Similarity Queries over Hierarchical Data

PhD Defense - December 9th 2021

Thomas Hütter

database research group

PARIS LODRON UNIVERSITY SALZBURG | Department of Computer Science

Supervisor: Univ.-Prof. DipI.-Ing. Nikolaus Augsten, Ph.D.
Co-Supervisor: Dr. Mateusz Pawlik

Supported by:

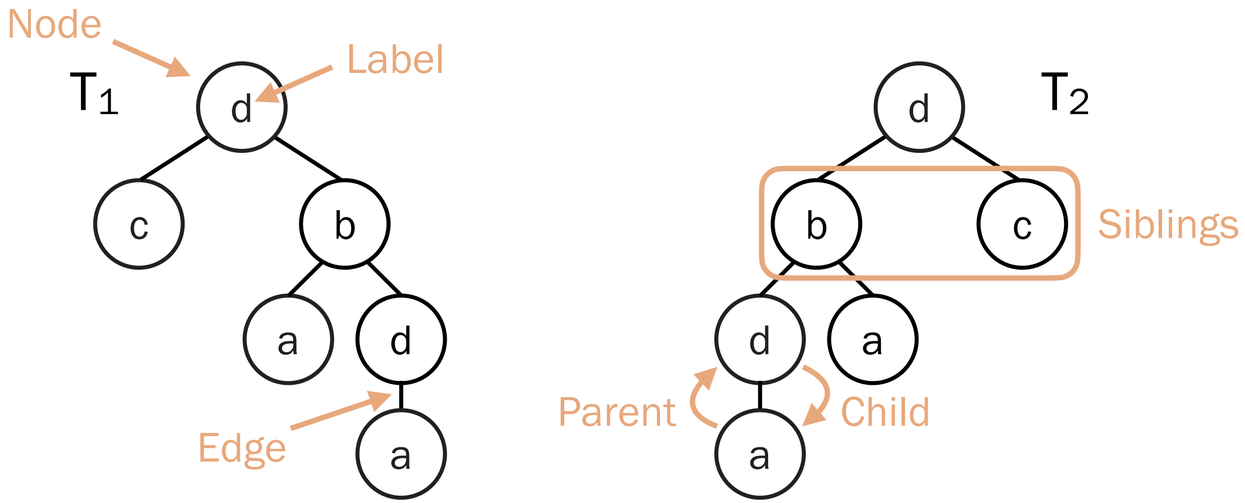
Marshallplan-Jubiläumsstiftung
Austrian Marshall Plan Foundation
Fostering Transatlantic Excellence

FWF
Der Wissenschaftsfonds.
Projektnummer P 29859, P 34962-N

Introduction

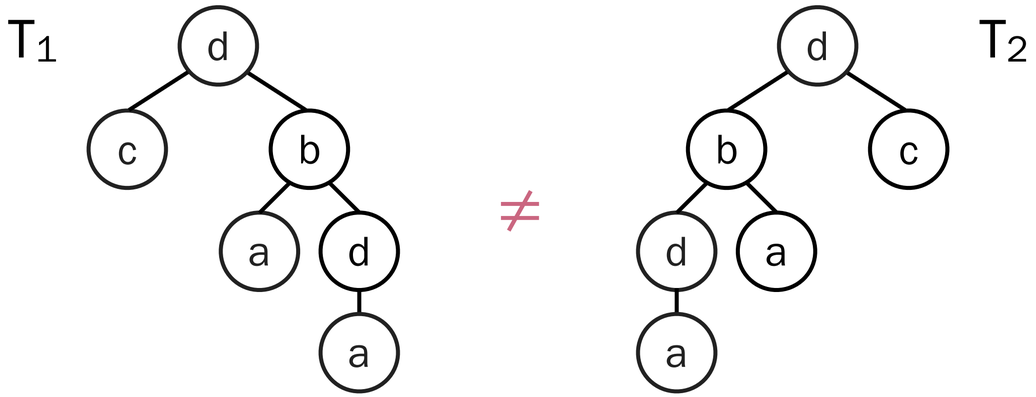
Trees, Tree Edit Distance, JSON

Trees



Trees - Sibling Order

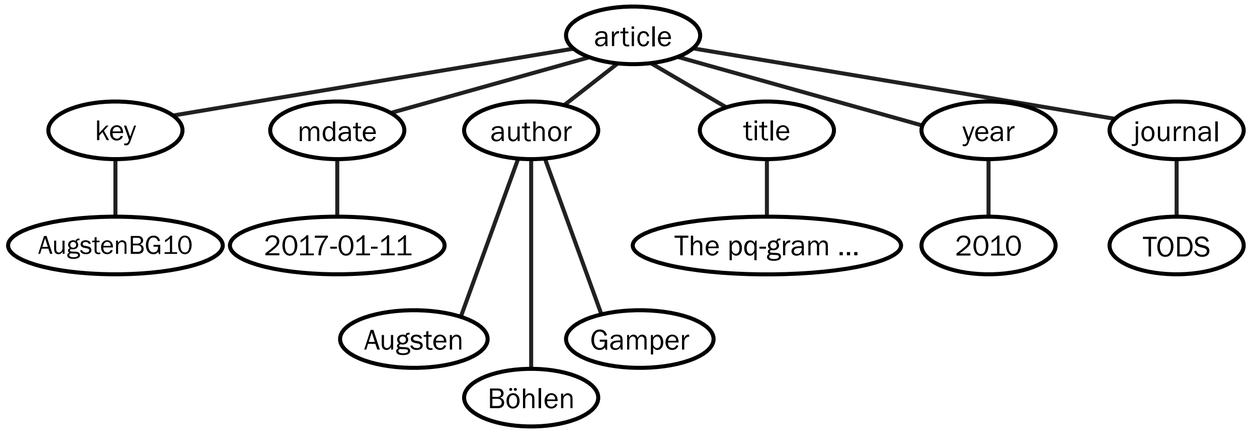
- **Ordered trees:** sibling order matters.



