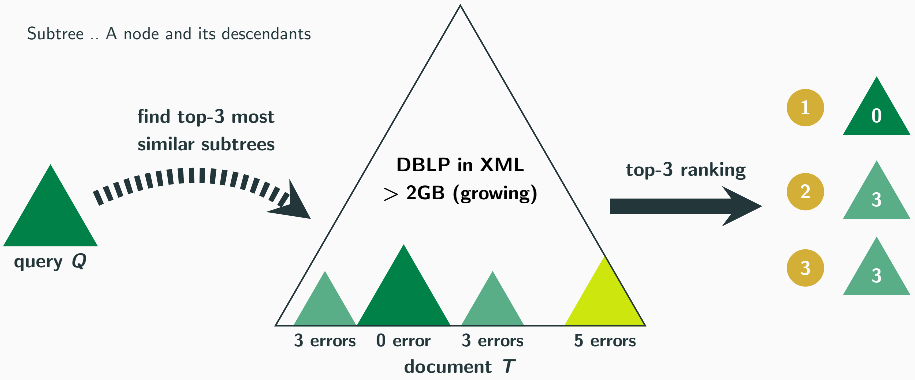
A Scalable Index for Top-k Subtree Similarity Queries

Daniel Kocher* and Nikolaus Augsten

Amsterdam, July 4, 2019

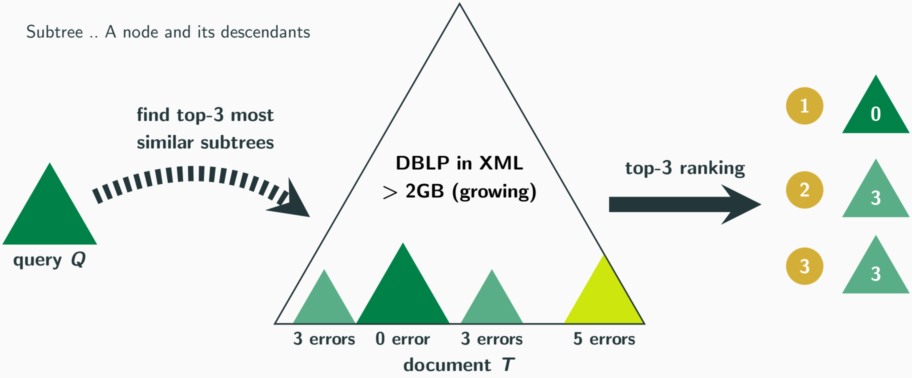
Department of Computer Sciences, University of Salzburg, Austria

Top-3 Subtree Similarity Query



Find ***k* most similar subtrees** for **query *Q*** in **large document *T***

Top-3 Subtree Similarity Query



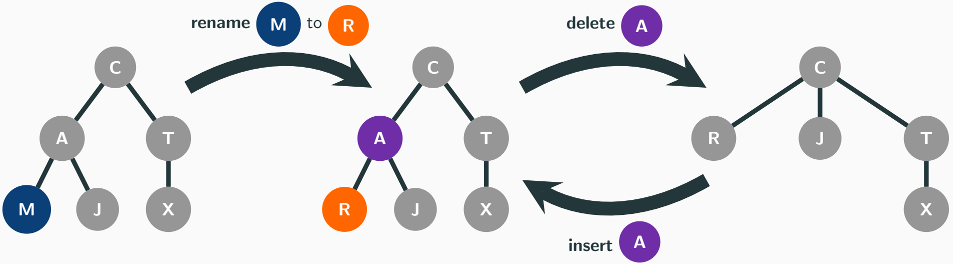
Find ***k*** **most similar subtrees** for **query *Q*** in **large document *T***

Fast queries ⧫ Scale to large documents ⧫ Support updates

Subtree Scoring Function

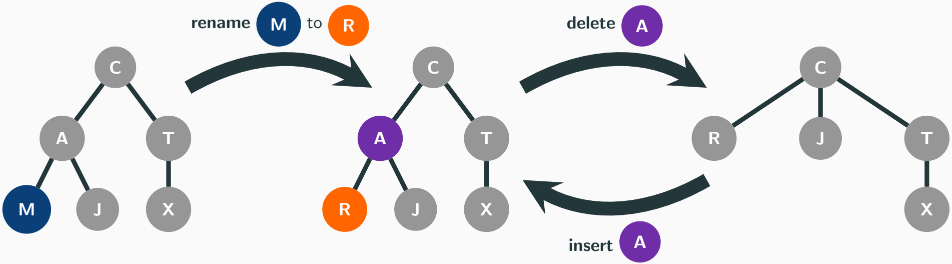
Subtree Scoring Function

Tree edit distance: **Minimum number** of **node edit operations** that **transform** one **tree into another**



