Efficient Graph Similarity Joins with Edit Distance Constraints

Never Stand Still | Faculty of Engineering | School of Computer Science and Engineering

Xiang Zhao[†§] **Chuan Xiao**[†] **Xuemin Lin**[‡†] **Wei Wang**[†]

[†] The University of New South Wales, Australia

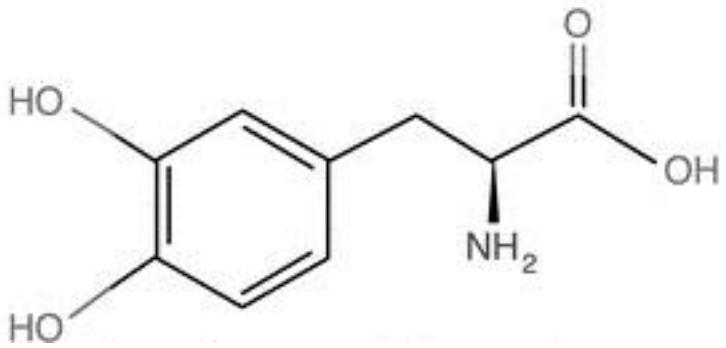
[‡] East China Normal University, China

[§] NICTA, Australia

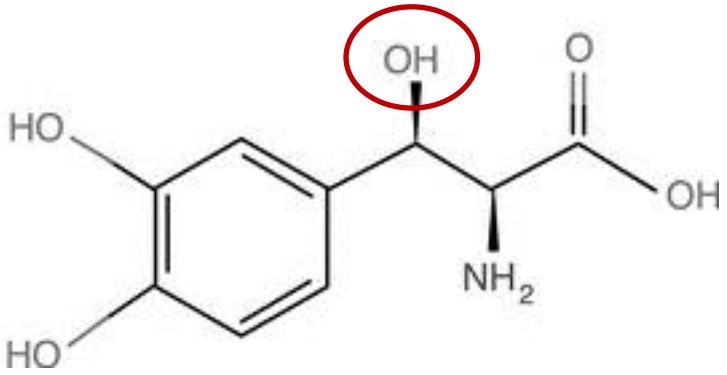
Outline

- Motivation
- Previous Approaches
- Our Approach
- Experiments
- Conclusion

Motivation



Levodopa



Droxidopa

Besides Cheminformatics,

- Bioinformatics: Similar DNA interactions
- Model repository: Search for relevant models
- Fingerprint archive: Identify suspicious persons
- …

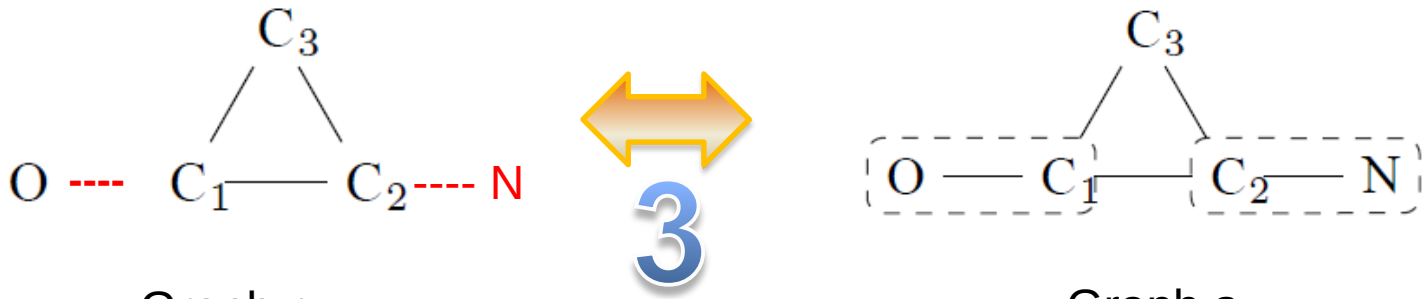
Graph Similarity Joins

Preliminaries

- A graph edit operation is an edit operation to transform one graph to another, including 6 types:
  - Insert an isolated vertex into the graph
  - Delete an isolated vertex from the graph
  - Change the label of a vertex
  - Insert an edge between two disconnected vertices
  - Delete an edge from the graph
  - Change the label of an edge

Preliminaries (con.)

- The graph edit distance between *r* and *s*, denoted by GED(*r*, *s*), is the minimum number of edit operations that transform *r* to a graph *isomorphic* to *s*.



Graph *r* Graph *s*

Computing the graph edit distance between two graphs is **NP-hard**
[Zeng et. al, PVLDB 2009]

Problem Statement

- Given two sets of graphs R and S, a graph similarity join with graph edit distance (GED) threshold τ returns pairs of graphs from each set, such that their GED is no larger than τ.
  - Focus on *self-join* case: (r_i, r_j) s.t. r_i, r_j from R, $i < j$, $\text{GED}(r_i, r_j) \leq \tau$
  - *Simple* labeled graphs: $r = (V, E, l_V, l_E)$

q-gram Approach

Different from Graphs

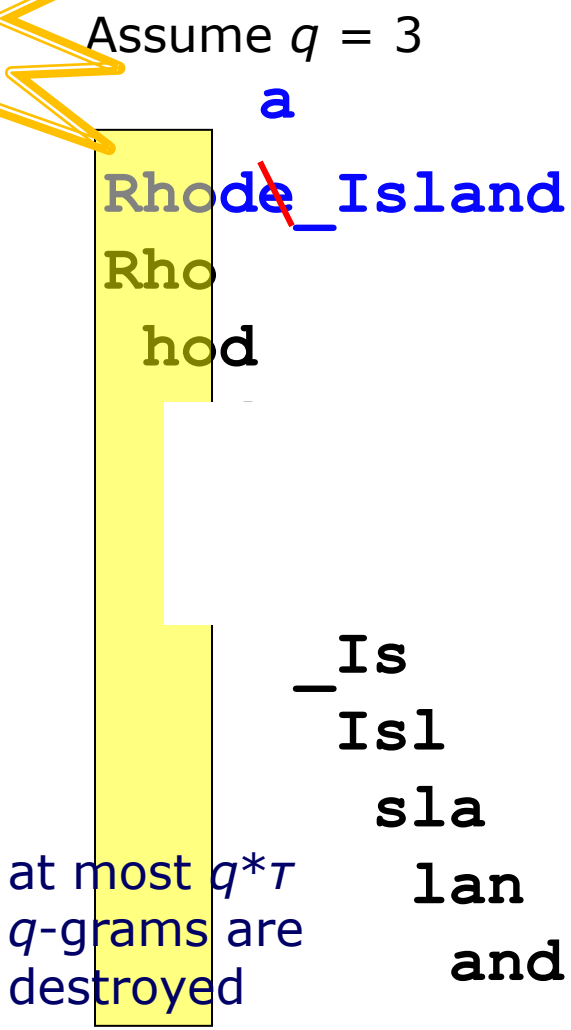
- *q*-grams on strings:

  Substrings of length *q*
- Principle: An edit operation will only affect a *limited* number of *q*-grams, similar strings will have certain amount of overlaps.

