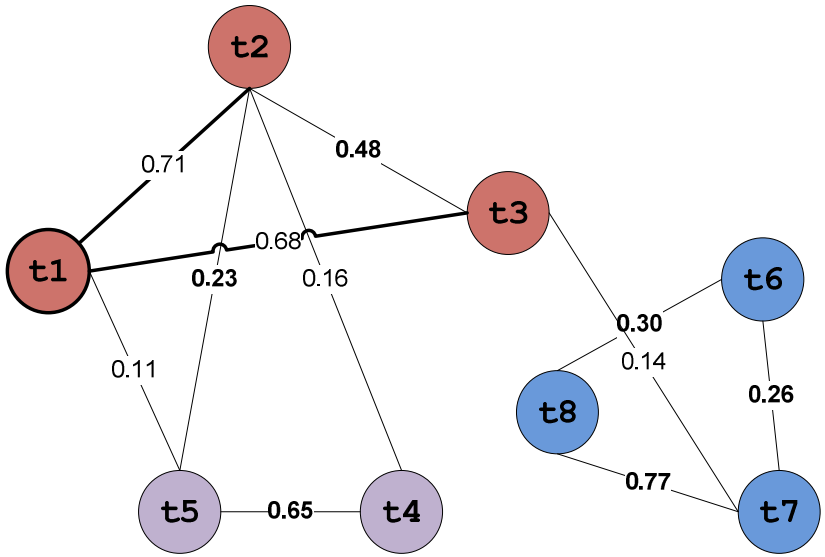
θ (similarity threshold) = 0.1

UofT:DB Group

Duplicate Detection Framework



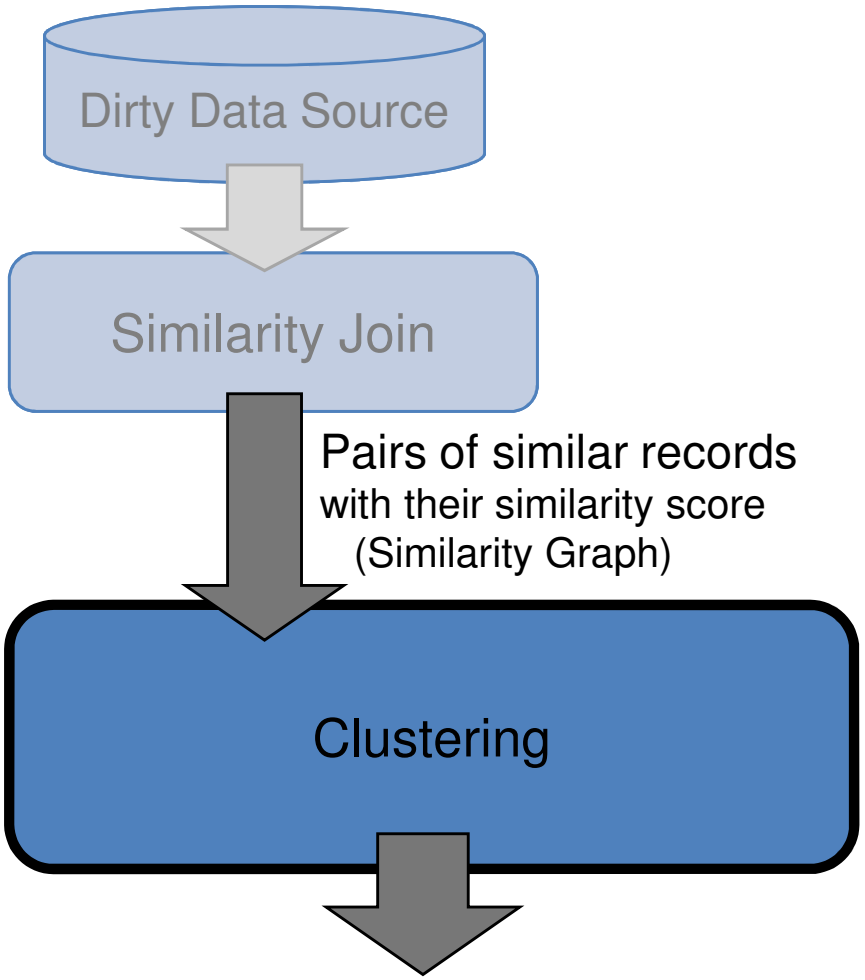
Similarity Graph



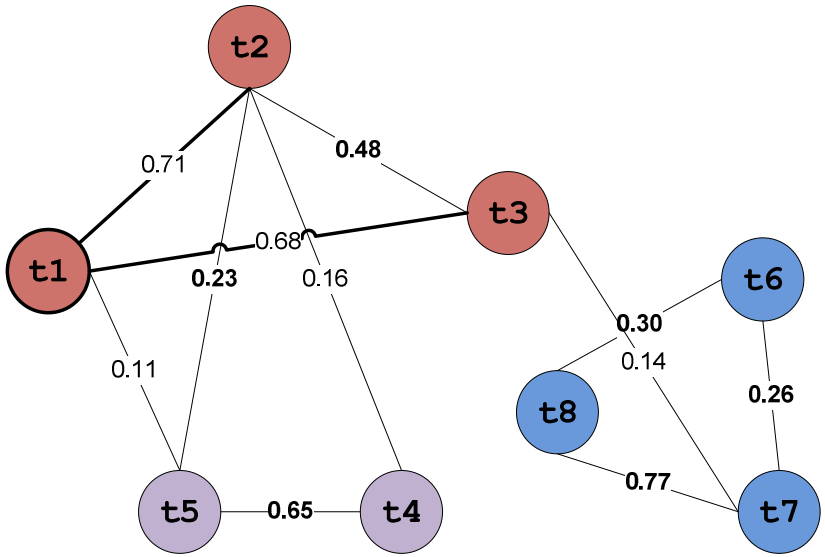
θ (similarity threshold) = 0.1

UofT:DB Group

Duplicate Detection Framework

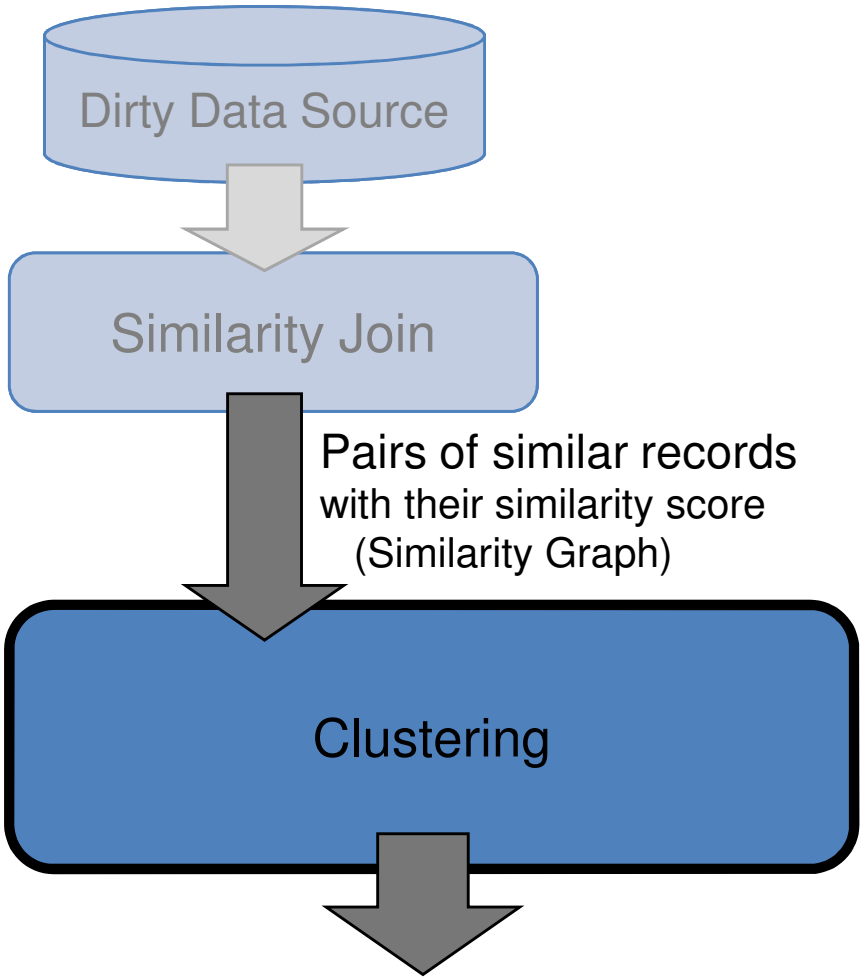


Similarity Graph

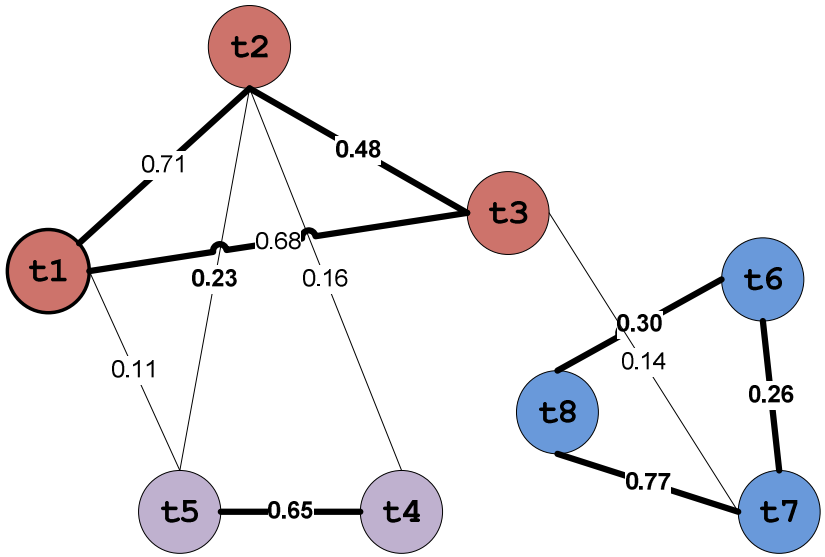


UofT:DB Group

Duplicate Detection Framework

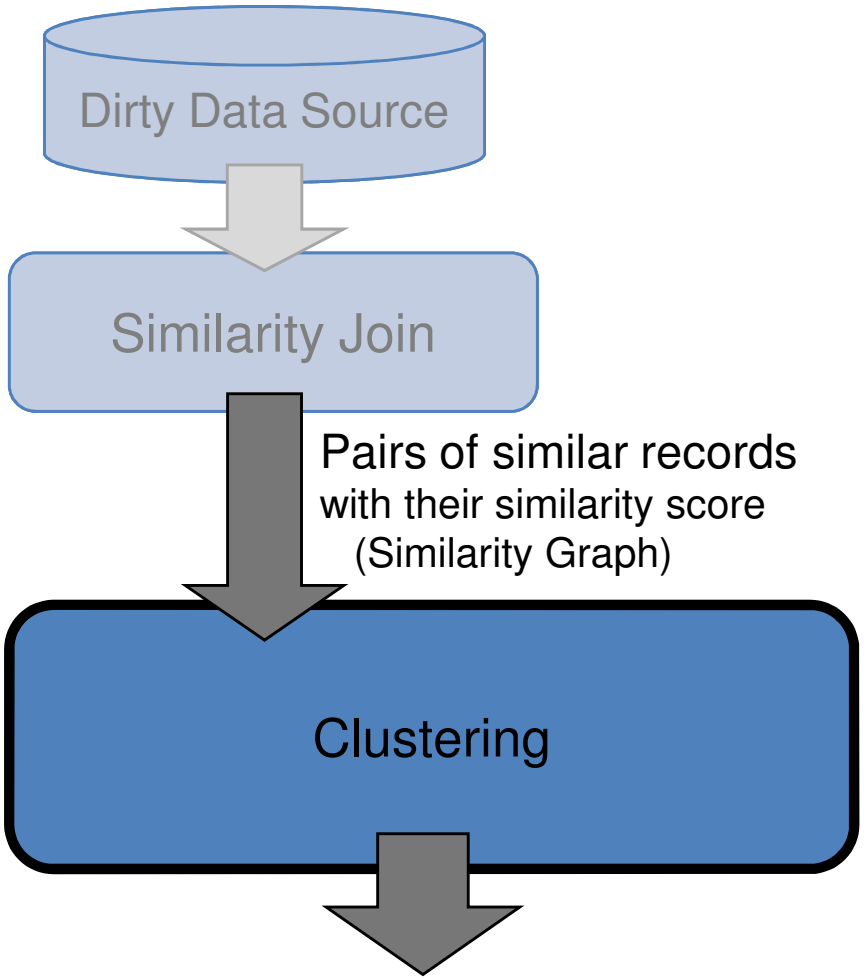


Similarity Graph