Trees in Computer Science

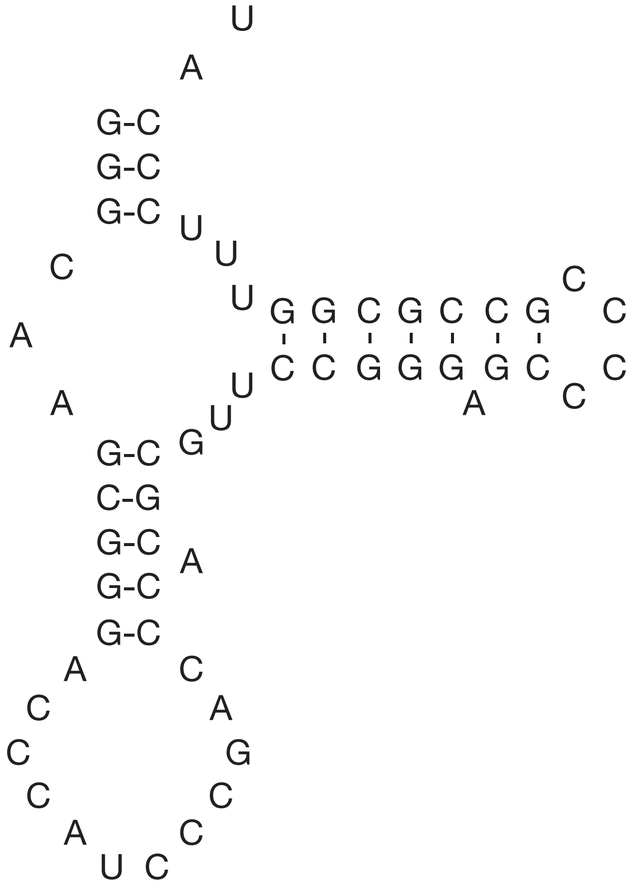
- **XML[1]:**

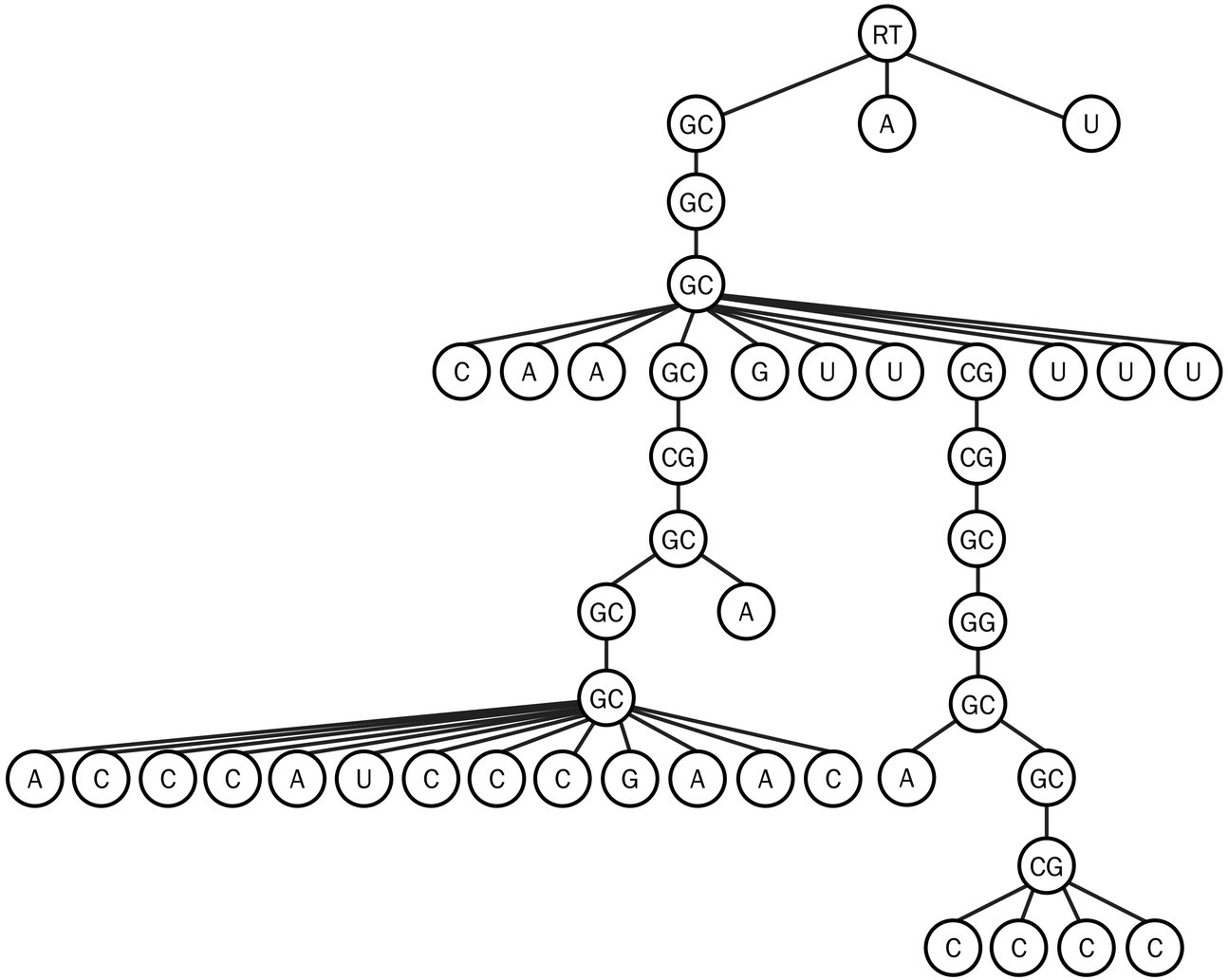
```
<article mdate="2017-01-11" key="…/AugstenBG10">
  <author>Nikolaus Augsten</author>
  <author>Michael H. Boehlen</author>
  <author>Johann Gamper</author>
  <title> The pq-gram distance
  between ordered labeled trees.</title>
  <year>2010</year>
  <volume>35</volume>
  <journal>ACM Trans. Database Syst.</journal>
  <number>1</number>
  <ee>http://doi.acm.org/
      10.1145/1670243.1670247</ee>
  <url>db/journals/…#AugstenBG10</url>
  <pages>4:1-4:36</pages>
</article>
```



Trees in Biology

- **RNA secondary structure[2]:**

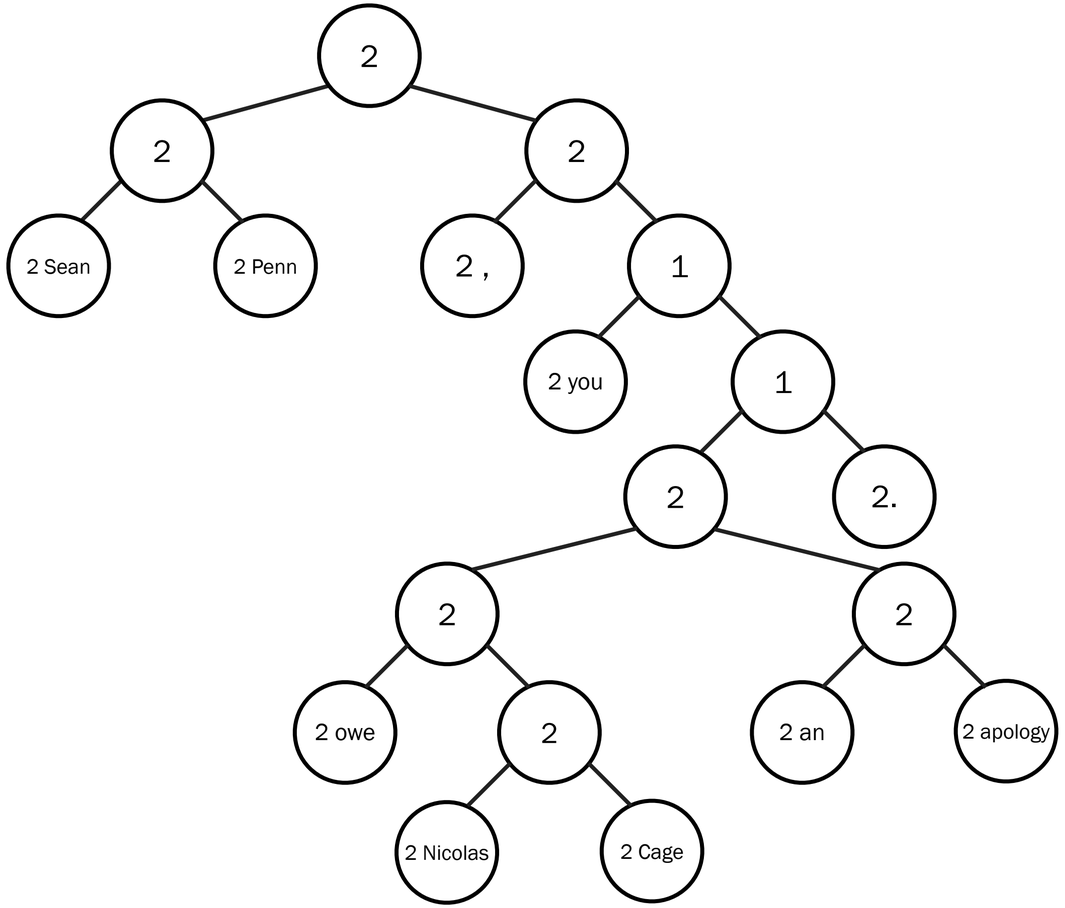




Trees in Linguistics

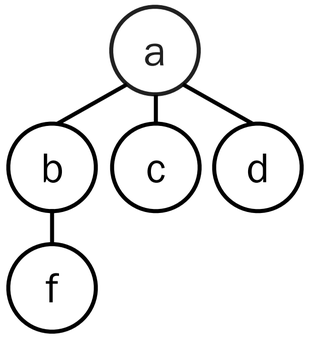
- **Sentimental analysis of movie ratings[3]:**

Sean Penn, you owe Nicolas Cage an apology.



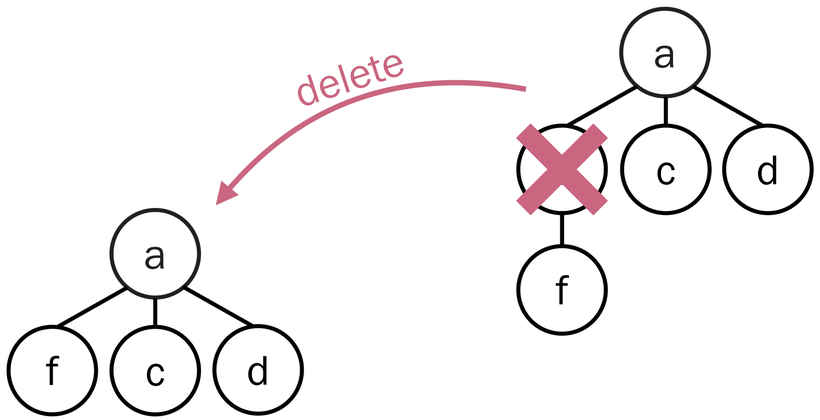
Tree Edit Distance (TED)

- **Definition:** the minimum number of edit operations that transform one tree into another.
- **Edit operations:**