Count Filtering Condition:

Lower Bound = $l - q^* \tau$

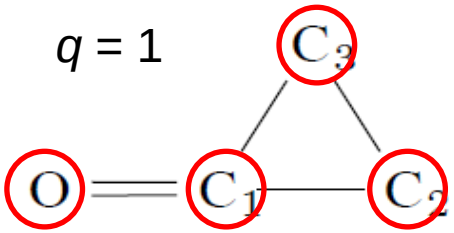
[Gravano et al., VLDB 2001]

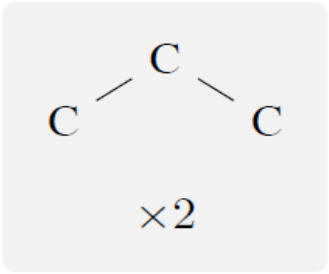


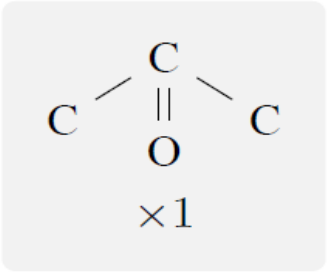
UNSW THE UNIVERSITY OF NEW SOUTH WALES

***k*-AT:** Tree-based *q*-gram [Wang et al.,TKDE 2012]

- For each vertex *u*, a tree-based *q*-gram is a set of vertices that can be reached from *u* in *q* hops, represented in a *breadth-first-search* tree rooted at *u*







- A lower bound is derived based on the *maximum* number of *q*-grams that can be affected by one edit operation

$$LB_{tree} = \max(|V(r)| - \tau \cdot D_{tree}(r), |V(s)| - \tau \cdot D_{tree}(s))$$

Dilemma of *k*-AT: Selectivity *v.s.* Tightness

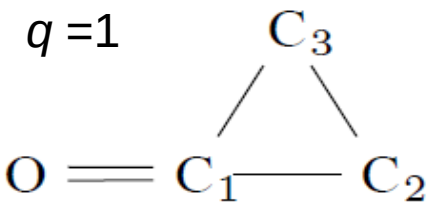
More selective

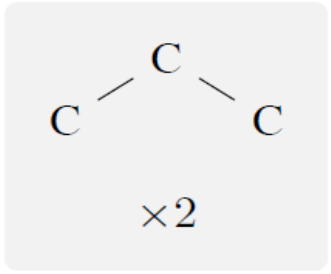
Size of *q*-gram

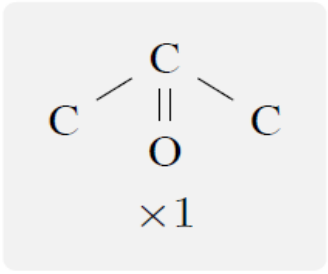
Size of D_{tree}

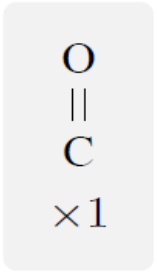
Tighter lower bound

q =1





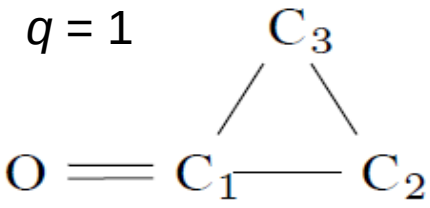


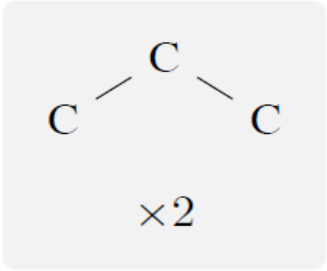


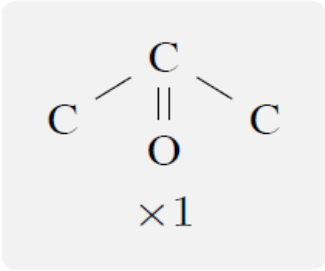
- When τ = 1, lower bound: $l - \tau * D_{tree} = 4 - 1*3 = 1$
- When τ = 2, lower bound: $l - \tau * D_{tree} = 4 - 2*3 = -2$
- Rather loose lower bound, and thus, many candidates

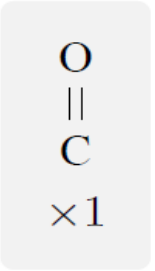
Star-structure [Zeng et al., PVLDB 2009]

- For each vertex *u*, a star structure is an attributed, single-level, rooted tree at *u*, equals to 1-gram of *k*-AT







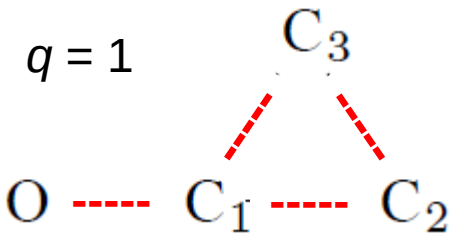


- GED lower bound and upper bound are derived via bipartite matching of star structures