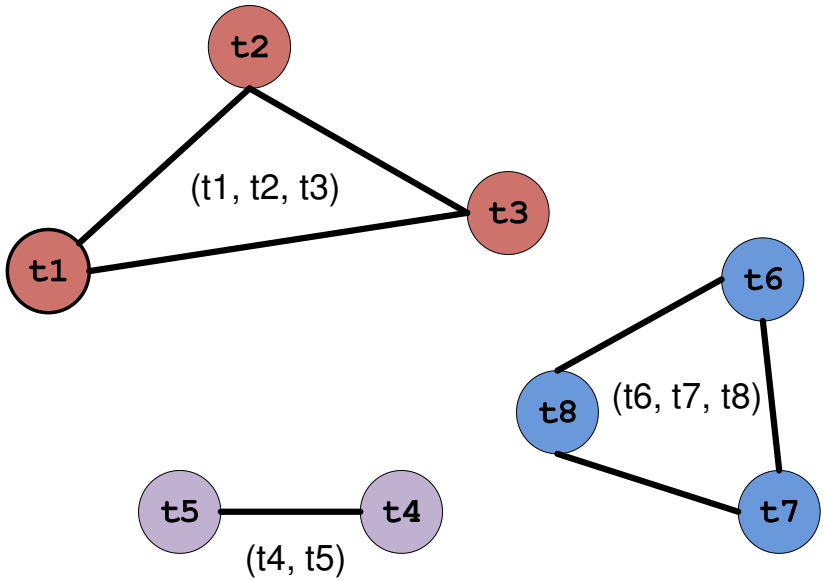


UofT:DB Group

Duplicate Detection Framework



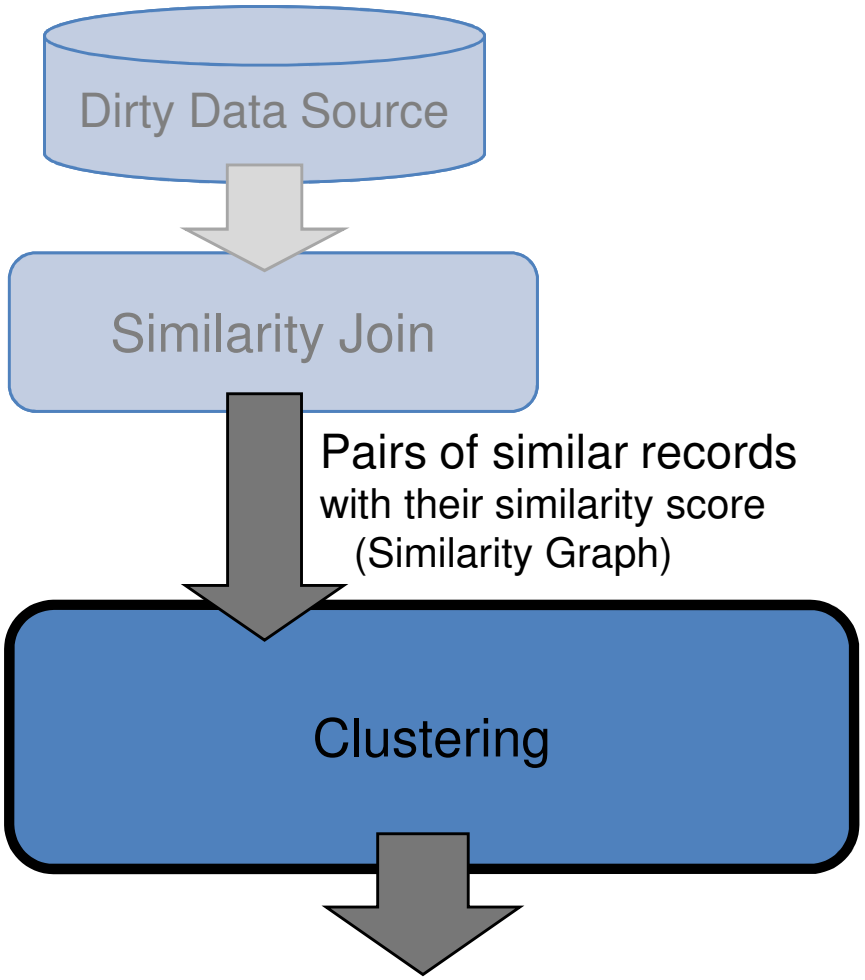
Groups of Duplicates

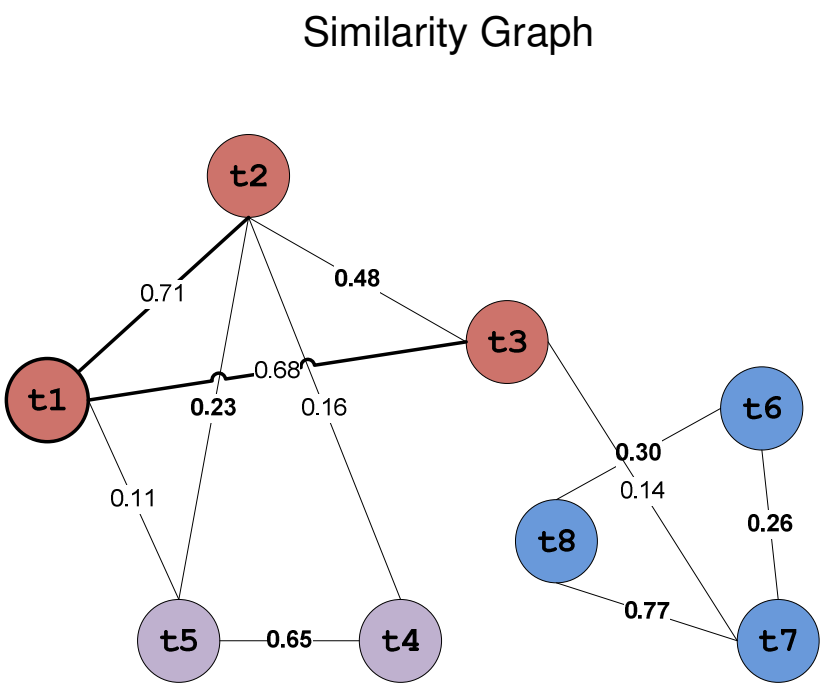


Common Approach in Deduplication

- Transitive closure
  - Finding 'components' in the similarity graph
- Problem
  - May put together many dissimilar records (low threshold)
  - May split many similar records (high thresholds)
  - Very sensitive to the value of the threshold used for the similarity join