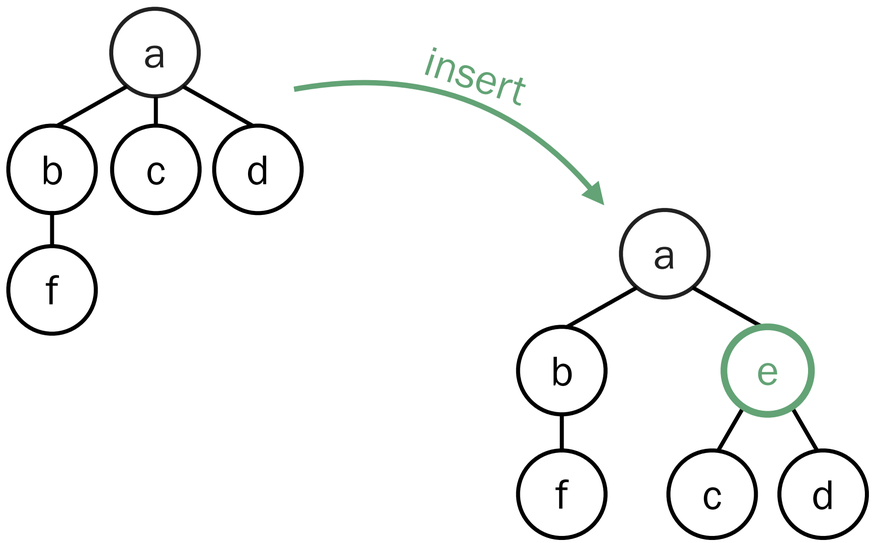
Tree Edit Distance (TED)

- **Definition:** the minimum number of edit operations that transform one tree into another.
- **Edit operations:**



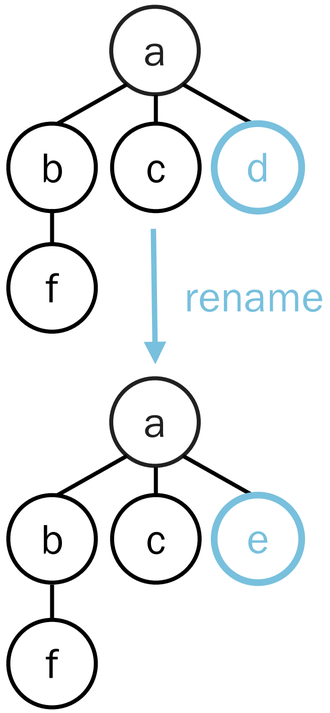
Tree Edit Distance (TED)

- **Definition:** the minimum number of edit operations that transform one tree into another.
- **Edit operations:**



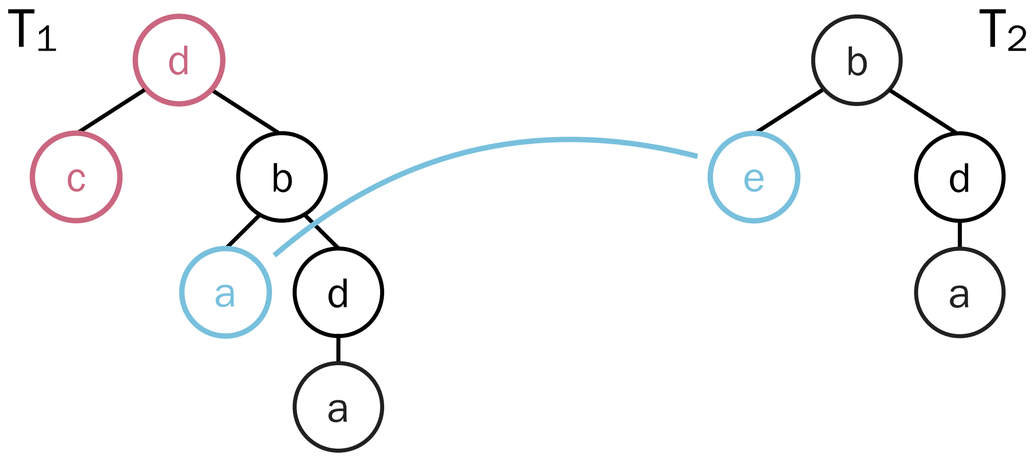
Tree Edit Distance (TED)

- **Definition:** the minimum number of edit operations that transform one tree into another.
- **Edit operations:**



Tree Edit Distance (TED)

- **Example:** trees T_1 and T_2 with $TED(T_1, T_2) = 3$.



- **Complexity:**
  - $O(n^3)$ time for ordered trees[4].
  - NP-hard for unordered trees[5].

JSON Data Format

- **Definition[6]:**
  - Objects: unordered collection of key-value pairs.
  - Arrays: ordered list of values.
  - Values: literals (e.g., string), arrays, and objects.

```
{
   "cast" : [
     "Ford",
     "Fisher"
   ],
   "running time" : 125,
   "name" : "Star Wars - A New Hope"
}
```

JSON document 1

Cumulative Thesis

- **“Effective filters and linear time verification for tree similarity joins”**
  Thomas Hütter, Mateusz Pawlik, Robert Löschinger, and Nikolaus Augsten,
  IEEE 35th International Conference on Data Engineering (ICDE), 2019.

- **“JEDI: These aren't the JSON documents you're looking for…”**
  Thomas Hütter, Nikolaus Augsten, Christoph M Kirsch, Michael J Carey, and Li Chen,
  Submitted to the International Conference on Management of Data (ACM SIGMOD), 2022.

- **“DeSignate: detecting signature characters in gene sequence alignments for taxon diagnoses”**
  Thomas Hütter, Maximilian H Ganser, Manuel Kocher, Merima Halkic, Sabine Agatha, and Nikolaus Augsten,
  BMC bioinformatics 21.1, 2020.

- **“A Link is not Enough – Reproducibility of Data”**
  Mateusz Pawlik, Thomas Hütter, Daniel Kocher, Willi Mann, and Nikolaus Augsten,
  Datenbank-Spektrum 19.2, 2019.

JSON Similarity Lookup

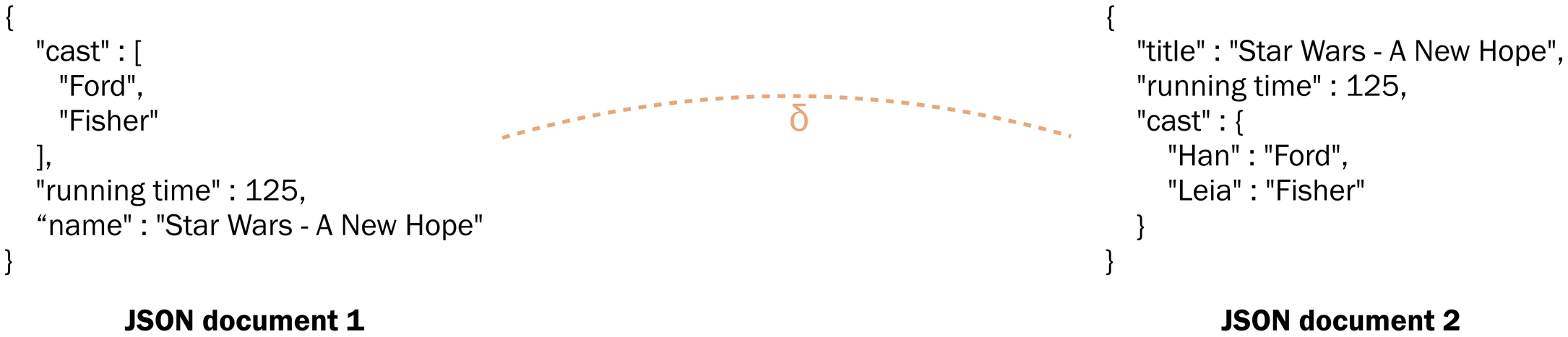
- **Definition:**
  - Given: distance function δ, user-defined threshold τ, and a query JSON document d_q.
  - Goal: retrieve all JSON documents d_i from a database D that are similar to d_q, i.e., $\delta(d_q, d_i) \leq \tau$.

Query document d_q

Database D

Distance Functions for JSON

- **Goal:** find a distance function δ for JSON documents.



JSON document 1 **JSON document 2**

- **Existing solutions:**
  - ✗ Ignore the structure (e.g., line-based approaches).
  - ✗ No quality guarantees (e.g., distance not minimal).