Subtree Scoring Function

Tree edit distance: **Minimum number** of **node edit operations** that **transform** one **tree into another**



Computation: $\mathcal{O}\left(n^3\right)$ **time** and $\mathcal{O}\left(n^2\right)$ **space**

State-of-the-Art Solutions

State of the Art

	Querying	Memory Footprint	Index Updates
Index-free[1]	**Slow (doc. scan)**	Low	-
Index-based[2]	Fast	**High (quadratic)**	**No**

[1] Augsten et al. TASM: Top-k Approximate Subtree Matching. IEEE ICDE. 2010.

[2] Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.

SlimCone Index

Efficient ■ Linear Space ■ Updatable

Candidate Generation

Algorithmic Model

Linear-Space Index

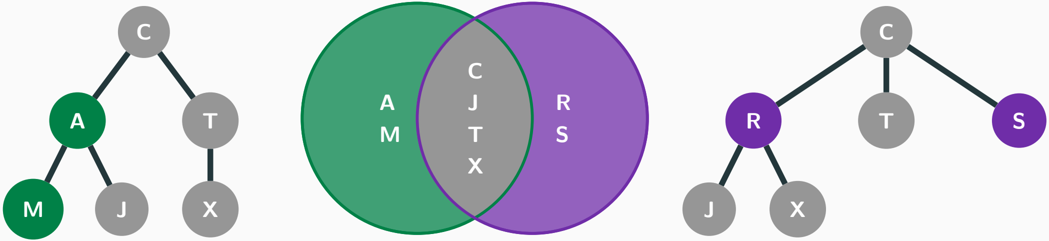
SlimCone Index

Efficient ■ Linear Space ■ Updatable

Candidate Generation | **Algorithmic Model** | **Linear-Space Index**

Background

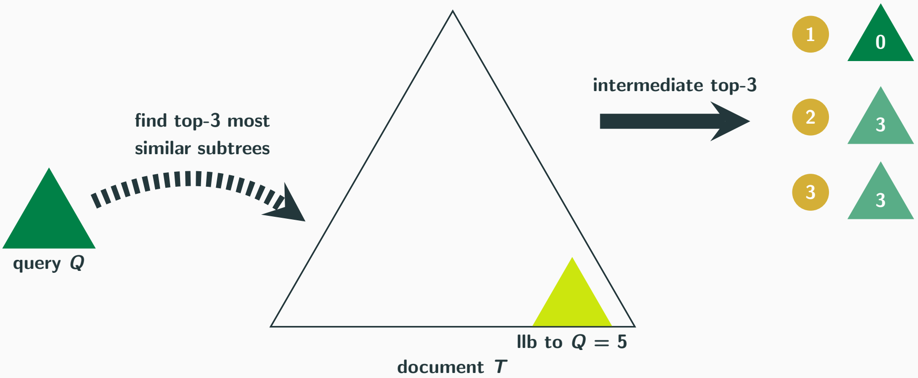
Label lower bound llb: Minimum edit distance based on **label information**



2 renames for **equal label sets**

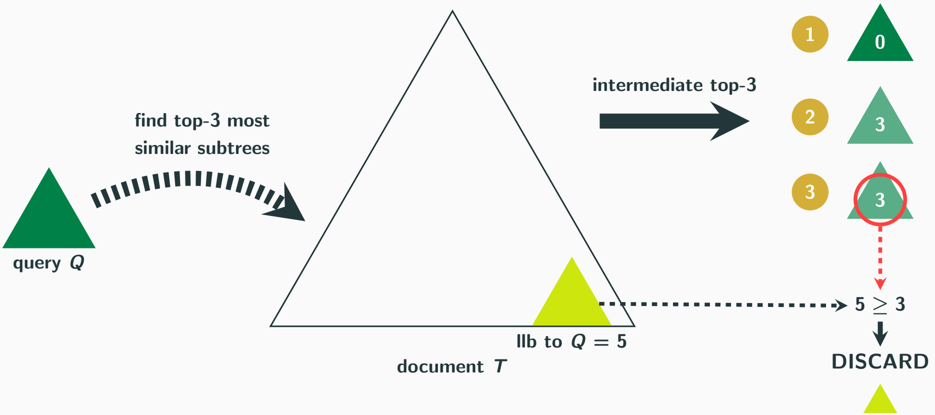
Effective Candidate Generation (1)