Partitioning the Similarity Graph

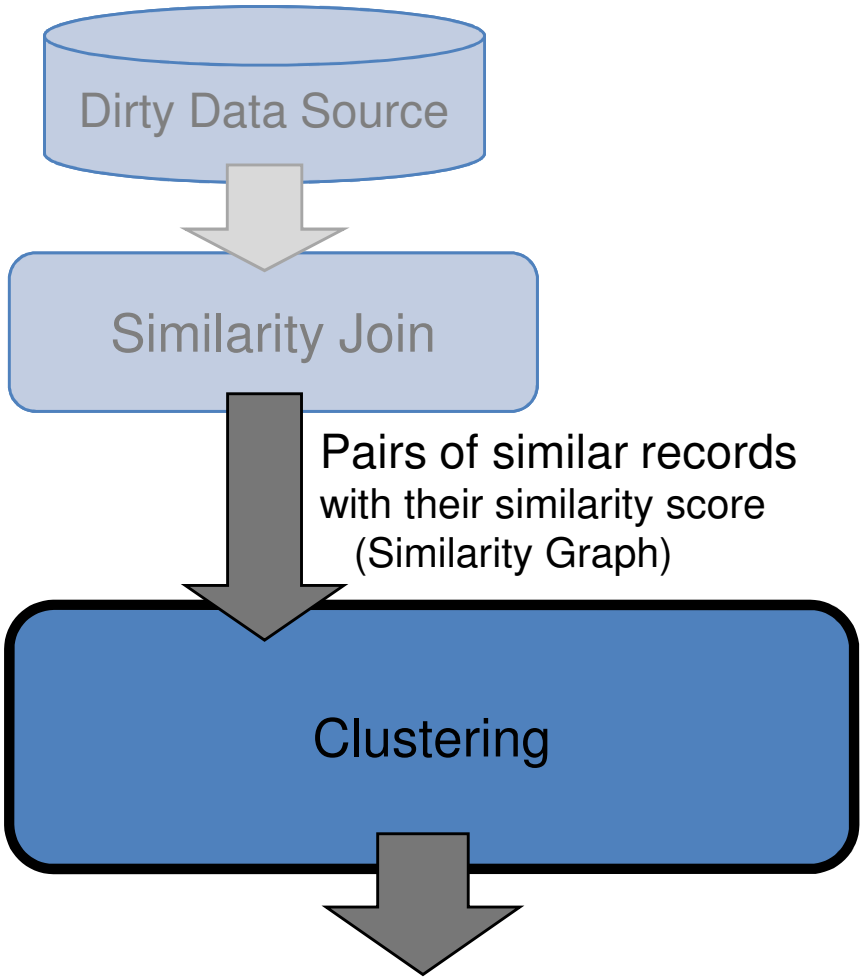


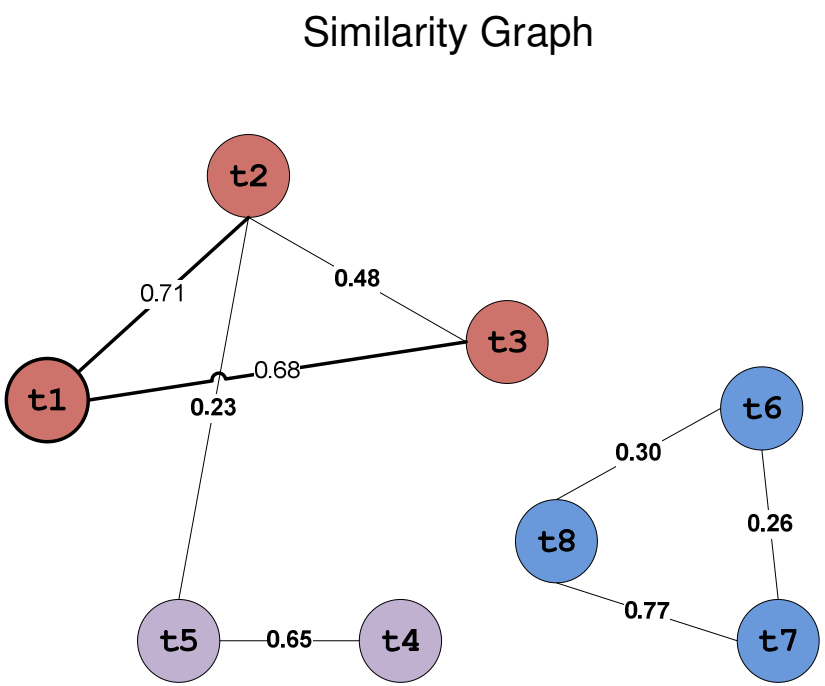


θ (similarity threshold) = 0.1

UofT:DB Group

Partitioning the Similarity Graph

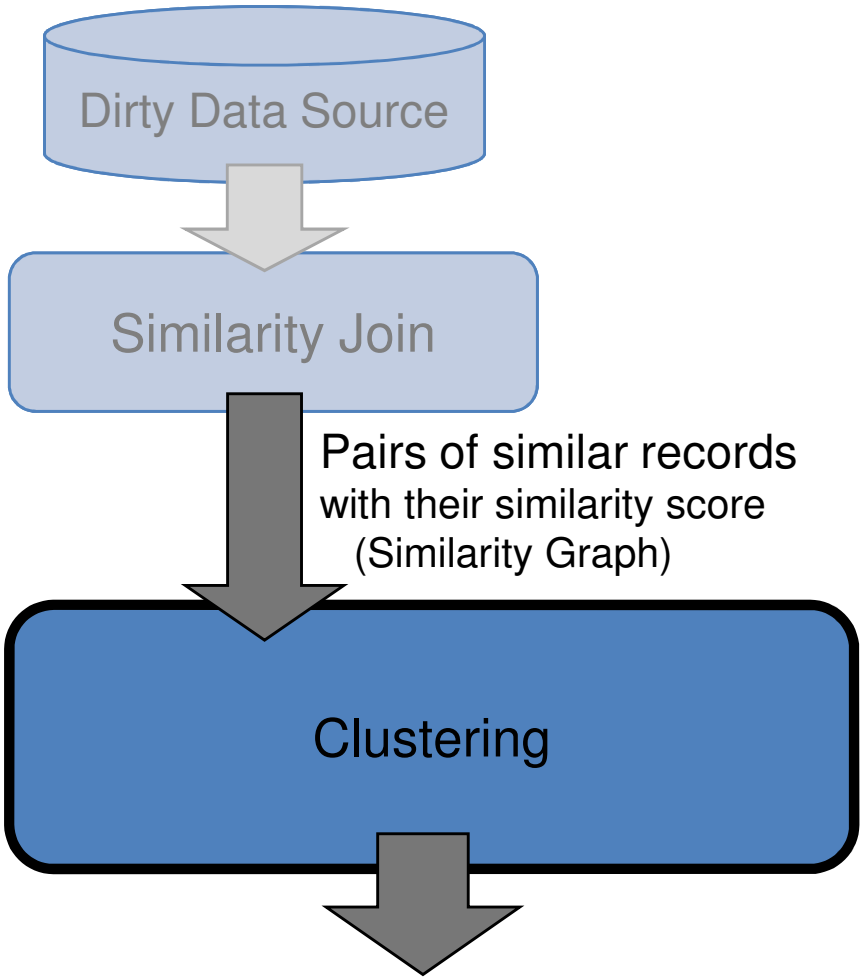


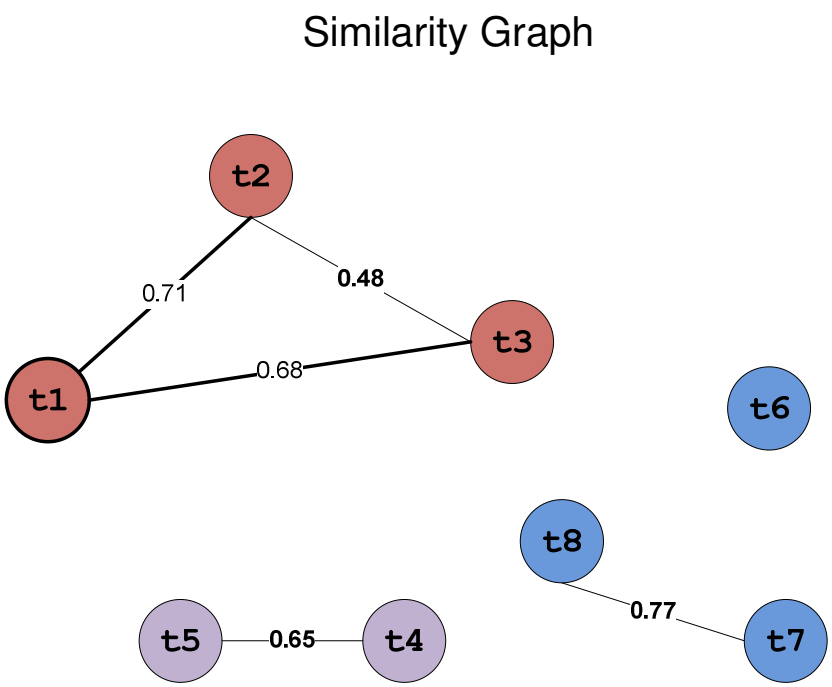


θ (similarity threshold) = 0.2

UofT:DB Group

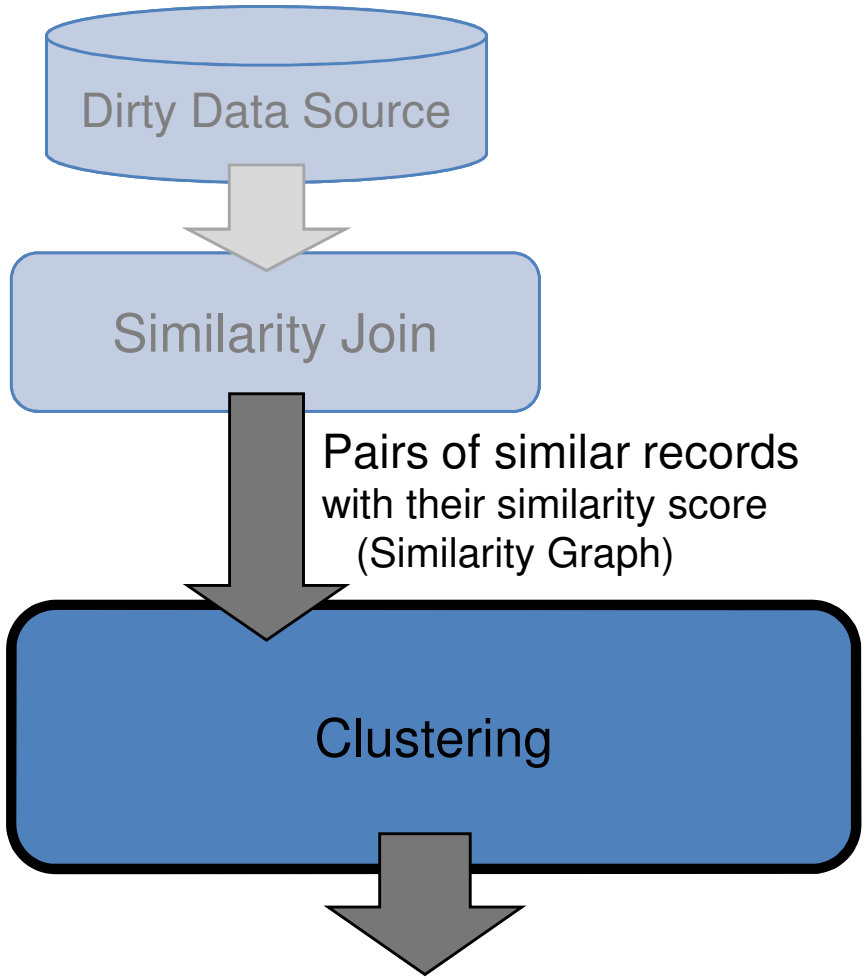
Partitioning the Similarity Graph

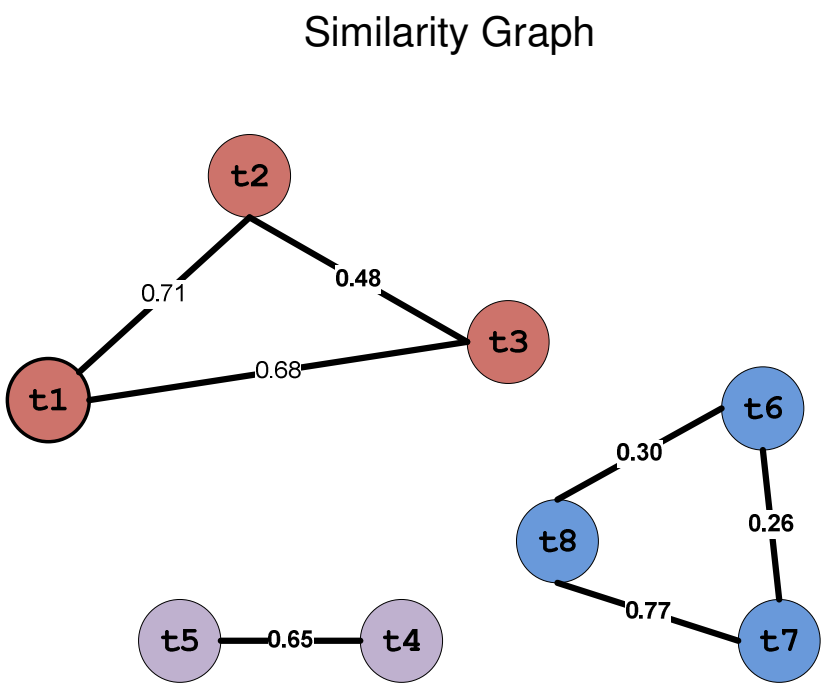




θ (similarity threshold) = 0.4

Clustering the Similarity Graph





θ (similarity threshold) = 0.2

The result of MERGE-CENTER clustering algorithm

Outline

- Stringer Duplicate Detection Framework
- **Overview of the Clustering Algorithms**
- Evaluation Framework
- Summary and Conclusion

Clustering Algorithms

- A wealth of literature on clustering algorithms
  - What we need
    - Partitional and disjoint algorithms
      - overlapping may also be desirable
  - Goal: Sets of clusters that:
    - maximize the intra-cluster weights
    - minimize the inter-cluster edge weights