- Limitation: Not take advantage of index

Path-based *q*-gram

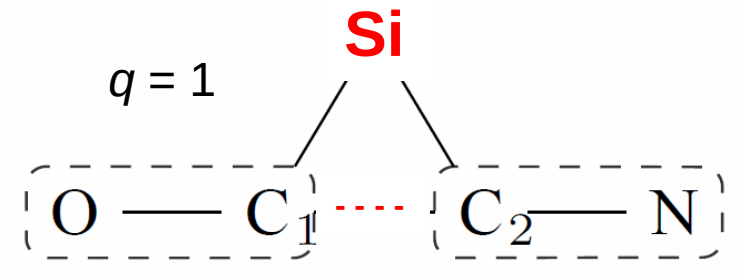
- A path-based *q*-gram in a graph is a *simple* path of length *q* (*q* hops)
  - Stored as a sequence
  - Only keep the *lexicographically smaller* one



C=O(×1) C-C(×3)

- 0-gram will be a single vertex

Edit Effects of Path-based *q*-gram



C_1 - O

C_2 - N

C_1 X C_2

C_2 X C_3

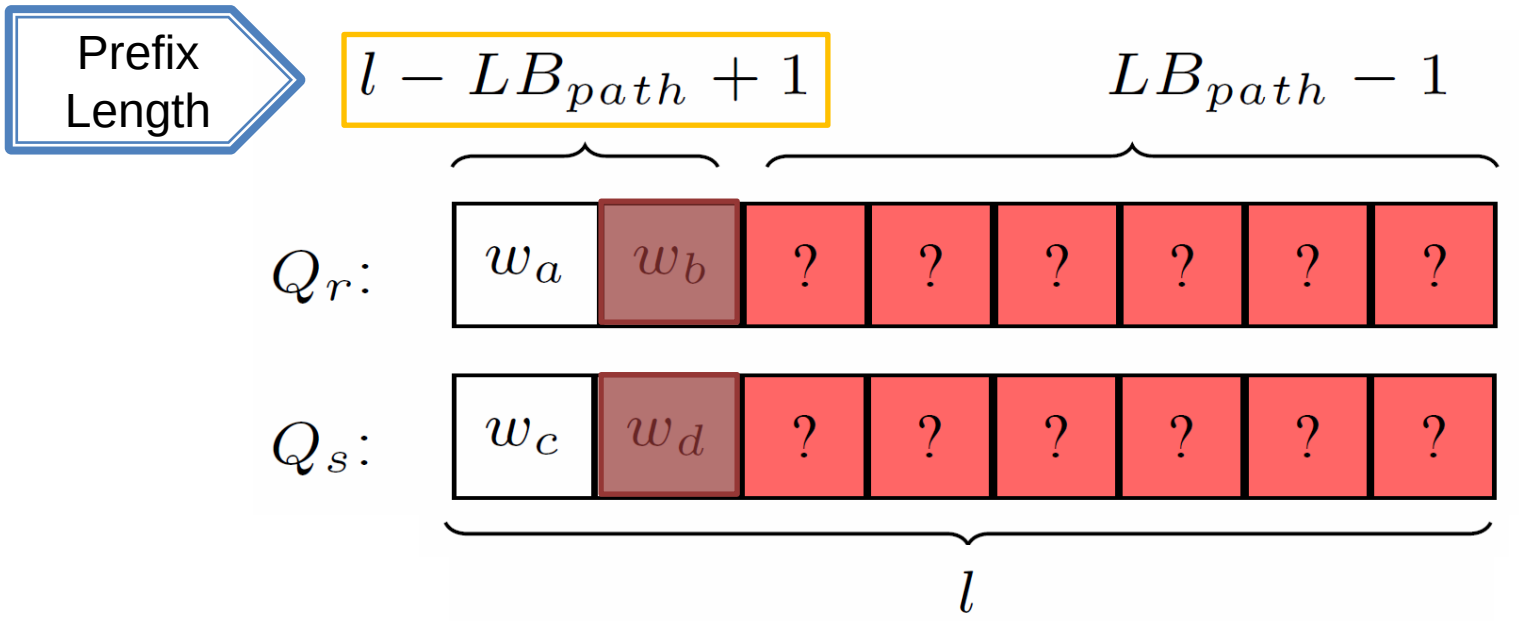
C_1 X C_3

- Insert an isolated vertex: 0
- Deleted an isolated vertex: 1 if q = 0; otherwise, 0
- Change the label of an vertex: $|Q_r^u|$
- Insert an edge: 0
- Delete an edge: 1(0) if q = 1(0); otherwise, $\max(|Q_r^u|, |Q_r^v|)$
- Change the label of en edge: $\max(|Q_r^u|, |Q_r^v|)$

$ALB_{path} = \max(|Q_r| - \tau \cdot D D_{path}(r) = \max_{u \in V(r)} |Q_r^u|_h(s))$ms

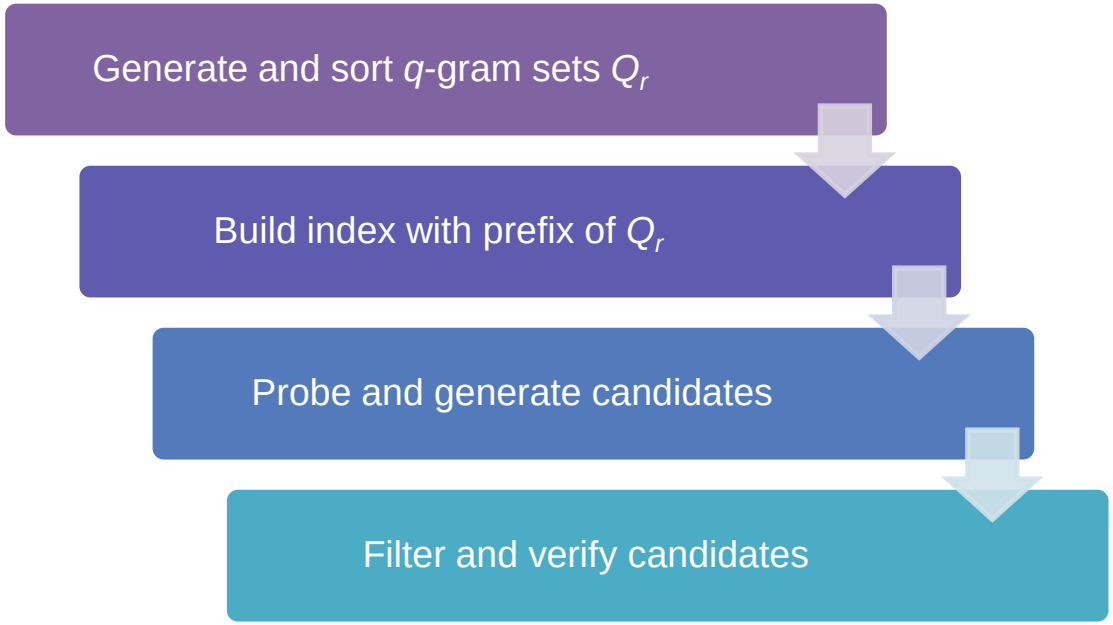
Prefix Filtering [Chaudhuri et al., ICDE 2006]