Ranking filter: **Worst edit distance** in intermediate **ranking** serves as **filter**

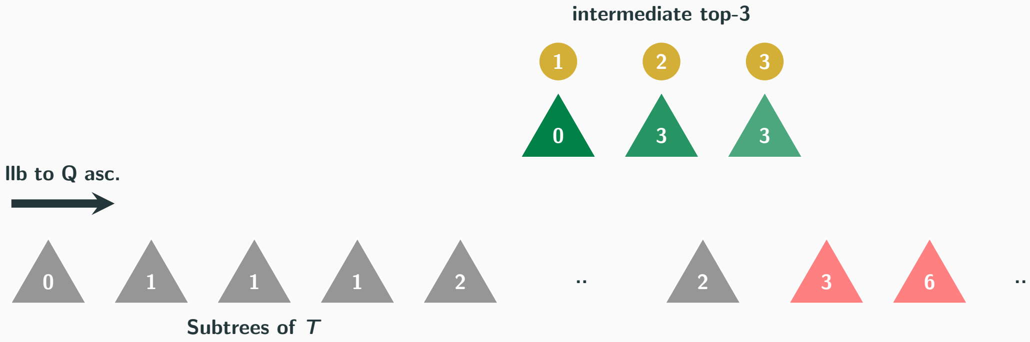


Effective Candidate Generation (1)

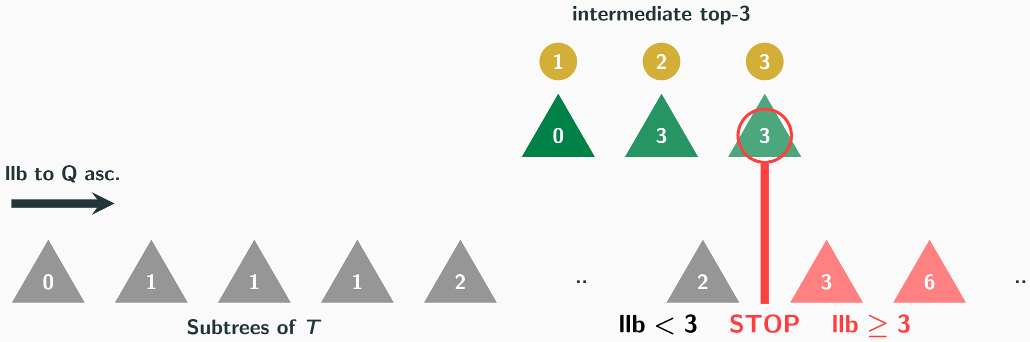
Ranking filter: **Worst edit distance** in intermediate **ranking** serves as **filter**



Effective Candidate Generation (2)



Effective Candidate Generation (2)



Early Termination: **Skip** all **subtrees** with **llb** **larger** or **equal** to **worst distance**