Clustering Algorithms

- A wealth of literature on clustering algorithms
  - What we need
    - Most important feature
      - "Unconstrained algorithms"
    - I.e. , algorithms that do not require as input:
      - The number of clusters
      - The diameter of the clusters
      - Any other domain specific parameters
    - Algorithms need to be able to predict the correct number of clusters

Clustering Algorithms

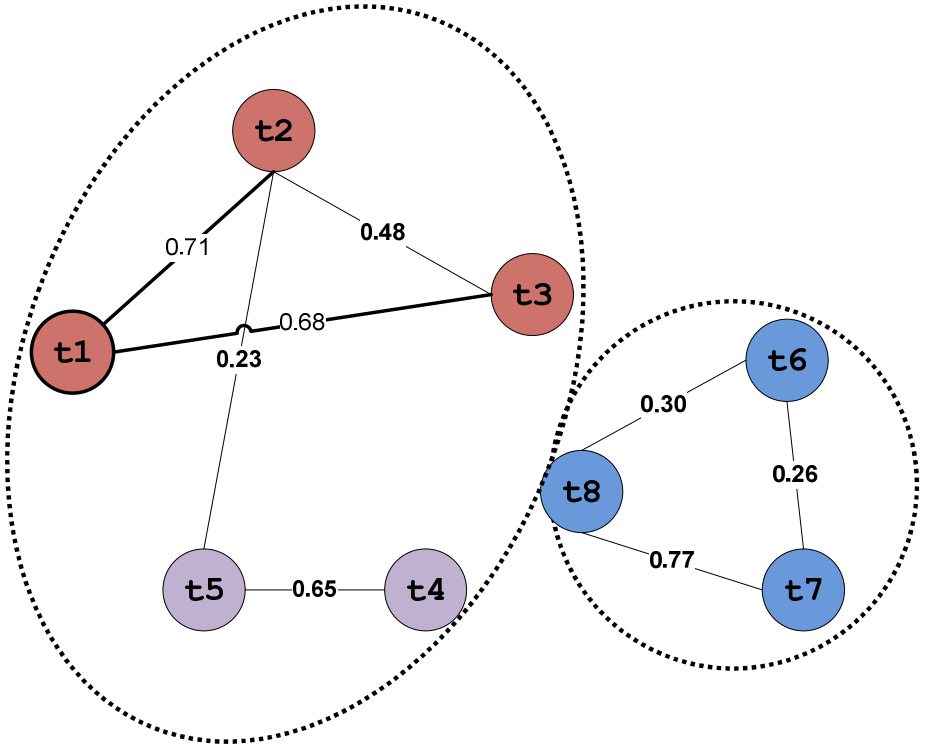
- A wealth of literature on clustering algorithms
  - What we need
    - Need to scale well
    - Robust with respect to the characteristics of the data
      - E.g., distribution of the duplicates
    - Capable of finding small clusters and singletons
      - Different from many clustering algorithms
        - E.g., algorithms proposed for image segmentation

Single-pass Algorithms

- Perform clustering by a single scan of the output of the similarity join (the edges of the graph)
  - Partitioning
    - Transitive Closure
  - CENTER [HGI-WebDB'00]
  - MERGE-CENTER [HM-VLDBJ09]