JSON Representation

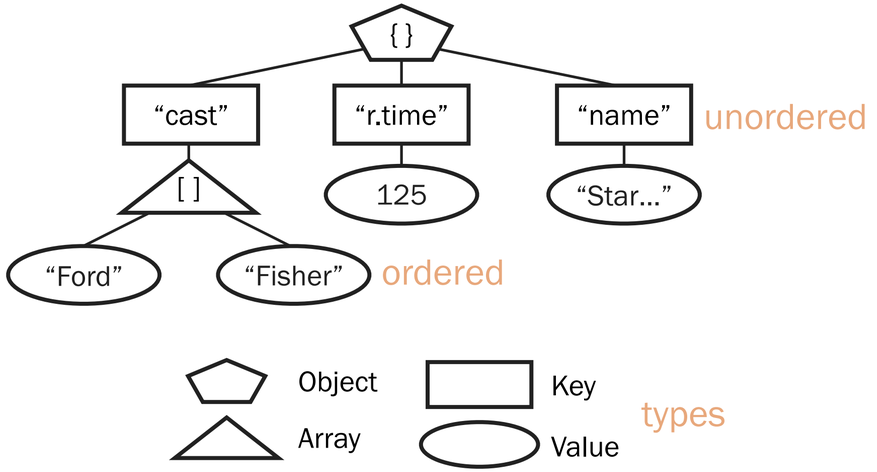
- **Goal:** find a lossless representation of JSON documents.

- **JSON trees:**

```
{
   "cast" : [
     "Ford",
     "Fisher"
   ],
   "running time" : 125,
   “name" : "Star Wars - A New Hope"
}
```

JSON document 1

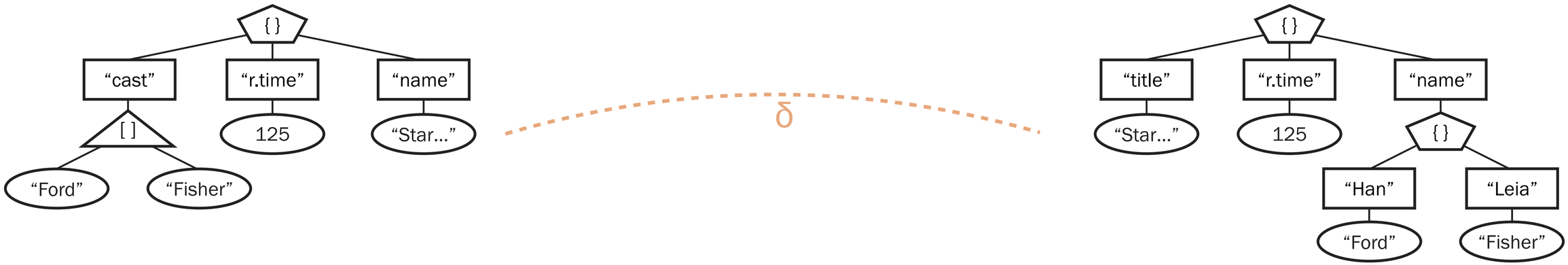
transformation



JSON tree 1

Distance Functions for JSON

- **Goal:** find a distance function δ for JSON trees.



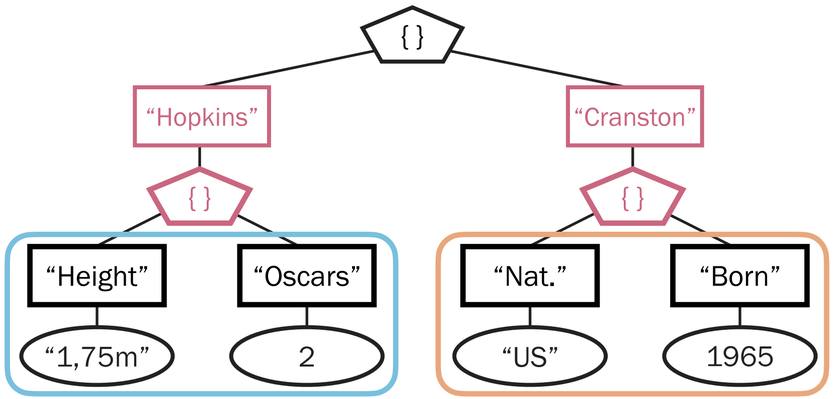
JSON tree 1

JSON tree 2

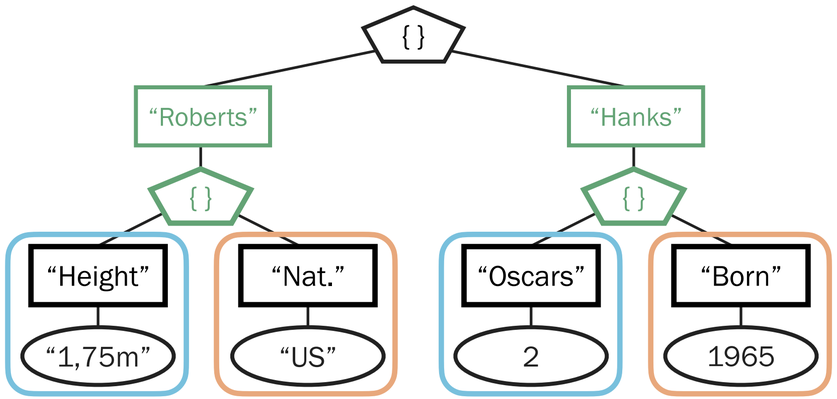
- **Existing solutions:**
  - ✗ TED is NP-complete for JSON trees.

Document-Preserving

- **Observation:** TED is too permissive.



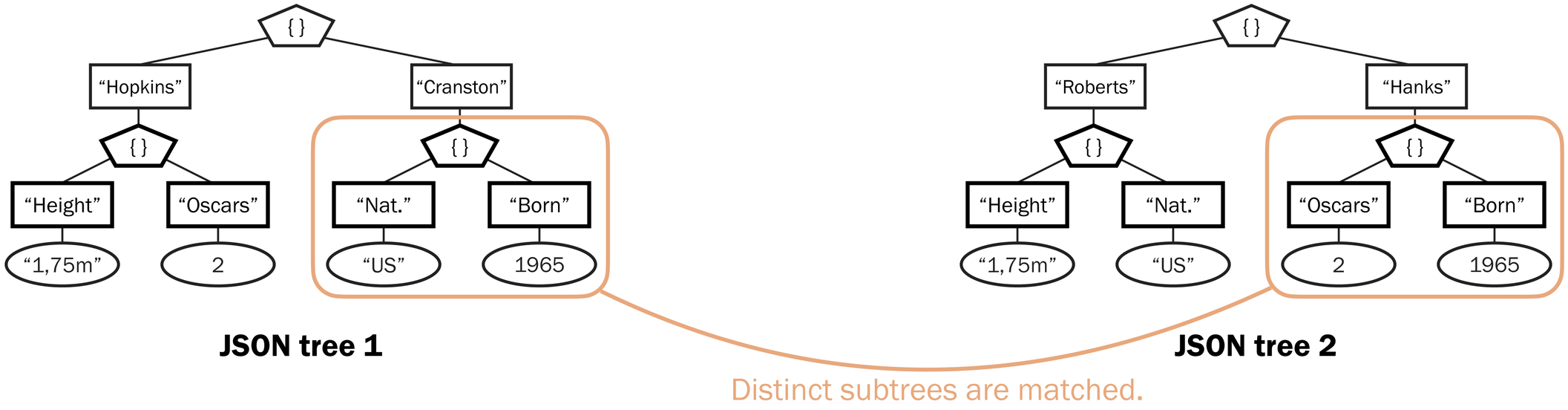
JSON tree 1



JSON tree 2

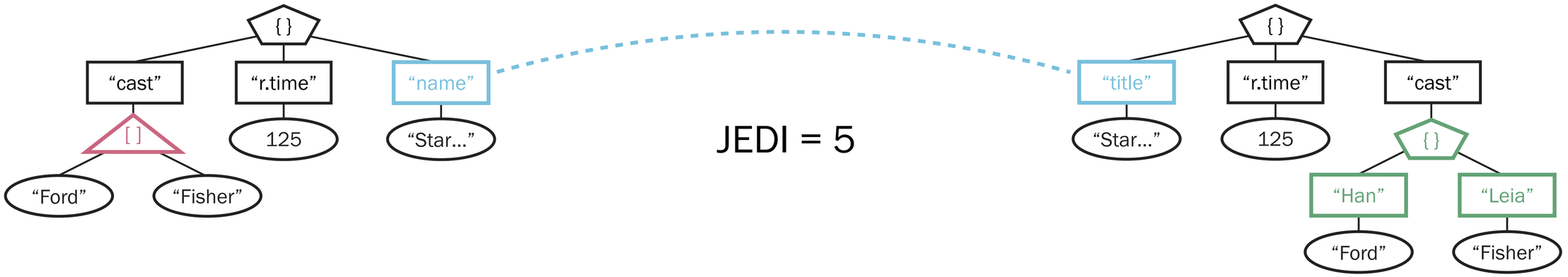
Document-Preserving

- **Intuition:** each subtree of a JSON tree is nested document.



JSON Edit Distance (JEDI)

- **Definition:** the minimum number of node edit operations (insert, delete, and rename) that transform one tree into the other satisfying the document-preserving constraint.