SlimCone Index

Efficient ■ Linear Space ■ Updatable

Candidate Generation

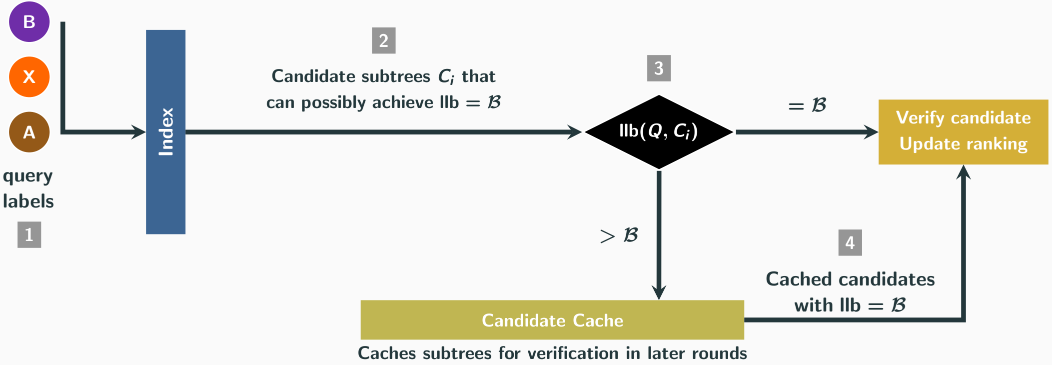
Algorithmic Model

Linear-Space Index

Round-based Algorithmic Model

Distance bound $\mathcal{B}$: **Starts** from **0** and is **incremented** in **each round**