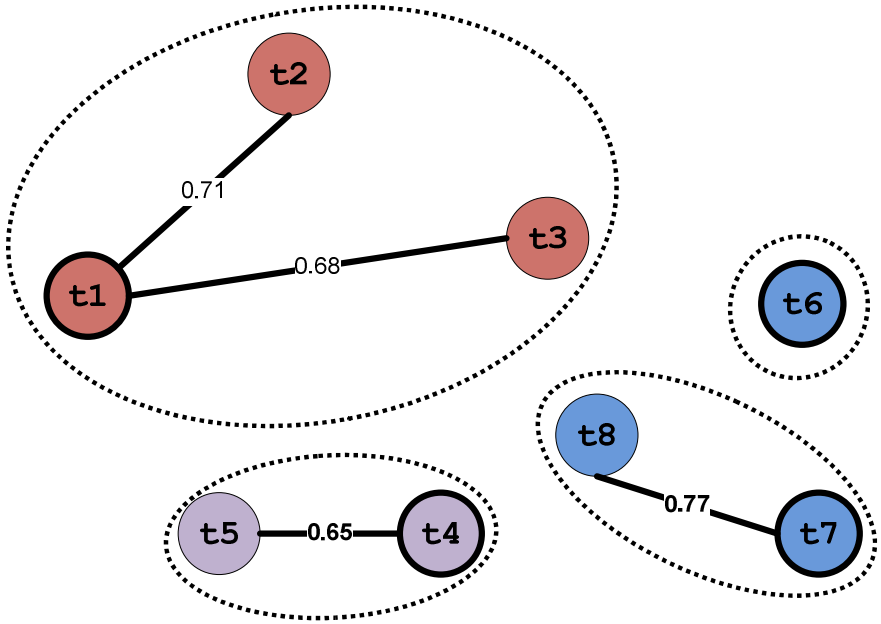
Single-pass Algorithms

- Perform clustering by a single scan of the output of the similarity join (the edges of the graph)
  - Partitioning
    - Transitive Closure
  - CENTER [HGI-WebDB'00]
  - MERGE-CENTER [HM-VLDBJ09]



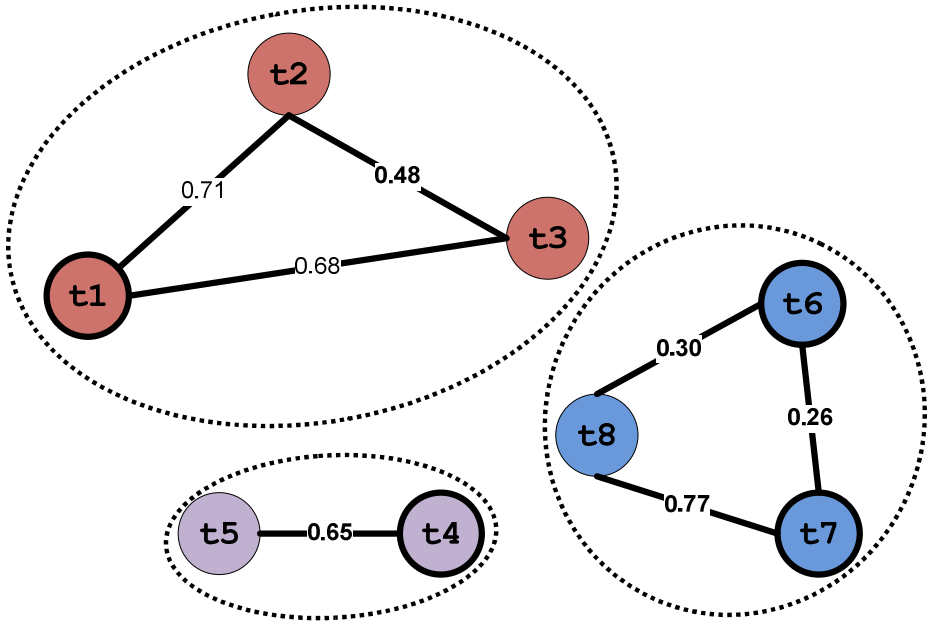
Single-pass Algorithms

- Perform clustering by a single scan of the output of the similarity join (the edges of the graph)
  - Partitioning
    - Transitive Closure
  - CENTER [HGI-WebDB'00]
  - MERGE-CENTER [HM-VLDBJ09]



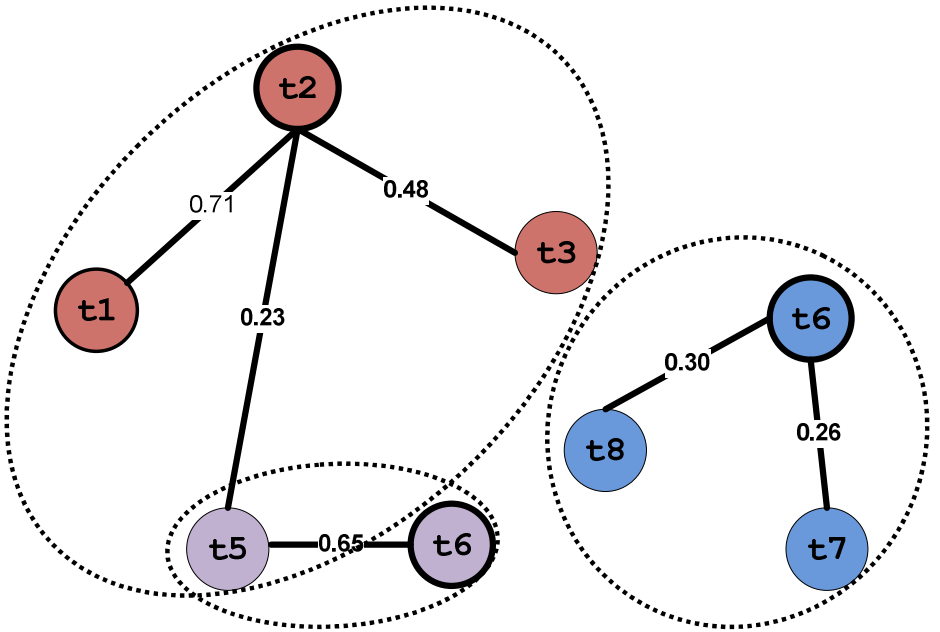
Single-pass Algorithms

- Perform clustering by a single scan of the output of the similarity join (the edges of the graph)
  - Partitioning
    - Transitive Closure
  - CENTER [HGI-WebDB'00]
  - MERGE-CENTER [HM-VLDBJ09]



Star Algorithm [APR-JGraph04]

- Creates star-shaped clusters
  - As a heuristic to approximate the problem of finding minimal clique cover of the graph
- Similar to CENTER but
  - Allows *overlapping* clusters
  - First sorts the nodes in descending order of their degrees

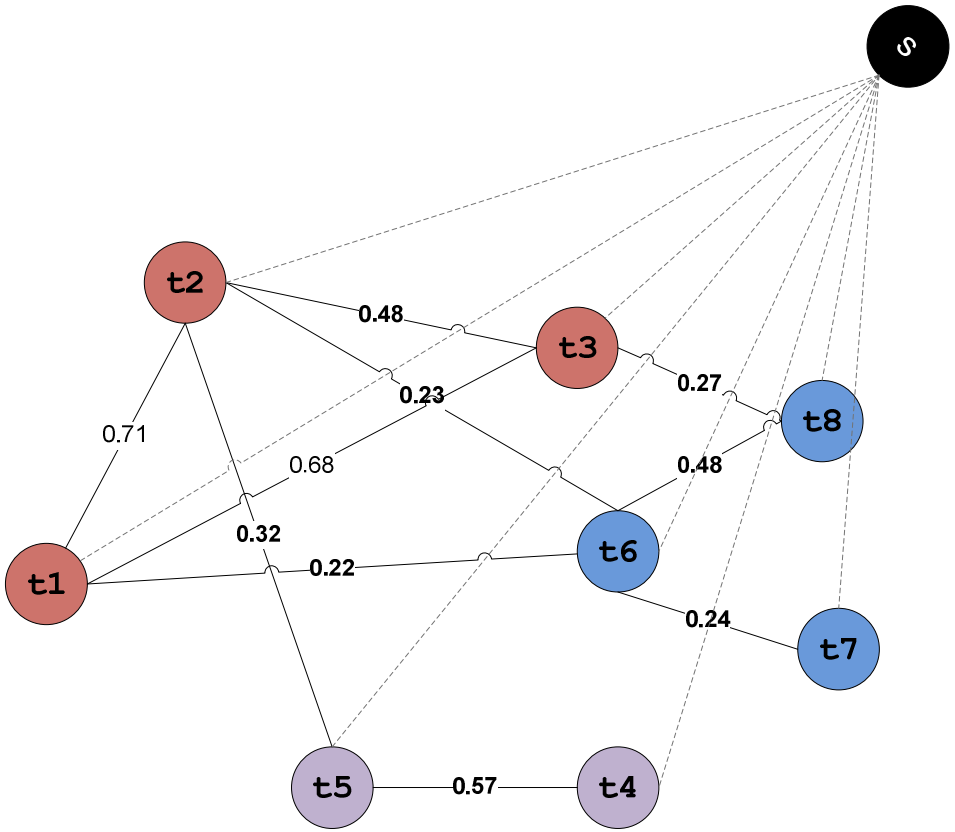


Ricochet Algorithms [WB-DASFAA'09]

- Ricochet family of algorithms
  - Four algorithms originally proposed for document clustering
    - SR, BSR, CR and OCR
  - Based on a strategy that resembles the rippling of stones thrown in a pond
  - Combine ideas from the classic K-means algorithm and the Star algorithm
    - Selects seeds (star centers) for the clusters and then iteratively refines each cluster
  - SR and BSR perform a sequential selection of the cluster seeds, CR and OCR perform a concurrent selection of the seeds