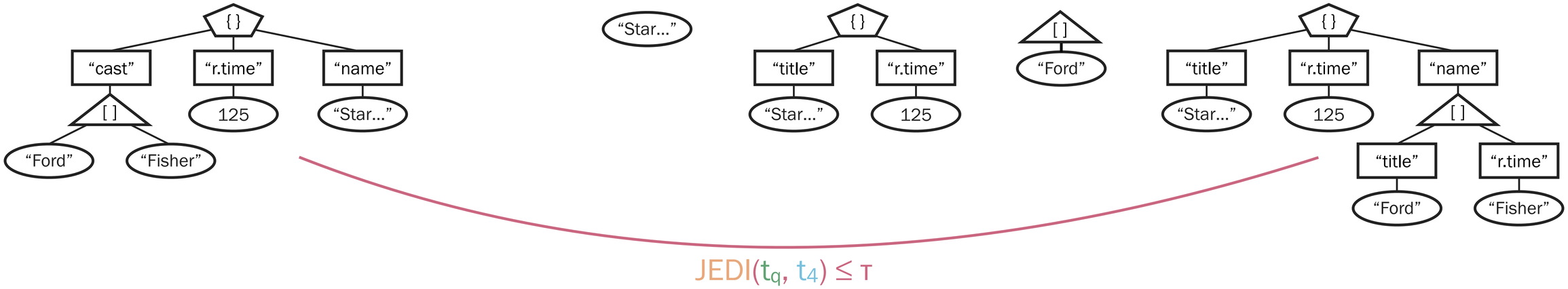
JSON tree 1 **JSON tree 2**

- **Baseline algorithm:** runs in $O(n^2 d \log(d))$ time and $O(n^2)$ space where n is the tree size and d the maximum degree of a tree[11,12].

JSON Similarity Lookup

- **Definition:**
  - Given: distance function JEDI, user-defined threshold τ, and a query JSON tree t_q.
  - Goal: retrieve all JSON trees t_i from a database D that are similar to t_q, i.e., $JEDI(t_q,t_i) \leq \tau$.

- **Example:** let $\tau = 5$.



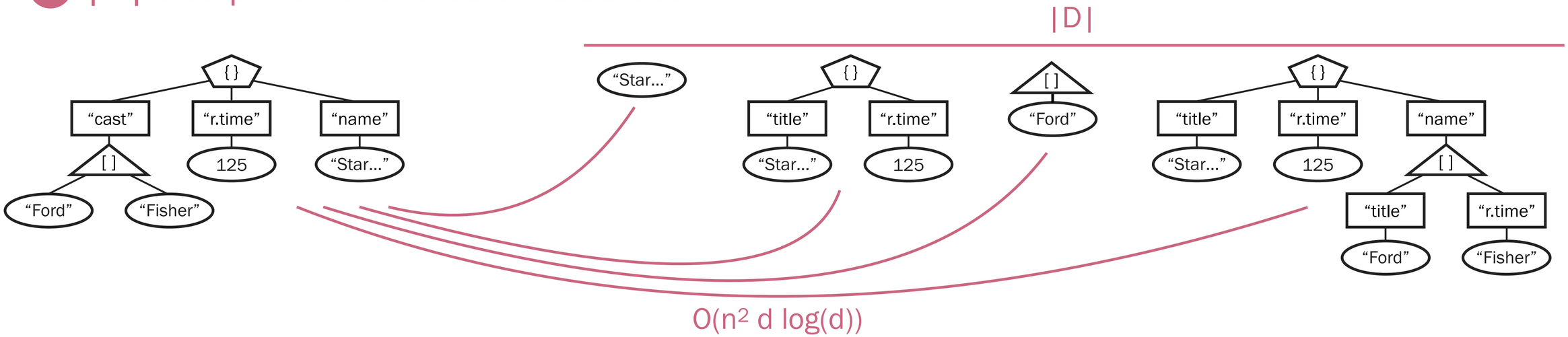
Query tree t_q

Database D

Challenges

- **Processing large JSON trees:**
  - ✗ JEDI verification takes $O(n^2\ d\ \log(d))$ time.

- **Processing large databases:**
  - ✗ $|D|$ tree pairs have to be considered.

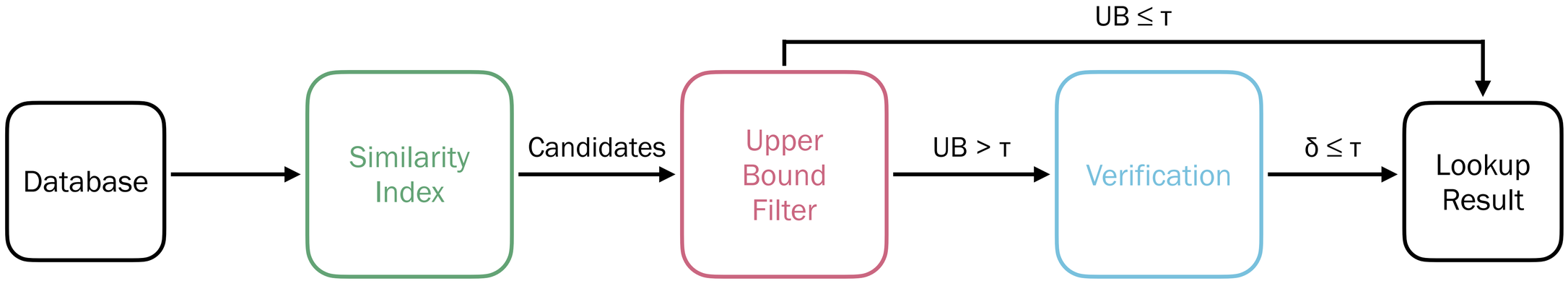


Query tree t_q

Database D

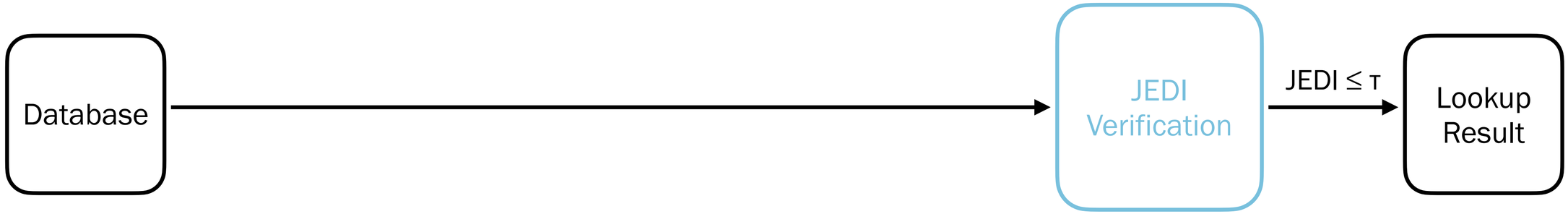
Filter and Verification Framework

- **Overview:**



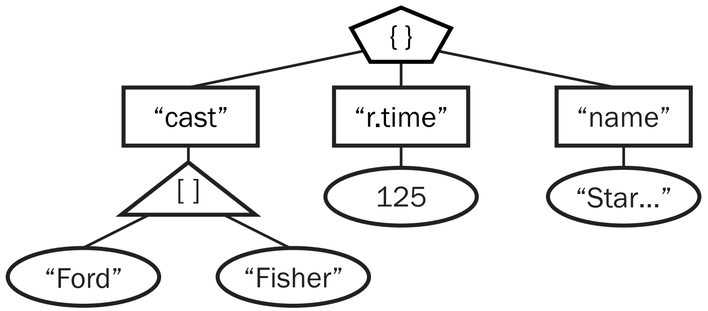
Filter and Verification Framework

- **Efficient verification:** QuickJedi reduces the runtime by up to one order of magnitude.

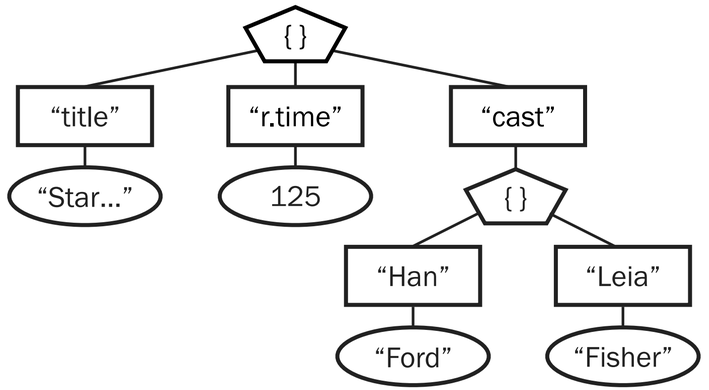


JEDI Baseline Algorithm

- **Dynamic programming algorithm:**
  - Process all node pairs in a bottom-up manner.
  - Compute the subtree distance for each node pair.