Intuition: **Round 1** unveils **all subtrees** that have **llb equal to 0**



SlimCone Index

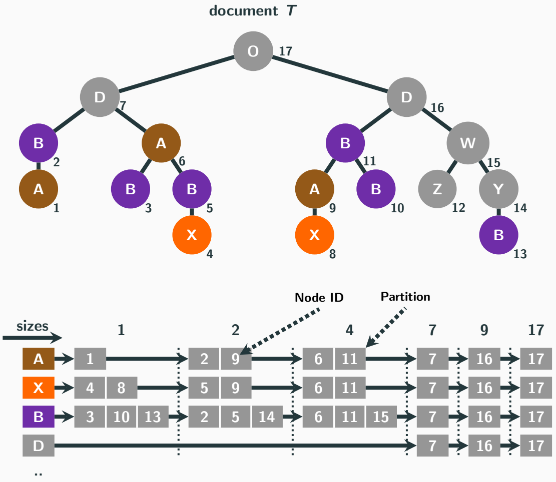
Efficient ■ Linear Space ■ Updatable

Candidate Generation

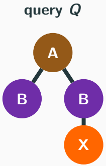
Algorithmic Model

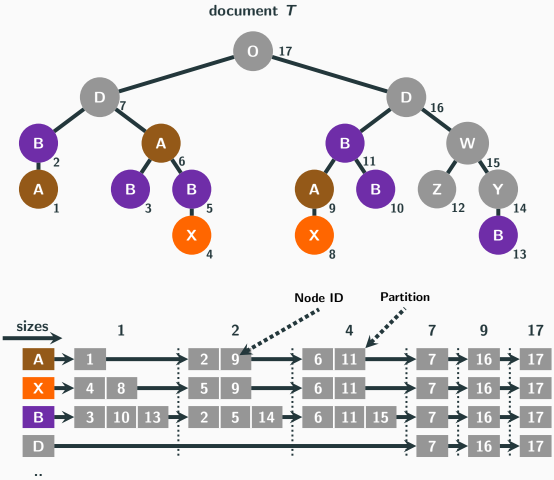
Linear-Space Index

Candidate Index

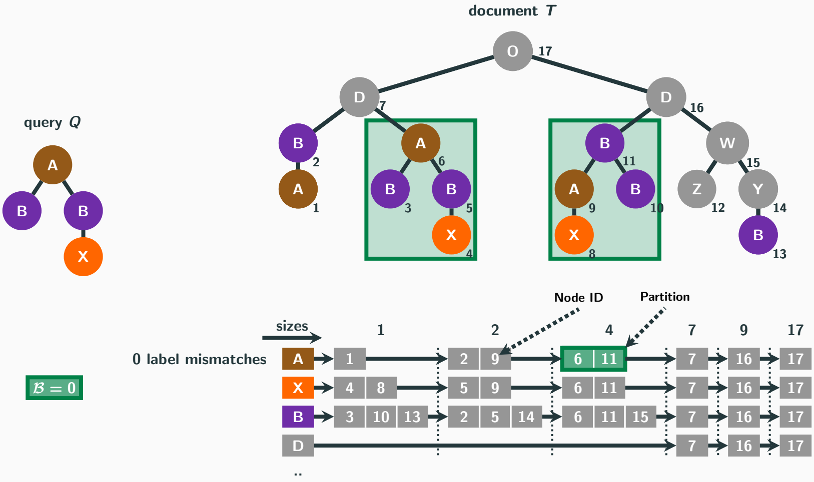


Candidate Index

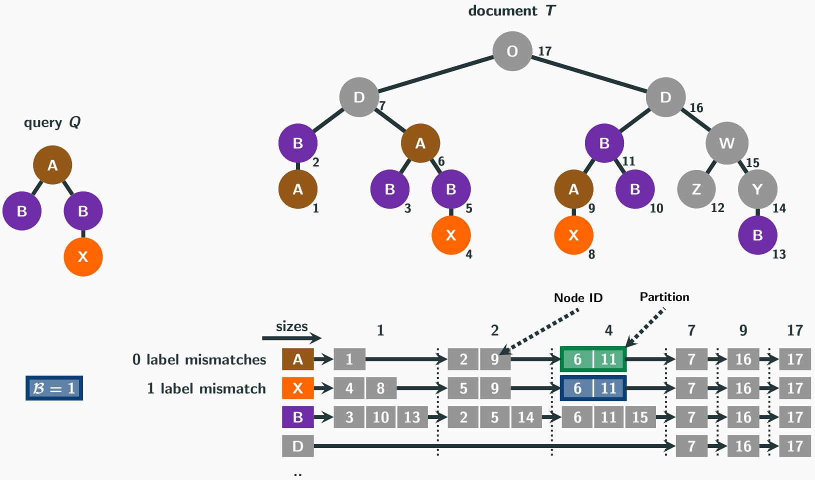




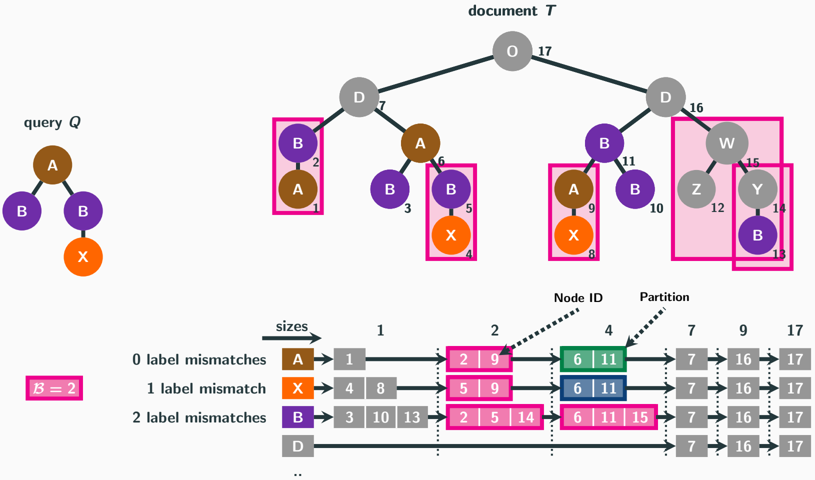
Candidate Index