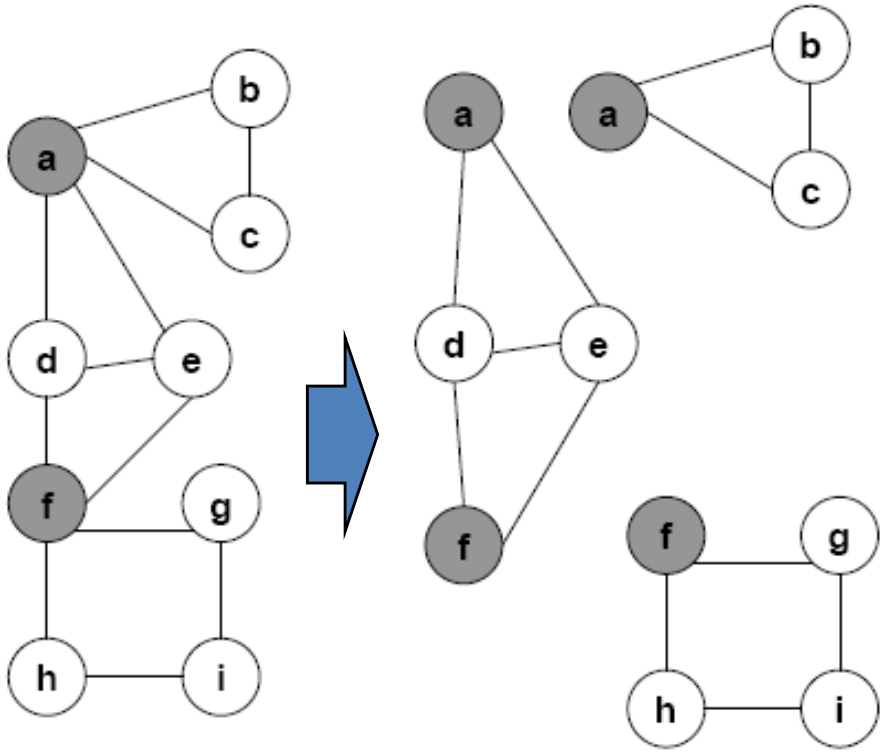
Min-Cut Clustering

- Based on the Cut-Clustering Algorithm [FTT-IM04]
  - Finding minimum cuts of edges in the similarity graph after inserting an artificial sink into similarity graph G



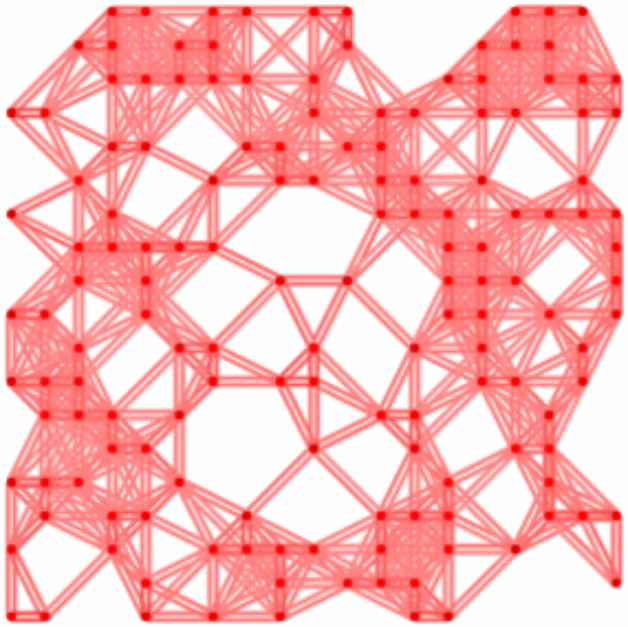
Articulation Point Clustering

- A scalable graph partitioning algorithm
- Based on finding articulation points
  - Articulation point: a vertex whose removal makes the graph disconnected
- Efficient implementations proposed for identifying chatter in the blogosphere [BCKT-VLDB07]



Markov Clustering (MCL) [Dongen-Thesis00]

- Based on simulation of stochastic flow in graphs
  - The graph is mapped onto a Markov matrix
  - Transition probabilities recomputed through the alternate application of two simple algebraic operations on matrices
    - *Expansion* and *Inflation*
- Clusterings with different scales of granularity by varying the inflation parameter of the algorithm
- Highly optimized implementation that makes the algorithm highly scalable



Correlation Clustering [BBC-ML04]

- Original problem: a graph clustering given edges labelled with '+' or '–'
  - '+' indicates correlation between the nodes
  - '–' indicates uncorrelated nodes.
- The goal is to find a clustering that agrees as much as possible with the edge labels.
  - NP-Hard, approximations needed.
- The labels can be assigned to edges based on the similarity scores of the records (edge weights) and a threshold value.
- Several approximations exist
  - We use algorithm Cautious from [BBC-ML04] in our paper

Summary of the Algorithms

- Clustering algorithms perform based on different goals and criteria
  - Some are heuristic algorithms
    - Best effort in a single pass over data
  - Some solutions are approximate
    - Optimal but hard solutions
    - Scalability may vary

This calls for a thorough experimental evaluation of the algorithms

Outline

- Stringer Duplicate Detection Framework
- Overview of the Clustering Algorithms
- **Evaluation Framework**
  - Datasets
  - Quality Measures
  - Summary of the results
- Conclusion and Future Directions

Datasets

- Synthetic Datasets
  - Using enhanced UIS Data Generator [HS-DMKD'98]
    - Gets a clean dataset as input
    - Creates clusters of erroneous records from each clean record by injecting several types of errors
    - Provides several parameters to adjust the characteristics of the data
  - Clean data sources
    - DBLP titles
    - Company Names
- Real Datasets
  - Widely-used Cora dataset

Datasets

Classification of the datasets used in the experiments

Group	Name	Percentage of			
		Erroneous Duplicates in the Dataset	Error in each Duplicate Record	Token Swap	Abbr. Error
High Error	H1	90	30	20	50
	H2	50	30	20	50
Medium Error	M1	30	30	20	50
	M2	10	30	20	50
	M3	90	10	20	50
	M4	50	10	20	50
Low Error	L1	30	10	20	50
	L2	10	10	20	50
Single Error	Abbr.	50	0	0	50
	TokenSwap	50	0	20	0
	LowEdit	50	10	0	0
	MediumEdit	50	20	0	0
	HighEdit	50	30	0	0

Clean source: Company names dataset
Distribution of Errors: Uniform

UofT:DB Group

Datasets

Classification of the datasets used in the experiments

Group	Name	Percentage of			
		Erroneous Duplicates in the Dataset	Error in each Duplicate Record	Token Swap	Abbr. Error
Zipfian High	ZH1	90	30	20	50
	ZH2	50	30	20	50
Zipfian Medium Error	ZM1	30	30	20	50
	ZM2	10	30	20	50
	ZM3	90	10	20	50
	ZM4	50	10	20	50
Zipfian Low	ZL1	30	10	20	50
	ZL2	10	10	20	50

Clean source: Company names dataset
Distribution of Errors: Zipfian

Datasets

Classification of the datasets used in the experiments

Group	Name	Percentage of			
		Erroneous Duplicates in the Dataset	Error in each Duplicate Record	Token Swap	Abbr. Error
DBLP High	DH1	90	30	20	50
	DH2	50	30	20	50
DBLP Medium Error	DM1	30	30	20	50
	DM2	10	30	20	50
	DM3	90	10	20	50
	DM4	50	10	20	50
DBLP Low	DL1	30	10	20	50
	DL2	10	10	20	50

Clean source: DBLP dataset
Distribution of Errors: Uniform

Evaluation of Accuracy

- Traditional accuracy measures from IR
  - Assuming *k* ground truth clusters $G=\{g_1, \ldots, g_k\}$ and a mapping *f* that maps the elements of G to the elements of the clustering $C=\{c_1, \ldots, c_k\}$
    - Precision and Recall of cluster c_i

$$Pr_i = \frac{|f(g_i) \cap g_i|}{|f(g_i)|} \qquad Re_i = \frac{|f(g_i) \cap g_i|}{|g_i|}$$

  - Precision and Recall

$$Pr = \sum_{i=1}^{k} \frac{|g_i|}{|R|} Pr_i \qquad Re = \sum_{i=1}^{k} \frac{|g_i|}{|R|} Re_i$$

  - F1-measure (F_1)
    - harmonic mean of precision and recall, i.e.:

$$F_1 = \frac{2 \times Pr \times Re}{Pr + Re}$$

Evaluation of Accuracy

- Accuracy measures suitable for duplicate detection
  - Clustering Precision (CPr)
    - The ability of the clustering algorithm to assign records that should be in the same cluster to a single cluster, regardless of the number and the size of the clusters produced.

$$CPr_i = \frac{|(t,s) \in c_i \times c_i | t \neq s \wedge \exists j \in 1 \ldots k, (t,s) \in g_j \times g_j|}{\binom{k'}{2}}$$

    - CPr is the average CPr_i value over all clusters
  - Penalized Clustering Precision (PCPr)
    - Penalizes those algorithms that create greater or fewer clusters than the ground truth

$$PCPr = \begin{cases} \frac{k}{k'} CPr & k < k' \\ \frac{k'}{k} CPr & k \geq k' \end{cases}$$