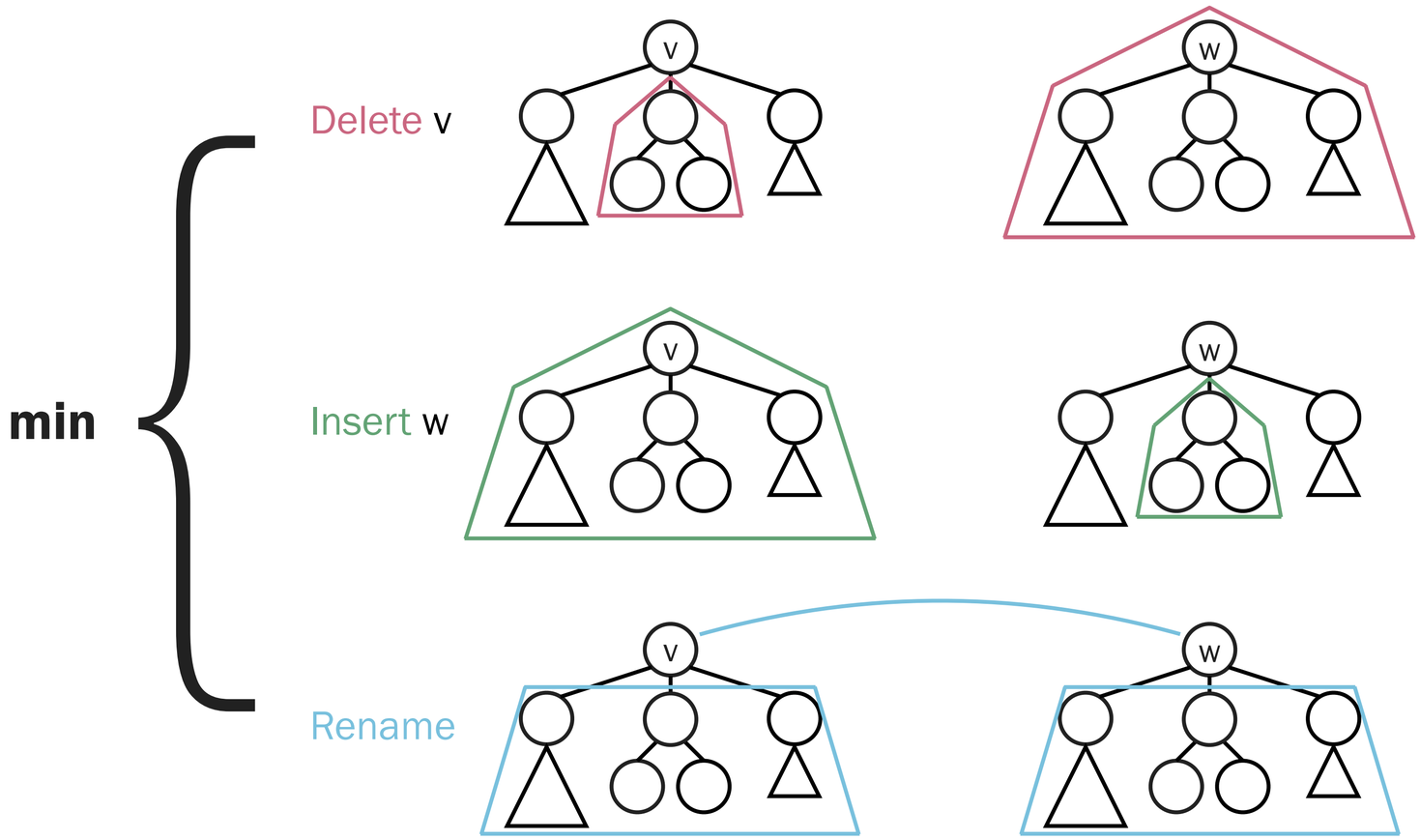
JSON tree 1



JSON tree 2

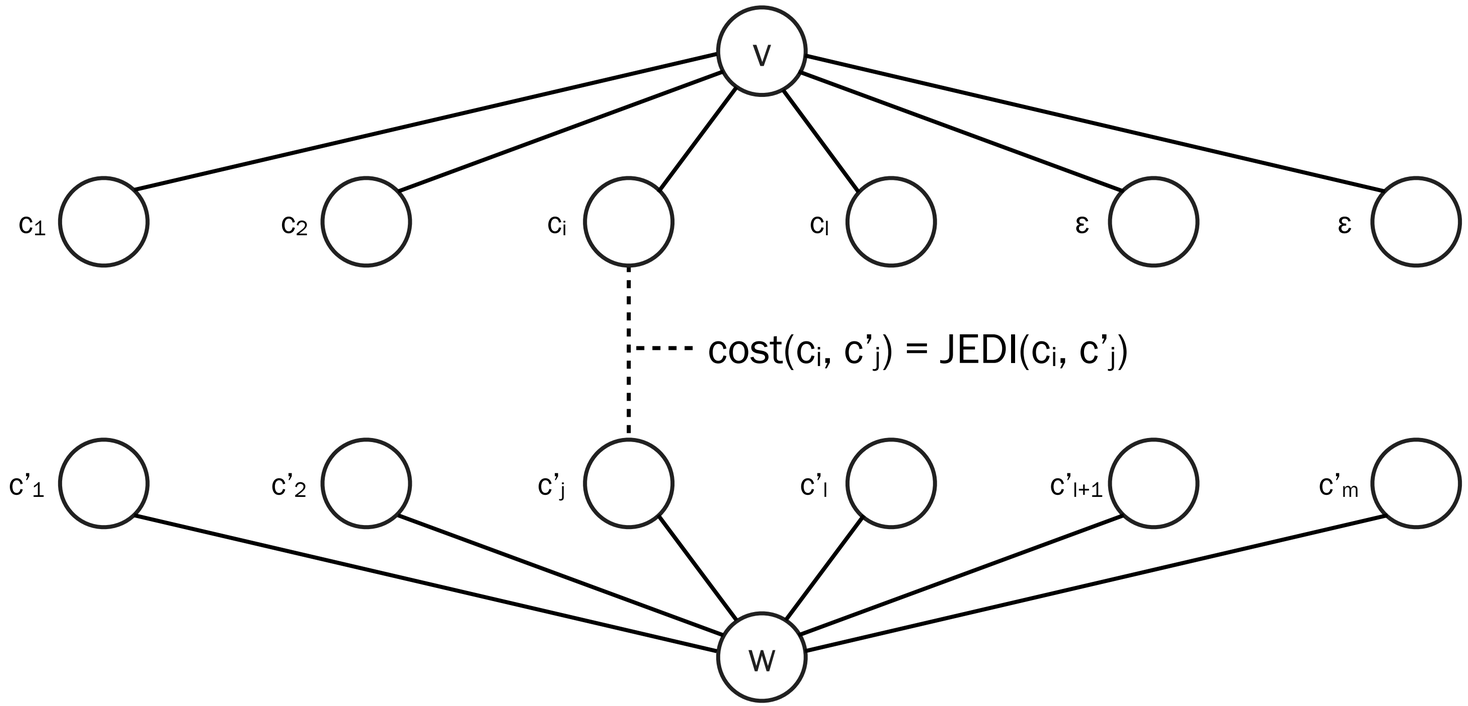
JEDI Baseline Algorithm

- **Subtree distance computation:** consider the edit operation with minimum distance.



JEDI Baseline Algorithm

- **Cost of children matching:**
  - v and w are arrays: sequence edit distance (quadratic runtime).
  - Otherwise: minimum-cost bipartite graph matching (cubic runtime).