- Bottleneck of algorithms based on Count Filtering: Long *q*-gram list incurs *high* accessing cost

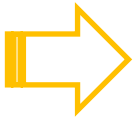


Algorithmic Framework

- **GSimJoin**: Batch join in *filtering-verification* framework

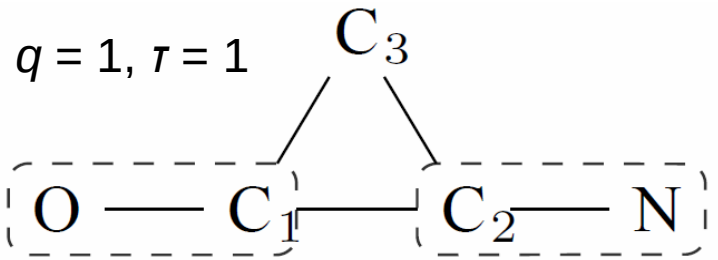


- Observation: Frequent q-grams are shared by many graphs, and result in repetitive probes of inverted index



Minimum prefix & Minimum edit filtering

Minimum Prefix



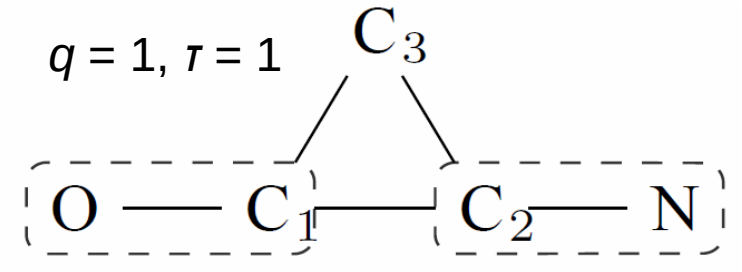
Frequency low → high

C_2 - N C_1 - O C_1 - C_2 C_2 - C_3 C_1 - C_3

1 2 3 4 5

- Previously, we have prefix length = 5 – (5 – 3*1) + 1 = 4
- Can we make it shorter?
  - Till which *q*-gram, we can safely prune the graph pair, if they have no common *q*-gram?
  - That is, find the first position till which the *q*-grams invoke *at least* (*τ*+1) errors, if all of them are mismatched.
- Yes, and let's do it with an example!

Minimum Prefix (con.)



Frequency low → high

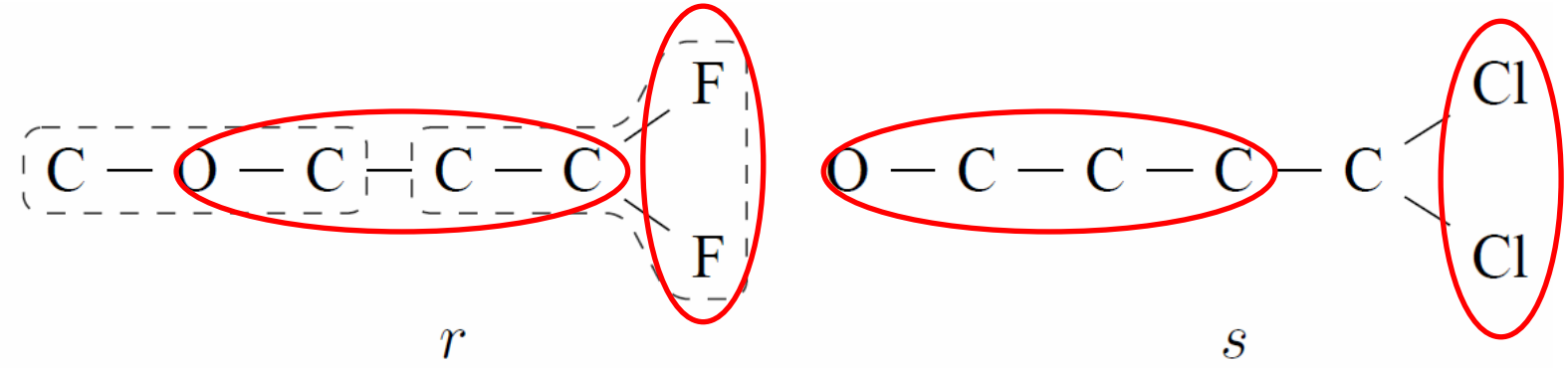
C_2 - N	C_1 - O	C_1 - C_2	C_2 - C_3	C_1 - C_3
1	2	3	4	5

- Pos 1, relabel C_2 suffices, giving 1 error = τ
- Pos 2, relabel C_1 and C_2 suffice, giving 2 errors > τ
- Done! → Minimum prefix length = 2

Minimum Graph Edit Operation problem: What is the minimum number of edit operations that suffice to destroy a given q-gram set?