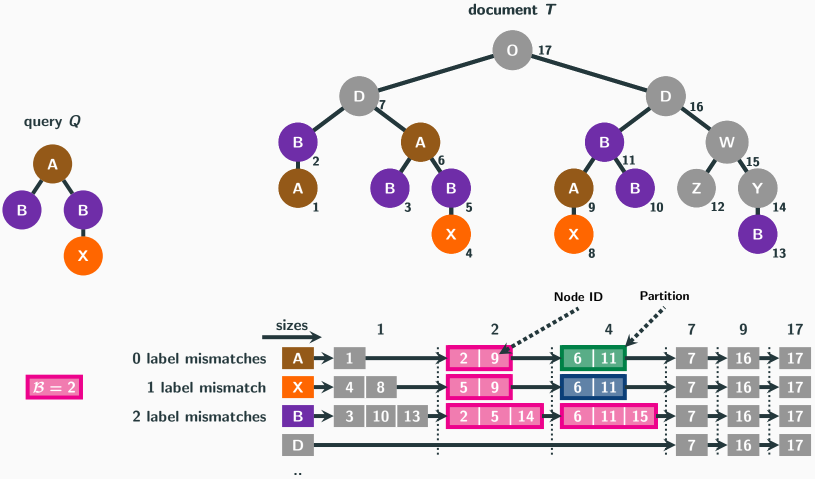
Candidate Index



Candidate Index



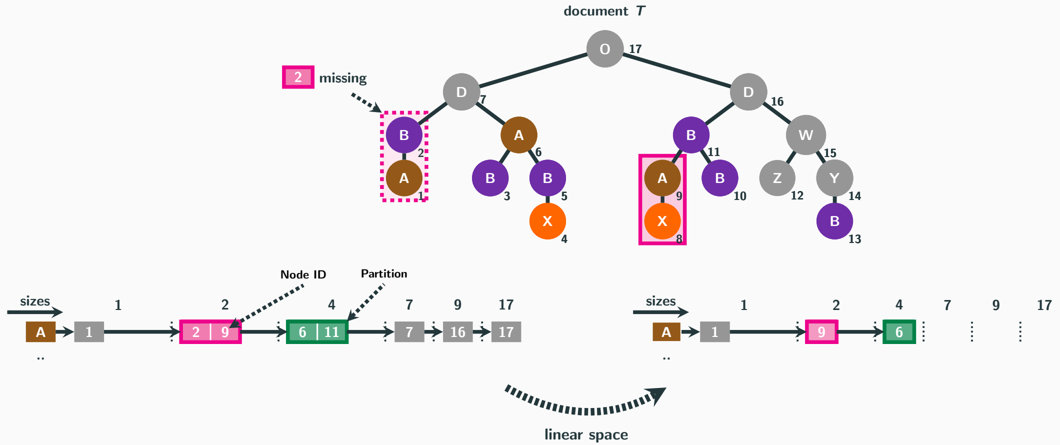
Candidate Index



Worst case: **Quadratic space** in the document size

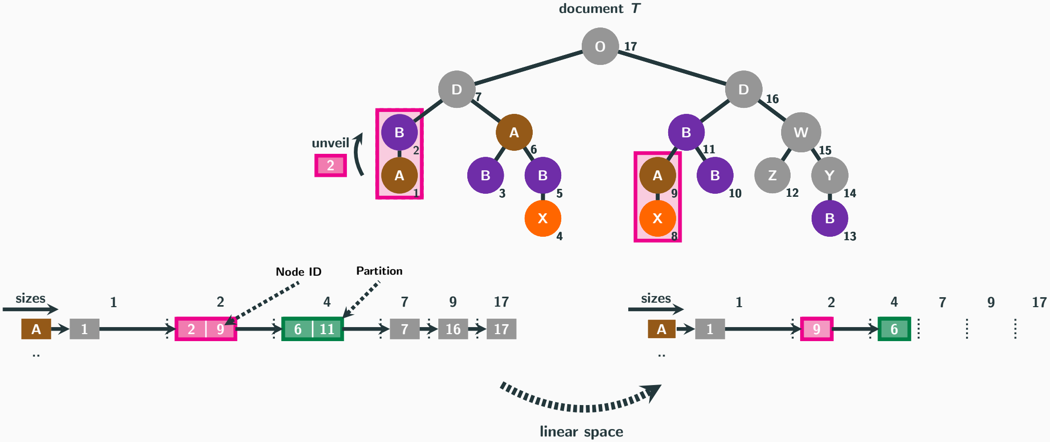
Linear-Space Candidate Index

(a) Store only **nodes having** a particular **label** and **(b)** build **partitions on the fly**



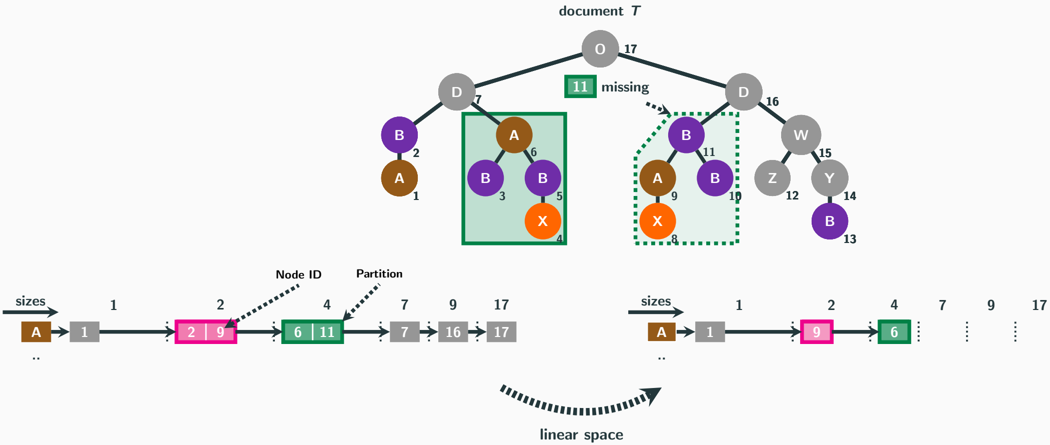
Linear-Space Candidate Index

(a) Store only nodes having a particular label and (b) build partitions on the fly