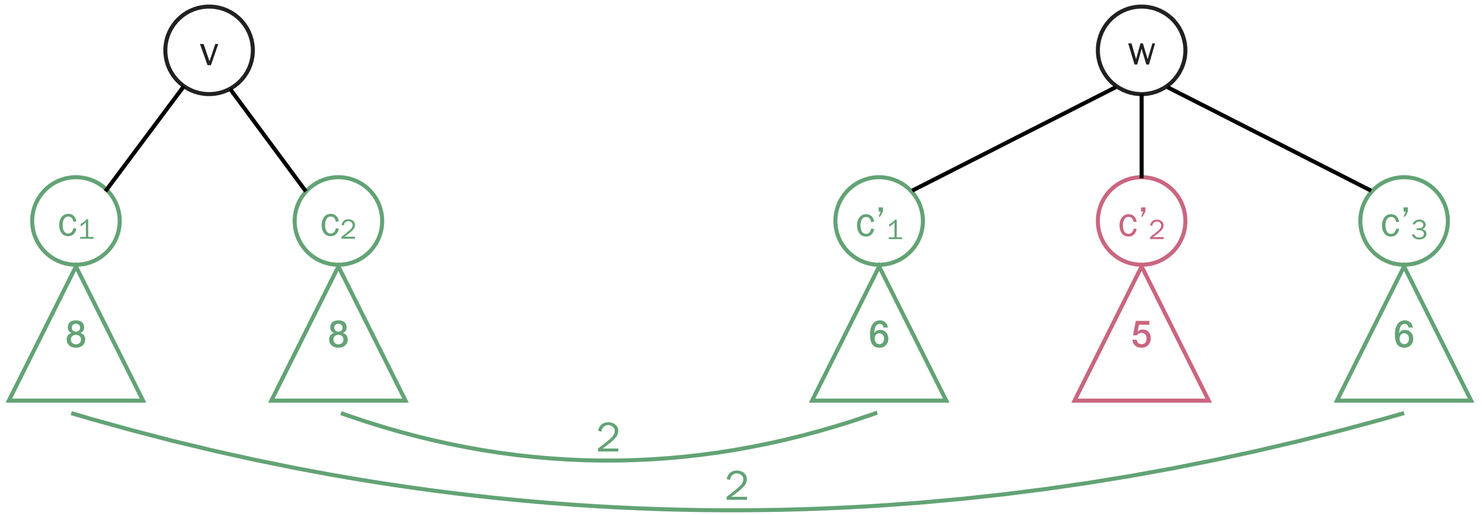


Skipping the Children Matching

- **Idea:** bound the rename costs. Skip if lower bound exceeds upper bound.
- **Upper bound:** insertion and deletion provide an upper bound.
- **Challenge:** identify a lower bound that is
  - efficient (applied for each node pair) and
  - effective (skip many matchings).

Aggregate Size Lower Bound

- **Key ideas:**
  - The k-smallest subtrees of the bigger amount of children are deleted.
  - The remaining subtrees are bounded by the size difference.

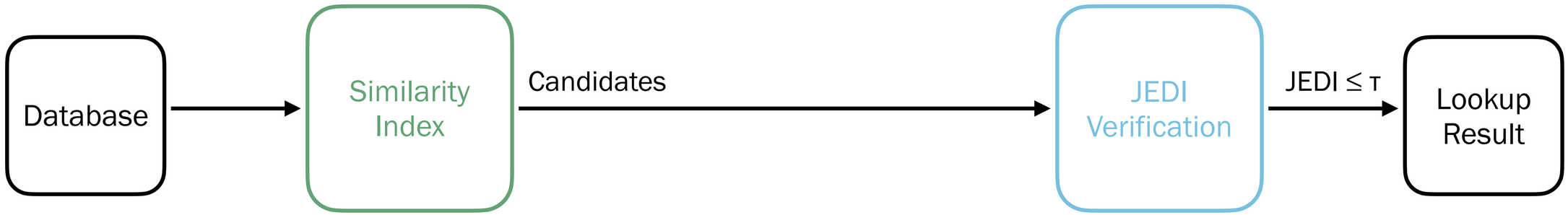


5 + 2 + 2 = 9

- **Efficiency:**
  - Maintain an array for constant time computation.

Filter and Verification Framework

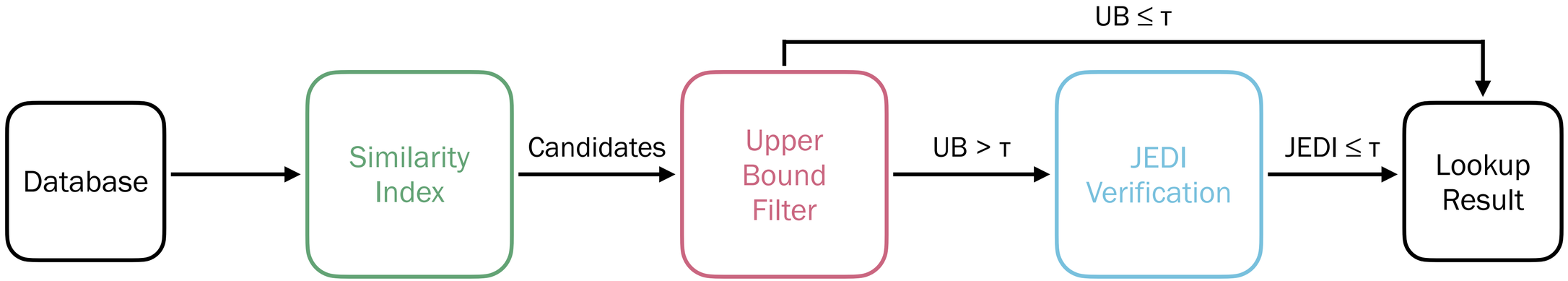
- **Process large databases:** JSIM is the first JSON similarity index.



QuickJedi skips the children matching and reduces the runtime by one order of magnitude.

Filter and Verification Framework

- **Process large trees:** JOFilter is a linear time and space upper bound filter.

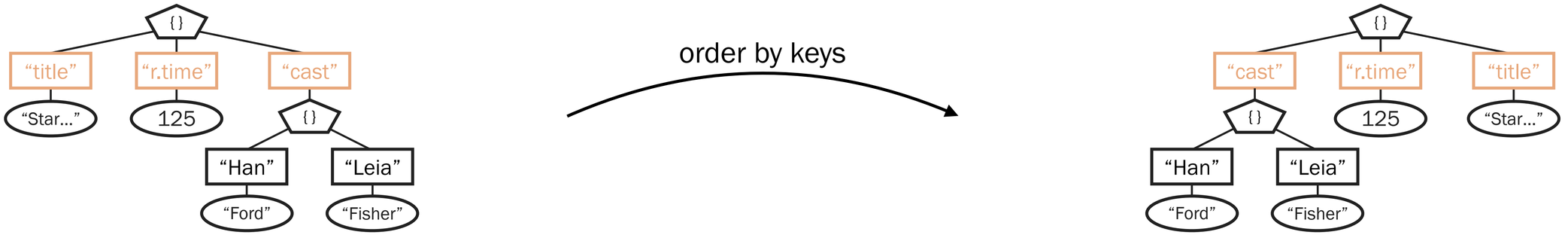


We introduced the first JSON similarity index.

QuickJedi skips the children matching and reduces the runtime by one order of magnitude.

JOFilter Upper Bound

- **Goal:** send candidates to the result set without verification.
- **Key idea:** apply the sequence edit distance on ordered keys.

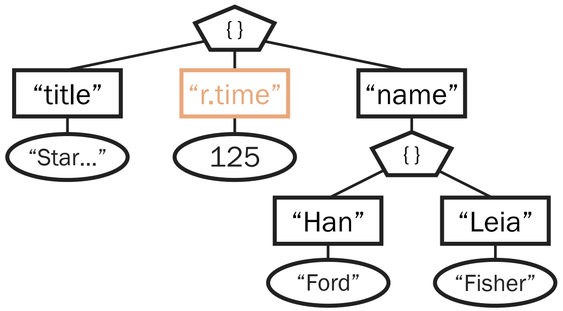


JSON tree 1 **JSON tree 2**

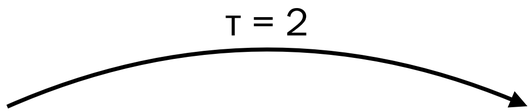
- **Related work:** fastest algorithm runs in quadratic time[8].
- **Challenge:** filter applied for each candidate.

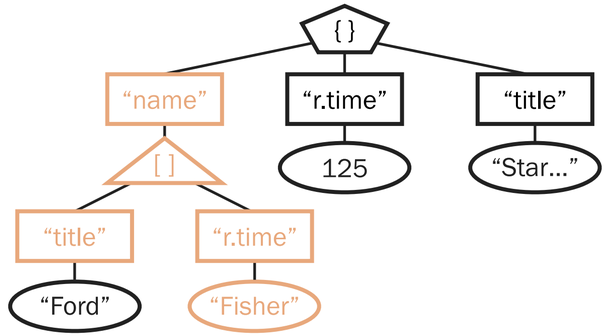
JOFilter Upper Bound

- **Key idea 2:** for a node in T_1 only $2\tau + 1$ nodes of T_2 have to be considered.



