Tsinghua University

Two Birds on One Stone:
An Efficient Hierarchical Framework for Top-k and Threshold-based String Similarity Search

Jin Wang, Guoliang Li, Dong Deng,
Yong Zhang, Jianhua Feng

Problem Definition

Edit Distance:
The minimum number of edit operations(insertion/deletion/substitution) needed to transform one string to another string.
For example: ED(*brother,* breather) = 2

brother
↓ *add e*
breother
↓ *substitute o with a*
breather

- **Threshold-based String Similarity Search:** *Given a string set S a query string q and threshold τ, threshold-based string similarity search finds all strings s ∈ S that ED(s, q) ≤ τ.*
- **Top-k String Similarity Search:** *Given a string set S and a query string q, top-k string similarity search returns a string set R ⊆ S such that |R|=k and for any string r ∈ R and s ∈ S – R, ED(r, q) ≤ ED(s, q).*

τ=1, k=2, q = "brothor"

ID	string	Length
s1	brother	7
s2	brothel	7
s3	broathe	7
s4	breathes	8
s5	swingable	9
s6	deduction	9

Application:
- Data cleaning & Data integration
- Spell Checking
- Copy Detection
- Entity Linking
- Macromolecules Sequence Alignment

....

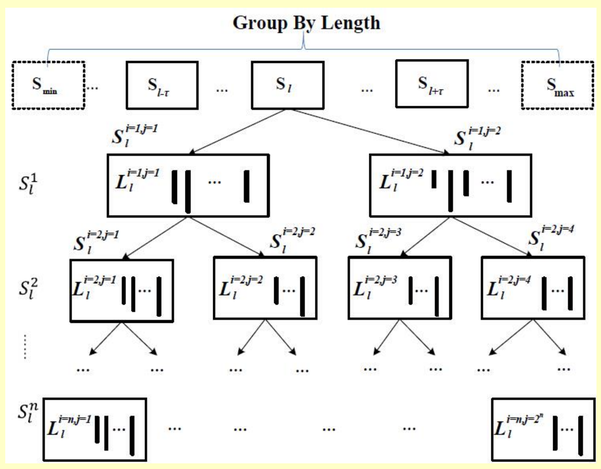
The HS-Tree Index

Iteratively String Partition:

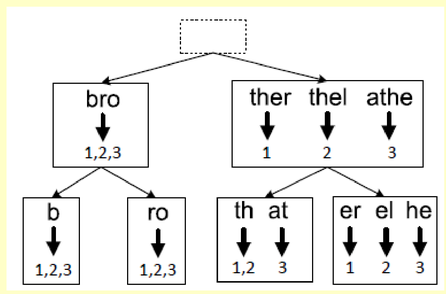
Two disjoint segments, prefix and suffix

Until we reach a level that has segments of length 1

Generate tree nodes and inverted lists



HS-Tree index

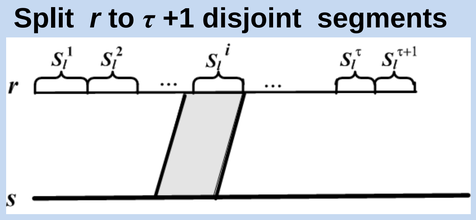


An example of group 7 in HS-Tree index

Extend Segment Filter

Pass-Join: Segment Filter

Split *r* to τ +1 disjoint segments



limitation:

Need a threshold before index construction

Can't support top-k similarity search

Locate level $i = \log_2(\tau + 1)$

Level i has 2^i segments

2^i-τ common segments between *s and q?*

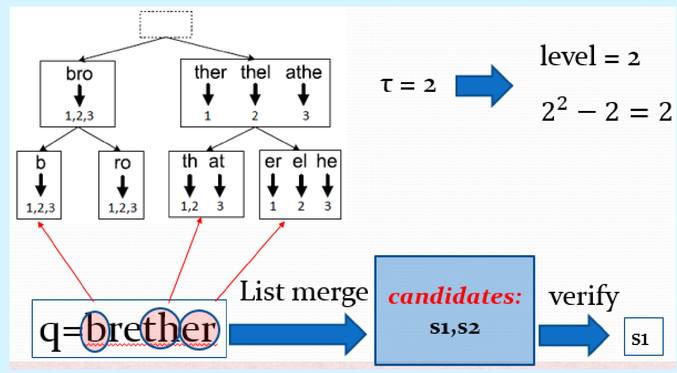
Yes → s is a candidate

No → s is pruned

- Threshold τ
- Query q
- Data s

Threshold-based Similarity Search Algorithm

Search:
1. Locate the right level
2. Generate substrings of query
3. Probe the inverted list, count the number of matched segments
4. Generate candidates
5. Perform verification