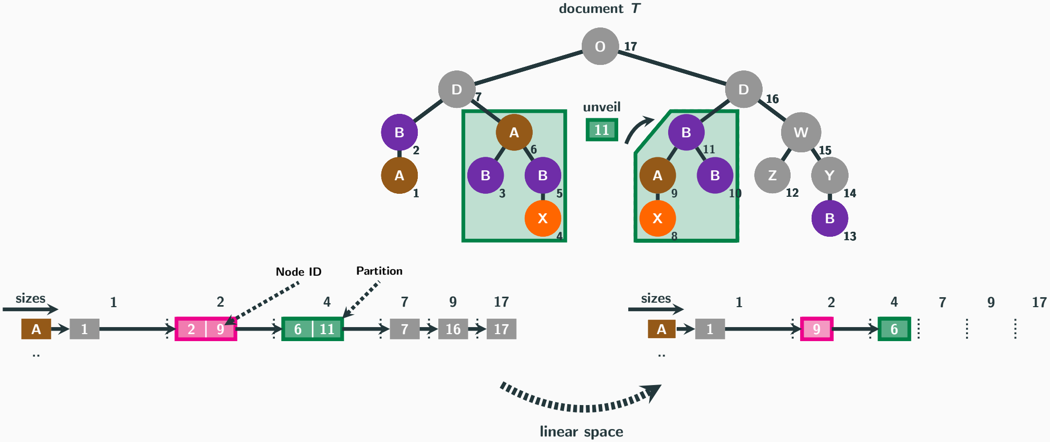
Linear-Space Candidate Index

(a) Store only nodes having a particular label and (b) build partitions on the fly



Linear-Space Candidate Index

(a) Store only **nodes having** a particular **label** and **(b)** build **partitions on the fly**



Empirical Evaluation

Experiments

Data Set	Size in Nodes
XMark	3.6 – 57.8 Mio.
TreeBank TB	3.8 Mio.
DBLP	126.5 Mio.
SwissProt SP	479.3 Mio.

State of the Art

TASM[1] index-free
STRUCT[2] index-based

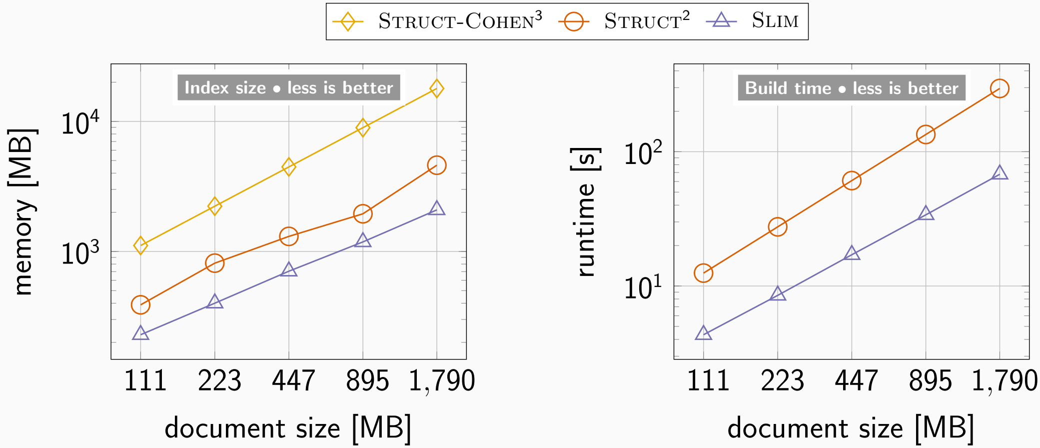
Our Solution

SLIM[3] index-based

Memory Scalability ♦ Efficiency ♦ Effectiveness

[1] Augsten et al. TASM: Top-*k* Approximate Subtree Matching. IEEE ICDE. 2010.
[2] Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.
[3] Kocher and Augsten. A Scalable Index for Top-*k* Subtree Similarity Queries. ACM SIGMOD. 2019.