JSON tree 1



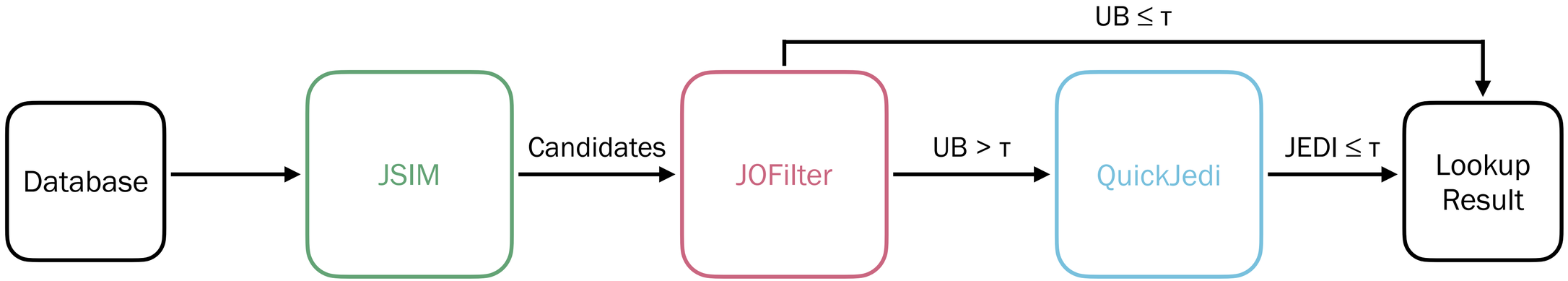


JSON tree 2

- **Results:** reduce the complexity to linear time and space.

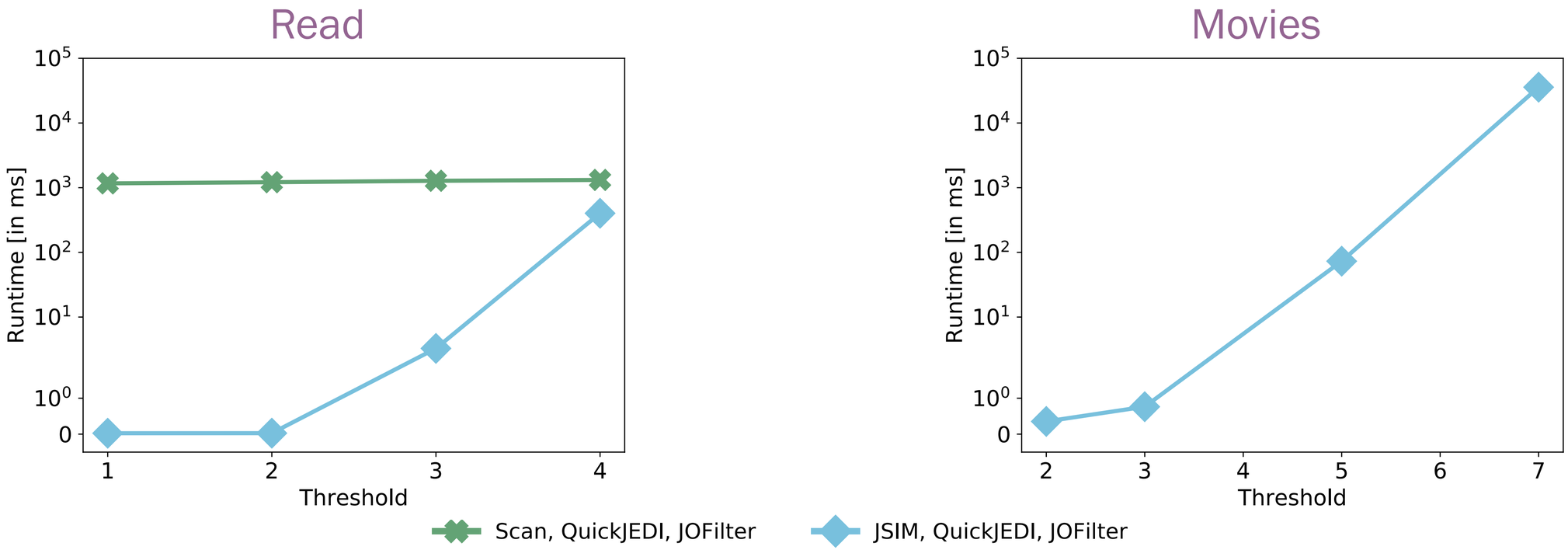
Filter and Verification Framework

- **JSON Similarity Lookup:**



Scaling to Large Databases

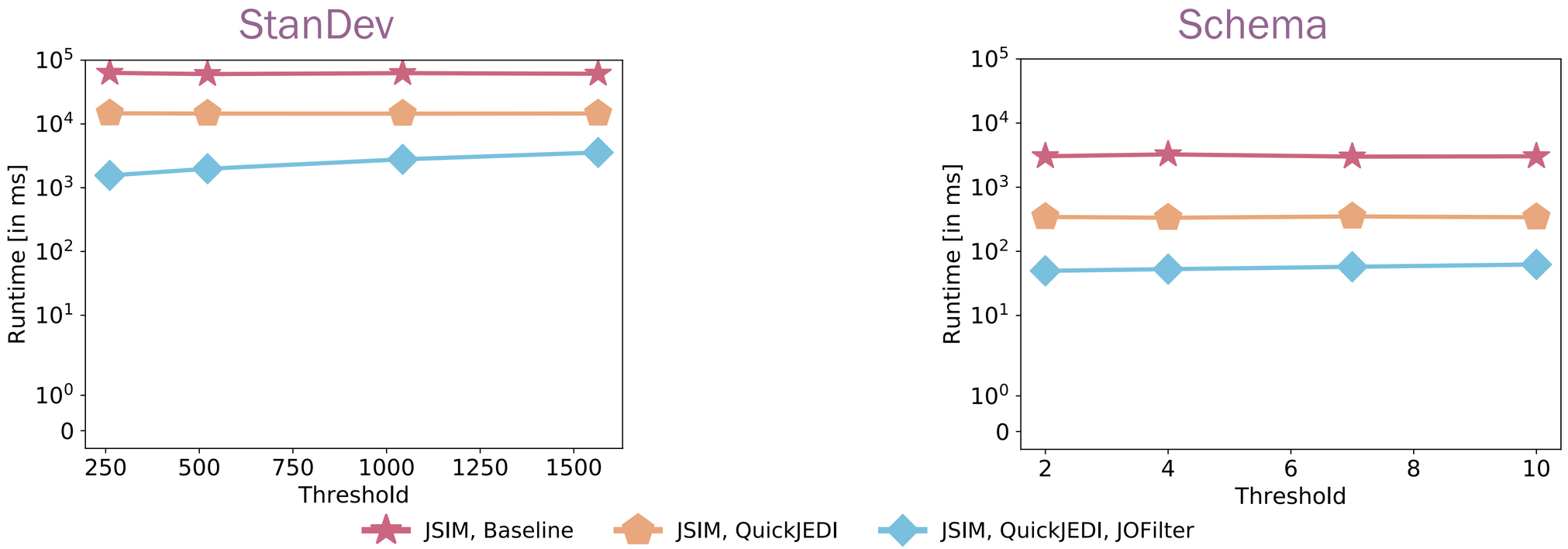
- **Datasets:** Read (30k trees) and Movies (8.7 million trees).



Result: a similarity index is needed to process large databases.

Scaling to Large Trees

- **Datasets:** StanDev (up to 18k nodes) and Schema (up to 48k nodes).



Result: QuickJedi and JOFilter reduce the runtime by up to two orders of magnitude.

References

[1] F. Li, H. Wang, L. Hao, J. Li, and H. Gao, “Approximate joins for XML at label level”, Information Sciences, 2014.

[2] T. Akutsu, “Tree edit distance problems algorithms and applications to bioinformatics”, IEICE Transactions on Information and Systems, 2010.

[3] Z. Lin, H. Wang, and S. I. McClean, “Measuring tree similarity for natural language processing based information retrieval”, International Conference on Applications of Natural Language to Information Systems, 2010.

[4] M. Pawlik and N. Augsten, “Tree edit distance: Robust and memory-efficient”, Information Systems, 2016.

[5] K. Zhang, R. Statman, and D. Shasha, “On the editing distance between unordered labeled trees”, Information Processing Letters, 1992.

[6] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format”, RFC 8259, 2017.

[7] R. Yang, P. Kalnis, and A. K. H. Tung, “Similarity evaluation on tree-structured data”, ACM SIGMOD International Conference on Management of Data, 2005.

[8] K. Kailing, H.-P. Kriegel, S. Schönauer, and T. Seidl, “Efficient similarity search for hierarchical data in large databases”, International Conference on Extending Database Technology, 2004.

[9] S. Guha, H. V. Jagadish, N. Koudas, D. Srivastava, and T. Yu, “Approximate XML joins”, ACM SIGMOD International Conference on Management of Data, 2002.

[10] Y. Tang, Y. Cai, and N. Mamoulis, “Scaling similarity joins over tree-structured data”, Proceedings of the VLDB Endowment, 2015.

[11] K. Zhang, “Algorithms for the constrained editing distance between ordered labeled trees and related problems”, Pattern recognition, 1995.

[12] K. Zhang, “A constrained edit distance between unordered labeled trees”, Algorithmica, 1996.