Theorem: The minimum graph edit operation problem is **NP-hard**

Example for Label Filtering



- Assume $\tau = 2$, $q = 2$, each has 5 q-grams
- Global label filtering give a GED lower bound = 2 PASS
- Count filtering requires them share at least 2 q-grams PASS
- Minimum edit filtering gives GED lower bound = 2 PASS
- Compare Q_r and Q_s, 2 mismatching components:
  - *Left* gives 1 error by minimum edit filtering
  - *Right* gives 2 errors by local label filtering

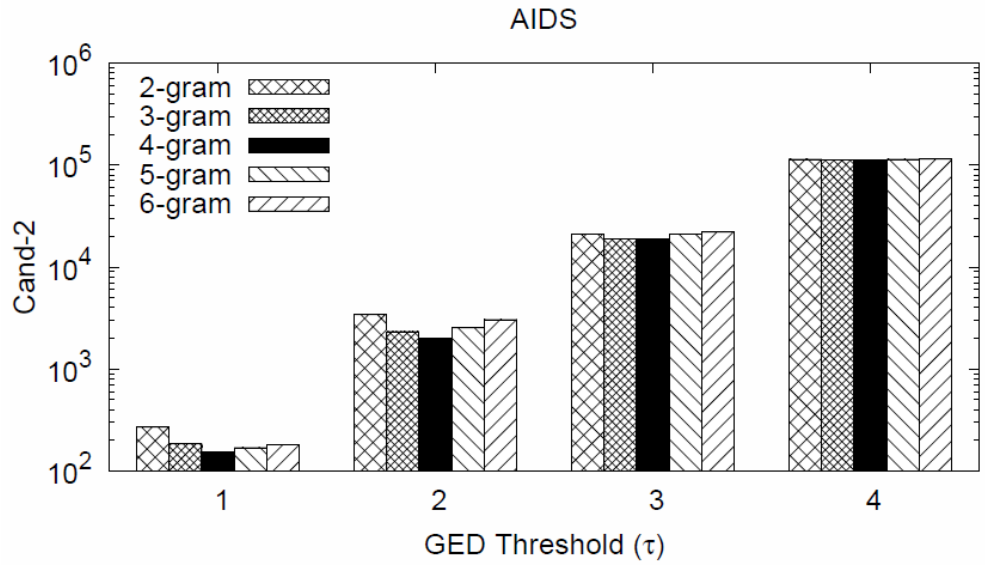
} $3 > \tau$ NO !

Experiments

- Algorithms:
  - *k*-AT: tree-based *q*-gram (*q* = 1) [Wang et al., TKDE 2012]
  - AppFull: star-structure [Zeng et al., PVLDB 2009]
  - GSimJoin: the proposed techniques
- Datasets:
  - AIDS: Antivirus screen chemical compounds from NCI/NIH
  - PROTEIN: Protein data from Protein Data Bank

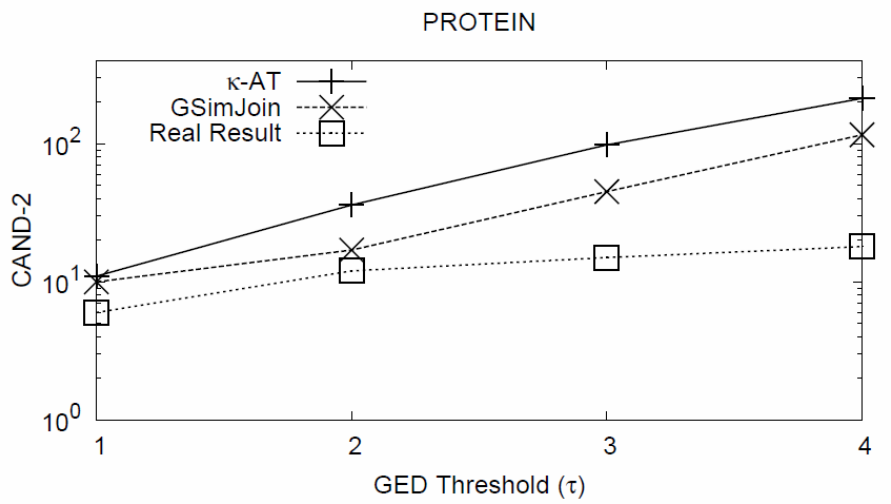
Dataset	$\|R\|$	avg $\|V\|$	avg $\|E\|$	avg $\|l_V\|$	avg $\|l_E\|$
AIDS	4,000	25.6	27.5	44	3
PROTEIN	600	32.6	62.1	3	2

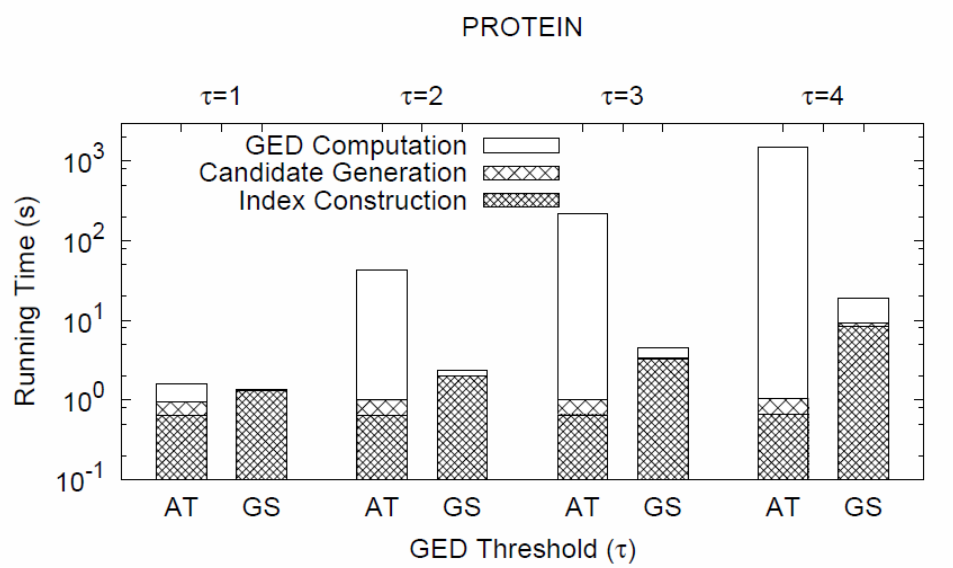
Evaluating *q*-gram Length



GSimJoin achieves the best when $q = 4$ on AIDS ($q = 3$ on PROTEIN).