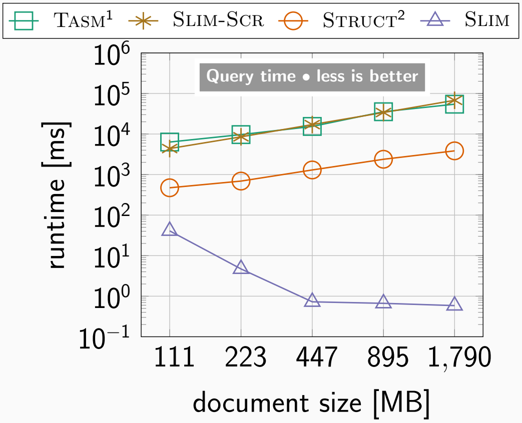
Memory Scalability (for varying document size)



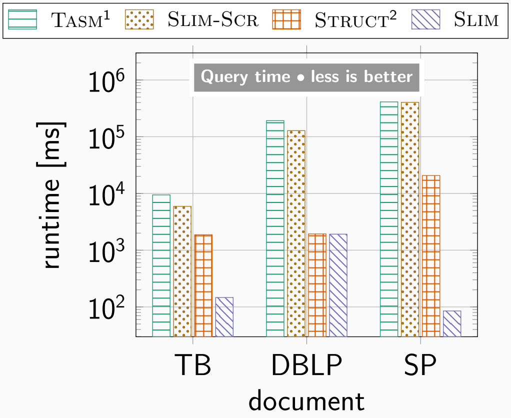
[2] Our C++ implementation of *Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.*

[3] Memory estimated according to *Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.*

Efficiency (for varying document size)



$|Q| = 16, \ k = 10$

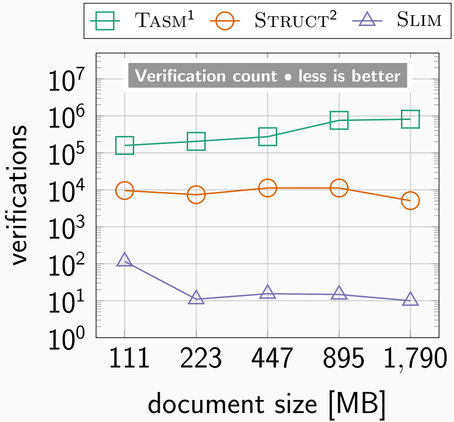


$|Q| = 16, \ k = 10$

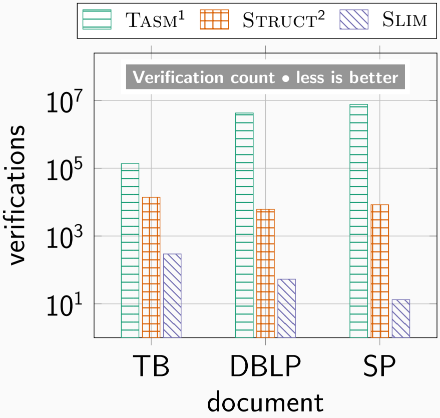
[1] Our C++ implementation of *Augsten et al. TASM: Top-k Approximate Subtree Matching. IEEE ICDE. 2010.*

[2] Our C++ implementation of *Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.*

Effectiveness (for varying document size)



$|Q| = 16,\ k = 10$



$|Q| = 16,\ k = 10$

[1] Our C++ implementation of *Augsten et al. TASM: Top-k Approximate Subtree Matching. IEEE ICDE. 2010.*

[2] Our C++ implementation of *Cohen. Indexing for Subtree Similarity-Search Using Edit Distance. ACM SIGMOD. 2013.*

Conclusion

Conclusion

- **Novel index-based approach** for **top-k subtree similarity queries**
- **Algorithmic model** that supports **effective candidate generation**
- Guaranteed **linear-space index**

Fast queries ♦ Scale to large documents ♦ Support updates

Thank you! Questions?

Contact:

Daniel Kocher	dkocher [at] cs.sbg.ac.at
Nikolaus Augsten	nikolaus.augsten [at] sbg.ac.at

database research group

UNIVERSITY of SALZBURG

FWF Der Wissenschaftsfonds.