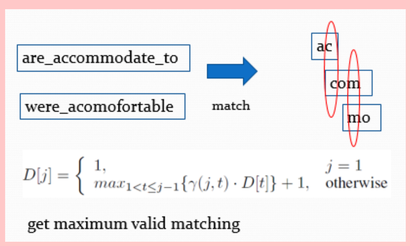


Improving Filter and Verification Step

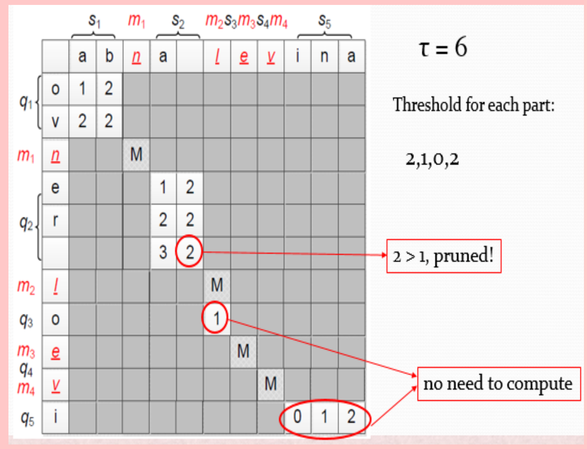
Reduce number of substrings:

$$[\max(P_j - (j-1),\ P_j + \Delta - (2^i - j)),\ \min(P_j + (j-1), P_j + \Delta + (2^i - j))]$$

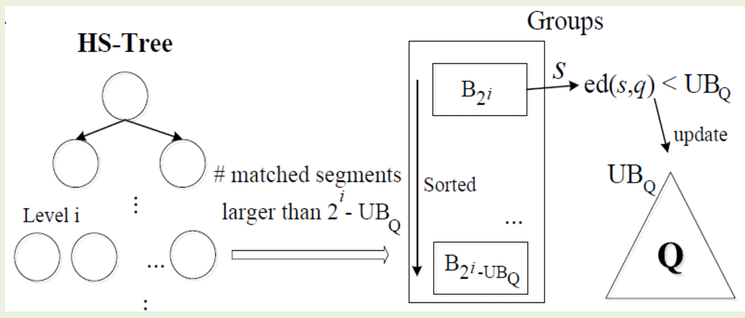
Remove invalid matching:



Improve Verification:
Multi-Extension method

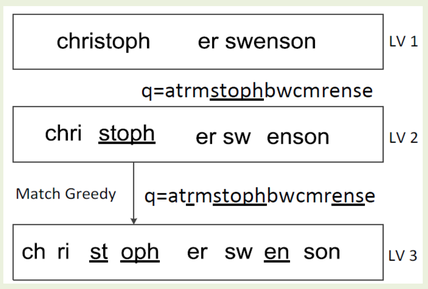


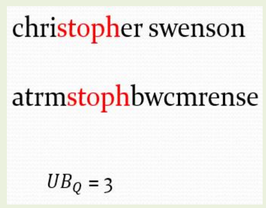
Top-k Similarity Search Algorithm



Batched Pruning

Avoid duplicate search





Greedy Matching

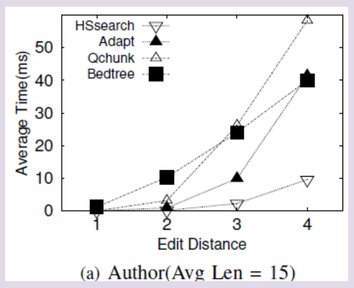
Eliminate consecutive errors within a segment

Experiments

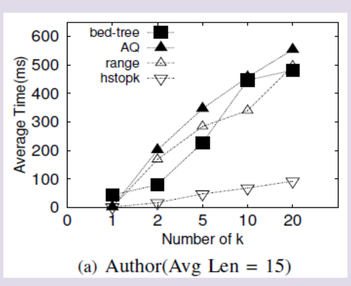
Settings:
C++, g++ 4.8.2 with -O3 flags
64bit Ubuntu Server 12.04 LTS version
Intel Xeon E5-2650 2.00GHz processor and 32GB memory.

TABLE II. DATASETS

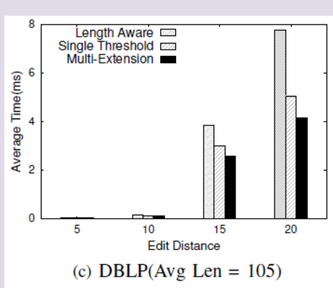
Datasets	#	Avg Len	Max Len	Min Len
Author	612,781	15	46	6
Query Log	464,189	45	522	30
DBLP	1,385,925	105	1626	1



(a) Author(Avg Len = 15)



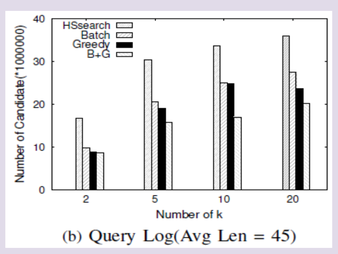
(a) Author(Avg Len = 15)



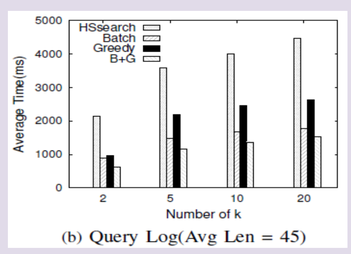
(c) DBLP(Avg Len = 105)

Comparison with State-the-art

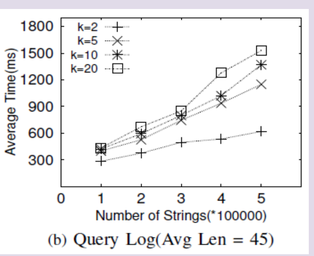
Evaluate Verification



(b) Query Log(Avg Len = 45)



(b) Query Log(Avg Len = 45)



(b) Query Log(Avg Len = 45)

Evaluating Filters for top-k search

Scalability