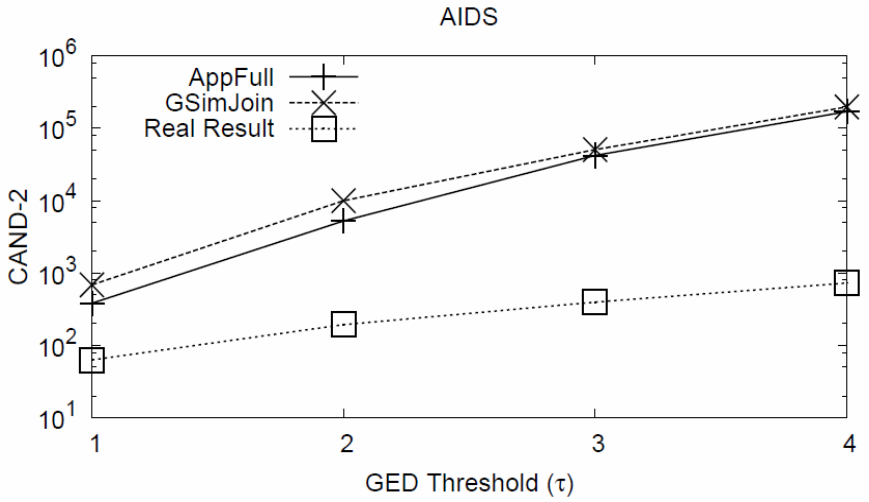
Comparing with Tree-based *q*-gram

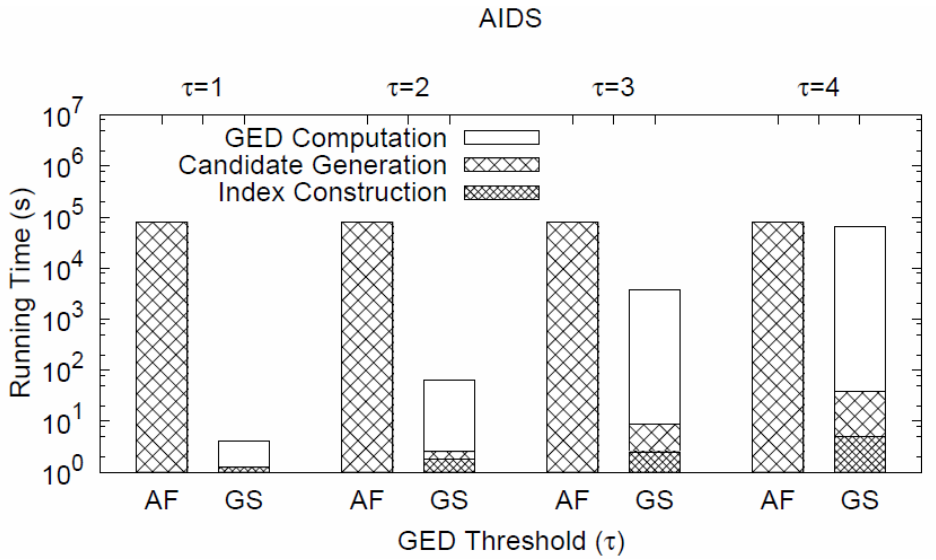




GSimJoin has smaller candidate size, and thus, less running time.

Comparing with Star-structure





GSimJoin has much better overall performance regarding running time.

Conclusion

- Contributions:
  - New notion of *q*-gram based on paths, and count filtering condition.
  - Devise minimum edit and label filtering techniques.
  - New algorithm GSimJoin, demonstrated by extensive experiments.

- Future work:
  - Similarity search, Similarity All-matching, etc.

Thank you!

Questions?

Related Work

- *q*-gram for String Similarity Joins
  - Fixed Length
    - L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava. *Approximate string joins in a database (almost) for free*. In VLDB, 2001.
    - C. Xiao, W. Wang, and X. Lin. *Ed-Join: an efficient algorithm for similarity joins with edit distance constraints*. PVLDB, 2008.
  - Variable Lengths
    - C. Li, B. Wang, and X. Yang. VGRAM: *Improving performance of approximate queries on string collections using variable-length grams*. In VLDB, 2007.
    - X. Yang, B. Wang, and C. Li. *Cost-based variable-length-gram selection for string collections to support approximate queries efficiently*. In SIGMOD Conference, 2008.

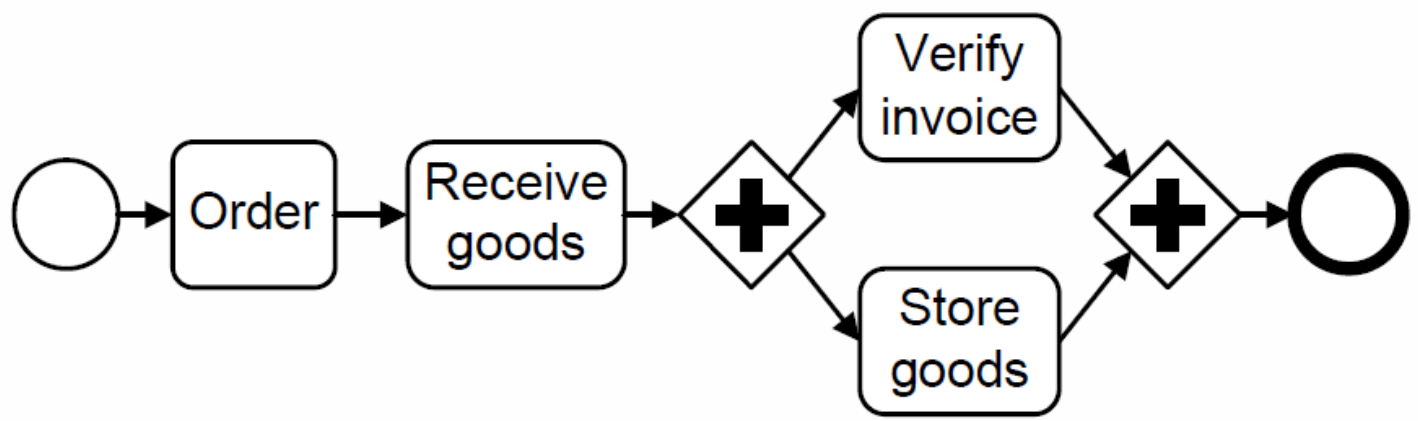
Related Work (con.)