Best Accuracy Results

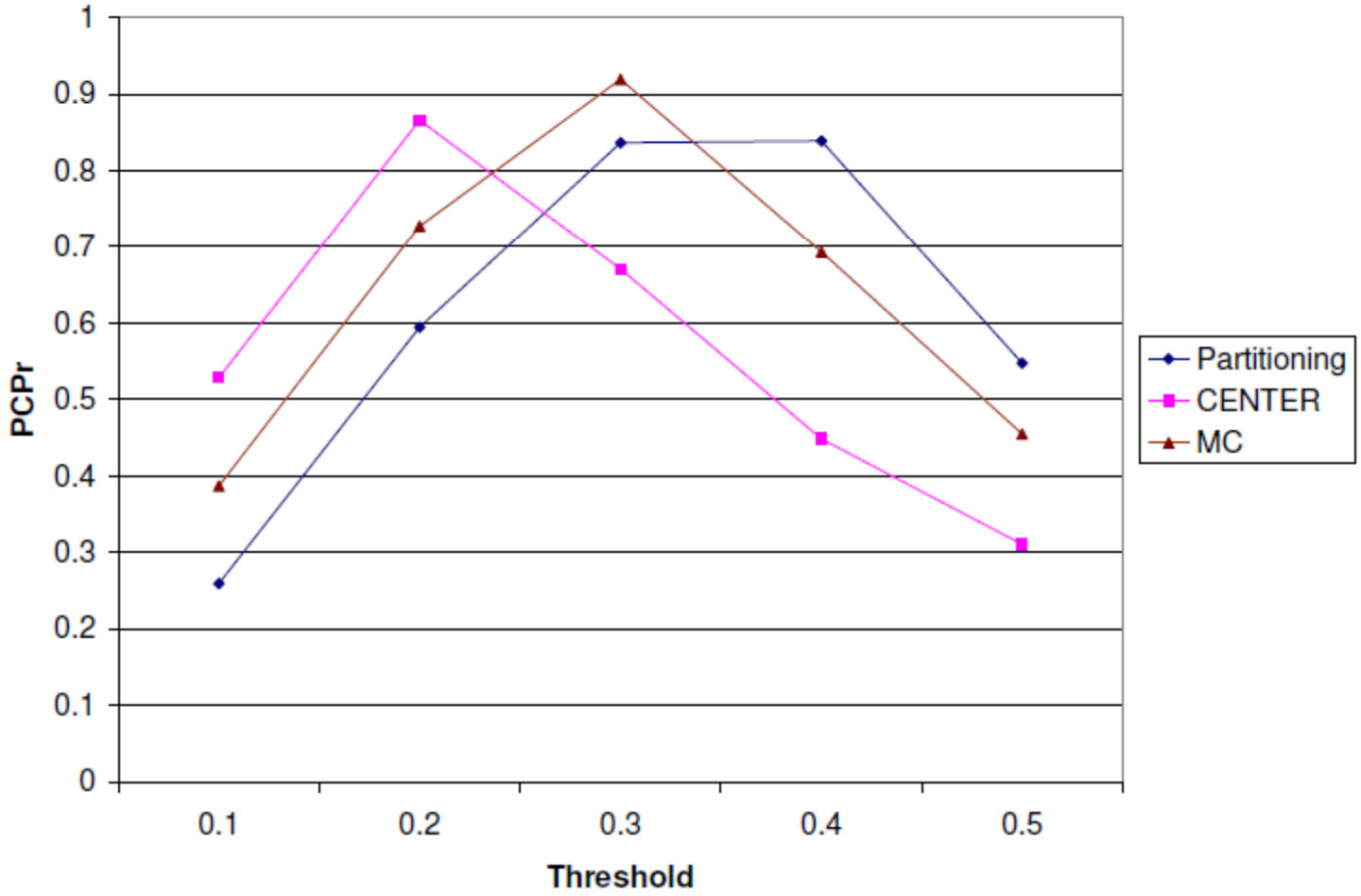
- Best accuracy gained using different threshold values

	Partitioning		Center		MERGE-CENTER	
	Best PCPr	Best F1	Best PCPr	Best F1	Best PCPr	Best F1
PCPr	0.554	0.469	0.638	0.298	0.695	0.437
F1	0.622	0.910	0.825	0.877	0.776	0.918
Cluster #	354	994	697	1305	459	1030

Results on medium error datasets
500 ground truth clusters

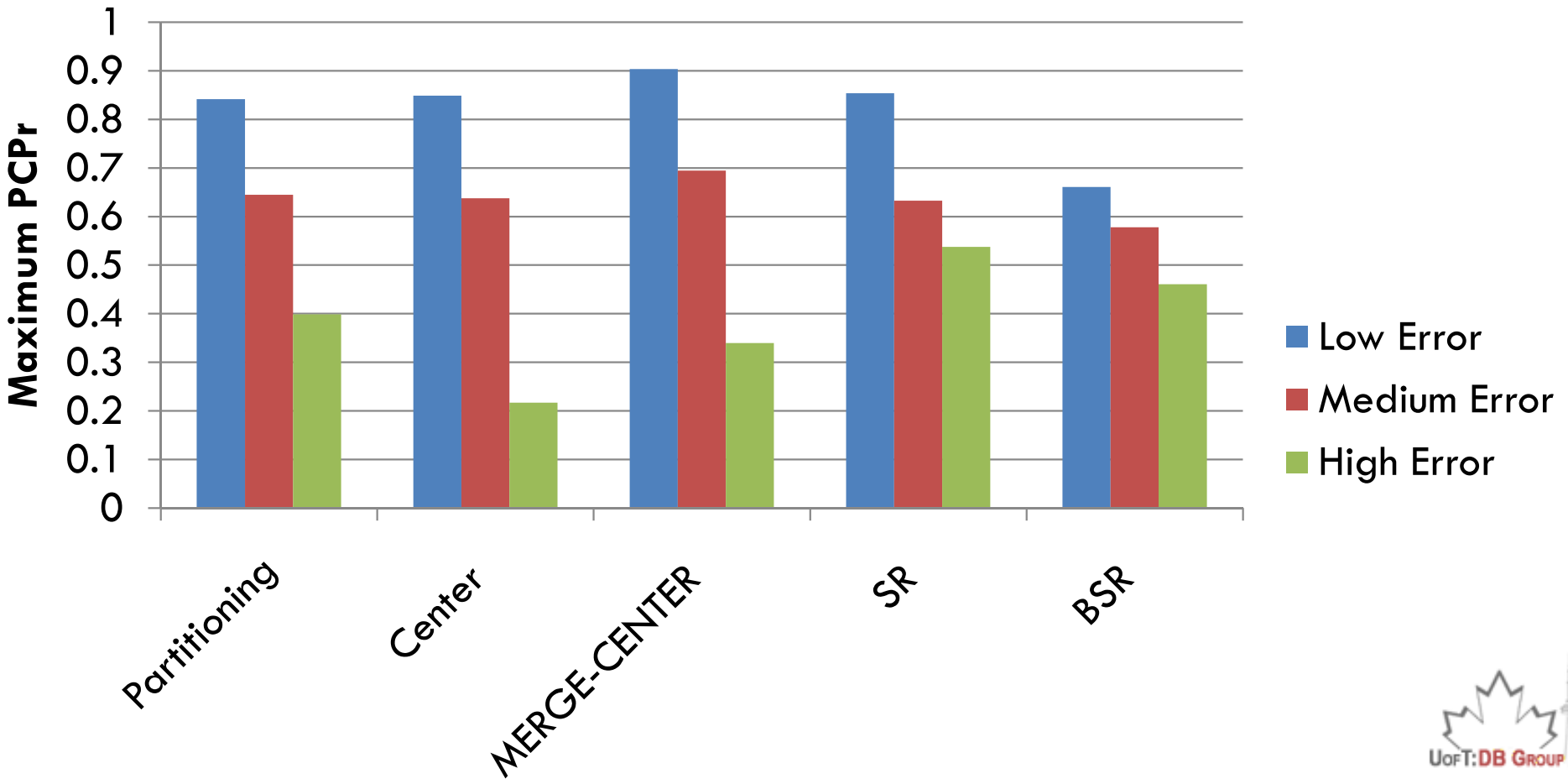
Effect of Threshold

- Accuracy Results on Cora Dataset



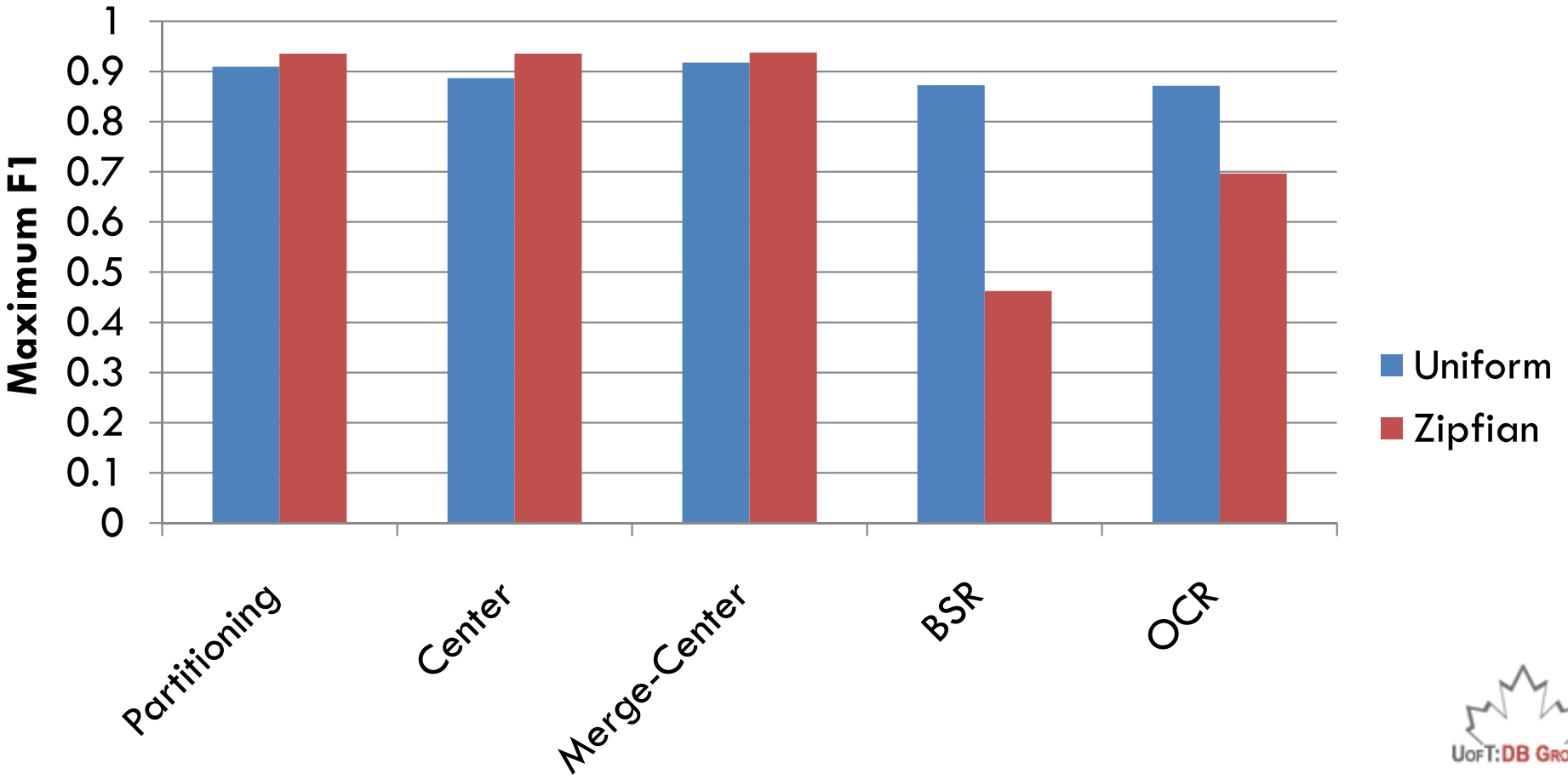
Effect of Amount of Errors

- The effect of amount of errors on the accuracy of the algorithms



Effect of Distribution of Errors

- The effect of amount of the distribution of errors on the accuracy of the algorithms



Running Time

Algorithm	Time
Partitioning	1.790 sec
CENTER	3.270 sec
MERGE-CENTER	3.581 sec
Star	5.9 min
CCL	83.5 min
MCL	8.395 sec
MinCut	52.1 min
ArtPt.	17.563 sec

Results on medium error DBLP dataset of 100K records

Our implementation of Ricochet algorithms require storing the full graph in memory and therefore are not scalable to large datasets.

Outline

- Stringer Duplicate Detection Framework
- Overview of the Clustering Algorithms
- Evaluation Framework
  - Quality Measures
  - Datasets
  - Summary of the results
- **Conclusion and Future Directions**

Summary of the Results

	Scalability (Current Implementation)	Ability to find the correct number of clusters	Robustness Against		
			Choice of threshold	Amount of Errors	Distribution of Errors
Partitioning	High	Low	Low	Low	High
CENTER	High	High	Low	Low	High
MERGE CENTER	**High**	**High**	**Medium**	**Low**	**High**
Star	Medium	High	Low	Low	High
SR	Low	Medium	High	High	Low
BSR	Low	Low	High	High	Low
CR	Low	High	Medium	High	High
OCR	Low	High	Medium	High	Low
Correlation Clustering	Low	High	Low	Low	High
Markov Clustering	**High**	**High**	**Medium**	**Medium**	**High**
Min-Cut Clustering	Low	Low	Low	Low	High
Articulation Point	High	Medium	Low	Low	High

Conclusion & Future Directions

- CR and OCR are highly accurate
  - Are there scalable implementations or approximations of these algorithms?
- Duplicate detection with uncertainty [HM-VLDBJ09]
  - A "perfect" clustering may not be possible
  - As an alternative, can we:
    - Create a probabilistic database out of duplicated data
    - Each duplicate gets a probability of being "correct"
    - Use one of the existing models for probabilistic query processing such as [BSIB-VLDB09]

Conclusion & Future Directions

- Other applications of unconstrained clustering
  - Semantic link discovery
  - Document clustering
  - Protein structure classification

- Extending the evaluation
  - Other quality measures
    - Such as merge-distance [MWG-InfoLabTR09]
  - More experiments on real data
    - Results available at
      - http://dblab.cs.toronto.edu/project/stringer

The End

Questions ?

References

- [LWY-VLDB'07] C. Li, B. Wang, and X. Yang. *VGRAM: Improving Performance of Approximate Queries on String Collections Using Variable-Length Grams*. In Proc. of the Int'l Conf. on Very Large Data Bases (VLDB), pages 303–314, Vienna, Austria, 2007.
- [AGK-VLDB'06] A. Arasu, V. Ganti, and R. Kaushik. *Efficient Exact Set-Similarity Joins*. In Proc. of the Int'l Conf. on Very Large Data Bases (VLDB), pages 918–929, 2006.
- [BMS-WWW07] R. J. Bayardo, Y. Ma, and R. Srikant. *Scaling Up All Pairs Similarity Search*. In Int'l World Wide Web Conference (WWW), pages 131–140, Banff, Canada, 2007.
- [CPR+-PODS07] F. Chierichetti, A. Panconesi, P. Raghavan, M. Sozio, A. Tiberi, and E. Upfal. *Finding Near Neighbors Through Cluster Pruning*. In Proc. of the ACM Symp. on Principles of Database Systems (PODS), pages 103–112, Beijing, China, 2007.
- [SK-SIGMOD'04] S. Sarawagi and A. Kirpal. *Efficient Set Joins On Similarity Predicates*. In ACM SIGMOD Int'l Conf. on the Mgmt. of Data, pages 743–754, Paris, France, 2004.

References

- [Jain&Dubes88] A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [HGI-WebDB'00] T. H. Haveliwala, A. Gionis, and P. Indyk. *Scalable Techniques for Clustering the Web*. In Proc. of the Int'l Workshop on the Web and Databases (WebDB), pages 129–134, Dallas, Texas, USA, 2000.
- [WB-DASFAA'09] D. T. Wijaya and S. Bressan. *Ricochet: A Family of Unconstrained Algorithms for Graph Clustering*. In Proc. of the Int'l Conf. on Database Systems for Advanced Applications (DASFAA), pages 153–167, Brisbane, Australia, 2009.
- [HM-VLDBJ09] O. Hassanzadeh and R. J. Miller. *Creating Probabilistic Databases from Duplicated Data*, To Appear in the VLDB Journal
- [HS-DMKD'98] M. A. Hernandez and S. J. Stolfo. *Real-world data is dirty: Data cleansing and the merge/purge problem*. Data Mining and Knowledge Discovery, 2(1):9–37,1998.

References

- [EIV-TKDE'07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis and Vassilios S. Verykios. *Duplicate Record Detection: A Survey*. In IEEE Transactions on Knowledge and Data Engineering. 2007
- [CHK+-SIGMOD'07] A. Chandel, O. Hassanzadeh, N.Koudas, M. Sadoghi, and D. Srivastava. *Benchmarking declarative approximate selection predicates*. In SIGMOD 2007.
- [HSM-QDB'07] Oktie Hassanzadeh, Mohammad Sadoghi, Renée J. Miller. *Accuracy of Approximate String Joins Using Grams*. In QDB'07 at VLDB 2007
- [MWG-InfoLabTR09] D. Menestrina, S. E. Whang and H. Garcia-Molina. *Evaluating Entity Resolution Results (Extended version)* Technical Report. Stanford InfoLab (June 2009)
- [BSIB-VLDB09] George Beskales, Mohamed A. Soliman, Ihab F. Ilyas and Shai Ben-David, *Modeling and Querying Possible Repairs in Duplicate Detection*. In VLDB 2009.

References

- [APR-JGraph04] J. A. Aslam, E. Pelekhov, and D. Rus. *The star clustering algorithm for static and dynamic information organization*. Journal of Graph Algorithms Appl., 8:95–129, 2004.
- [FTT-IM04] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. *Graph clustering and minimum cut trees*. Internet Mathematics, 1:385–408, 2004.
- [BCKT-VLDB07] N. Bansal, F. Chiang, N. Koudas, and F. W. Tompa. *Seeking stable clusters in the blogosphere*. In VLDB'07.
- [Dongen-Thesis00] S. van Dongen. *Graph clustering by flow simulation*. PhD thesis, University of Utrecht, 2000.
- [BBC-ML04] N. Bansal, A. Blum, and S. Chawla. *Correlation clustering*. Machine Learning, 56(1-3):89–113, 2004.