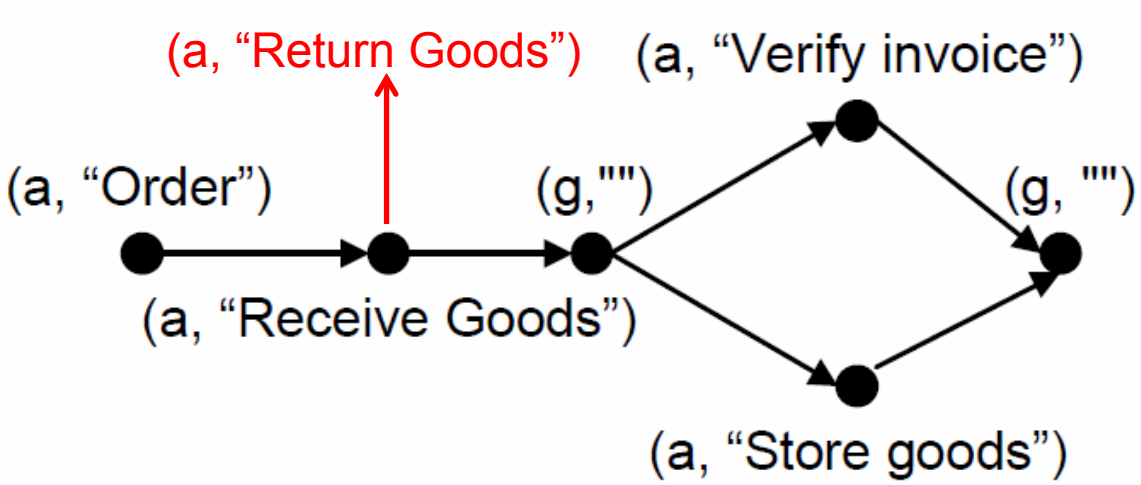
- Graph Structure Similarity Search
  - Graph Similarity Selection
    - G. Wang, B. Wang, X. Yang, and G. Yu. *Efficiently indexing large sparse graphs for similarity search*. TKDE, 2012.
    - Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. *Comparing stars: On approximating graph edit distance*. PVLDB, 2009.
  - Subgraph Similarity
    - H. He and A. K. Singh. *Closure-tree: An index structure for graph queries*. In ICDE, 2006.
    - X. Yan, P. S. Yu, and J. Han. *Substructure similarity search in graph databases*. In SIGMOD Conference, 2005.
    - H. Shang, X. Lin, Y. Zhang, J. X. Yu, and W. Wang. *Connected substructure similarity search*. In SIGMOD Conference, pages 903–914, 2010.
  - Supergraph Similarity
    - H. Shang, K. Zhu, X. Lin, Y. Zhang, and R. Ichise. *Similarity search on supergraph containment*. In ICDE, 2010.

Related Work (con.)

- Graph Edit Distance Computation
  - Exact Algorithm
    - K. Riesen, S. Fankhauser, and H. Bunke. *Speeding up graph edit distance computation with a bipartite heuristic*. In MLG, 2007.
  - Sub-optimal Algorithm
    - S. Fankhauser, K. Riesen, and H. Bunke. *Speeding up graph edit distance computation through fast bipartite matching*. In GbRPR, 2011.
    - R. Raveaux, J.-C. Burie, and J.-M. Ogier. *A graph matching method and a graph matching distance based on subgraph assignments*. Pattern Recognition Letters, 2010.

More Applications: Business Model Repository





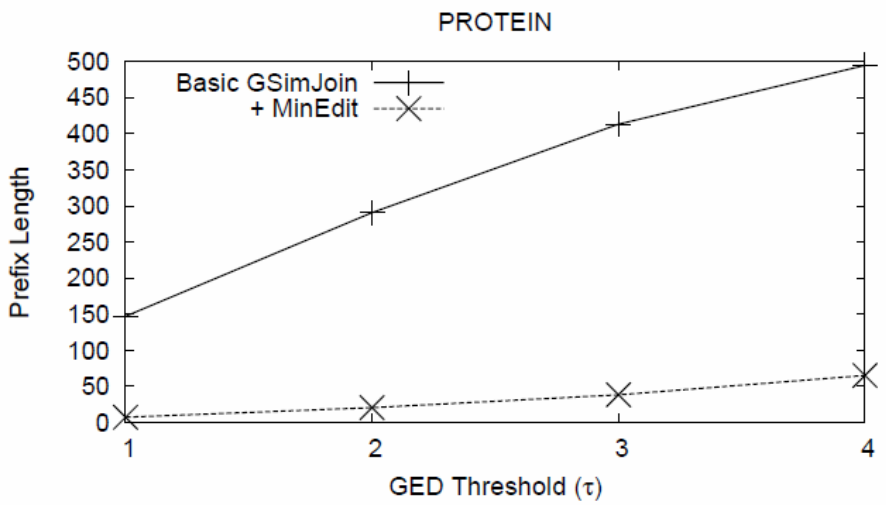
Accelerating Verification

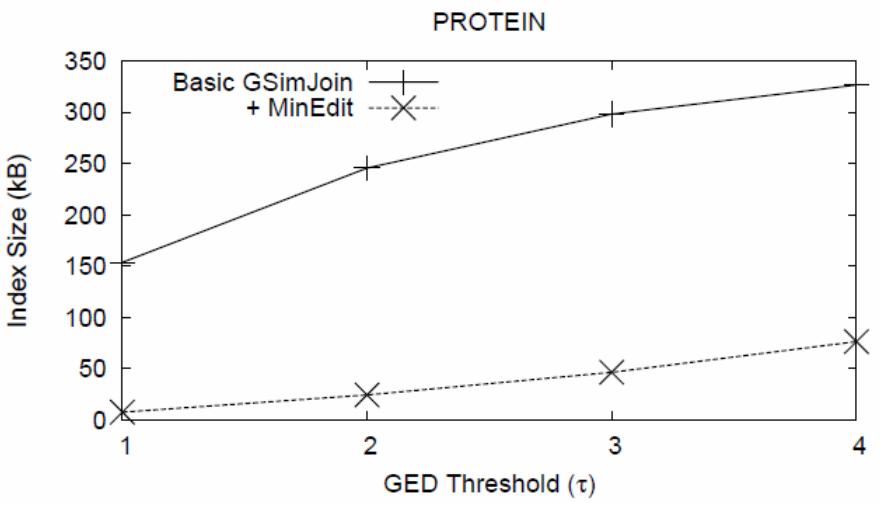
- Fastest exact algorithm: based on A* search [Riesen et. al, MLG 2007]
- *Best-first* search with GED estimation as $g(x) + h(x)$
  - $g(x) = \text{GED}(r_p, s_p)$
  - $h(x) = \Gamma(L_V(r_q), L_V(s_q)) + \Gamma(L_E(r_q), L_E(s_q))$
- Improvement:

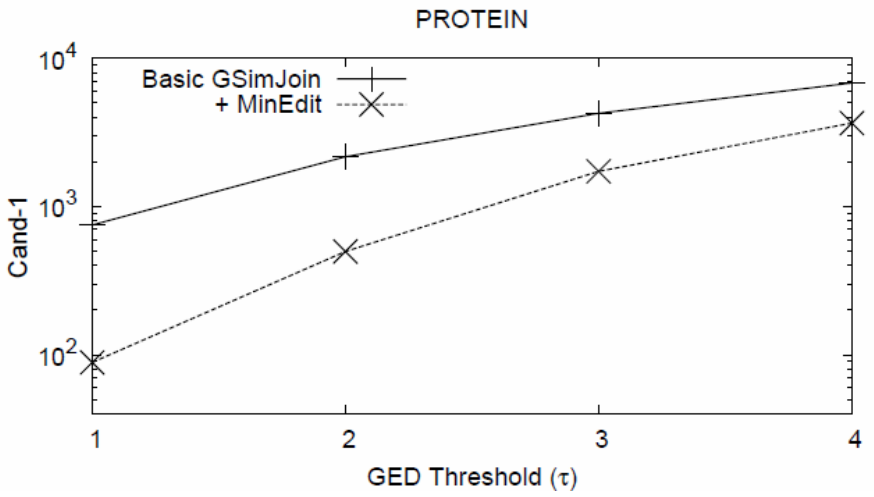
  1) Use better matching order by minimum edit filtering
     - To match the vertices of mismatching components ahead of others

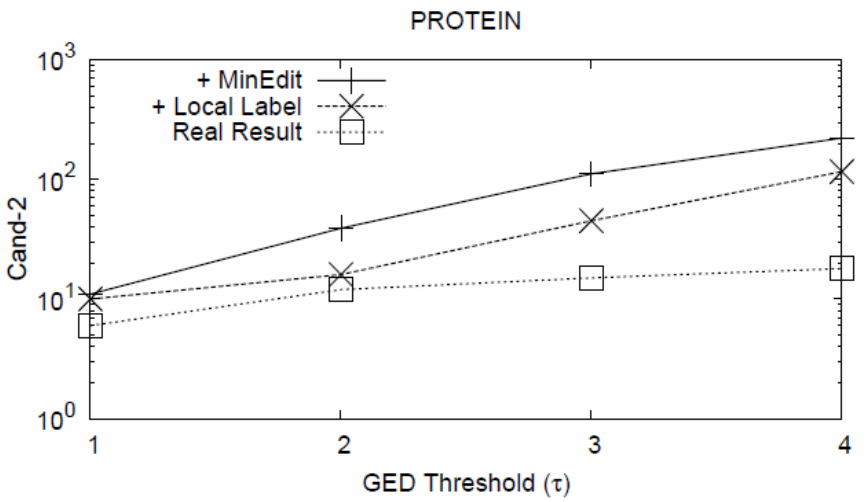
  2) Get better estimation by local label filtering
     - Compare q-gram sets Q_r' of r_q and Q_s' of s_q
     - $h(x) = \max \{ \Gamma(L_V(r_q), L_V(s_q)) + \Gamma(L_E(r_q), L_E(s_q)),$
       Err_r (from Q_r' to s_q),
       Err_s(from Q_s' to r_q) $\}$

Evaluating Filters

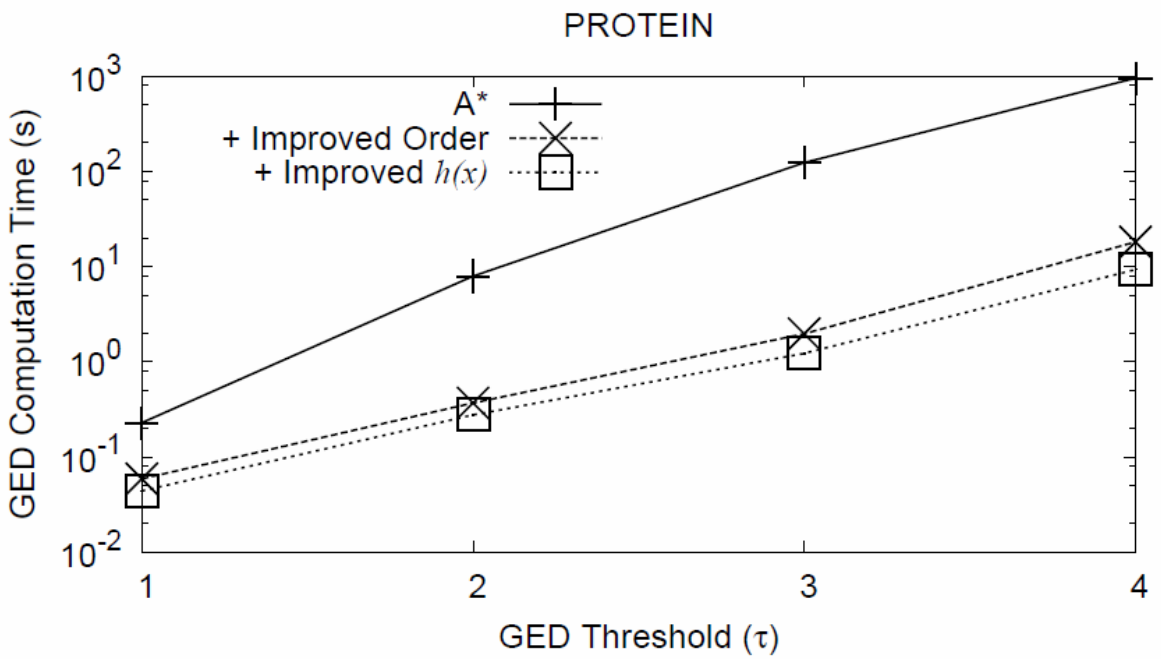








Evaluating GED Computation



Proposed improvements successfully reduce the verification time.