ACM SIGMOD Programming Contest 2017

WordTrie-Based Dynamic Dictionary Matching

Team PartTimeContestants: Frank Tetzel, Ismail Oukid, Thomas Bach

Dynamic Dictionary Matching

Dynamic dictionary D:

Set of patterns with operations

- insert(String)
- delete(String)
- contains(String) -> Boolean

Find all occurrences of any pattern of D in a text T

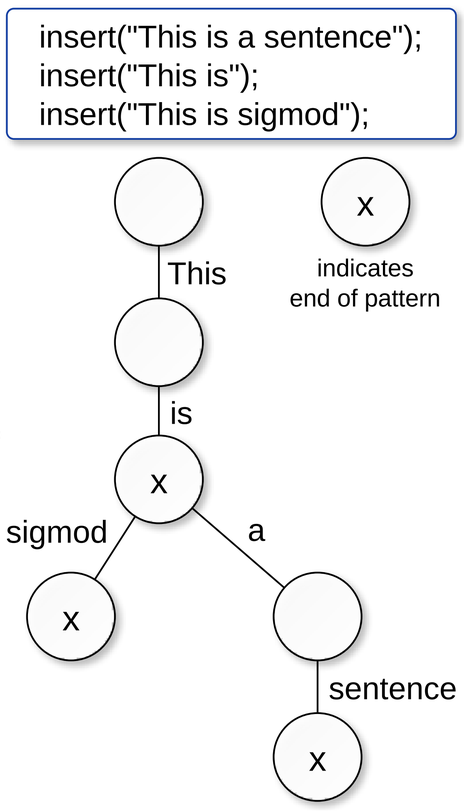
Example

D = {};
insert("This is a sentence");
contains("This is sigmod") -> false
insert("sigmod");
T = "This is sigmod 2017 and This is a sentence"
find(D,T) -> "sigmod", "This is a sentence"

Main Ideas

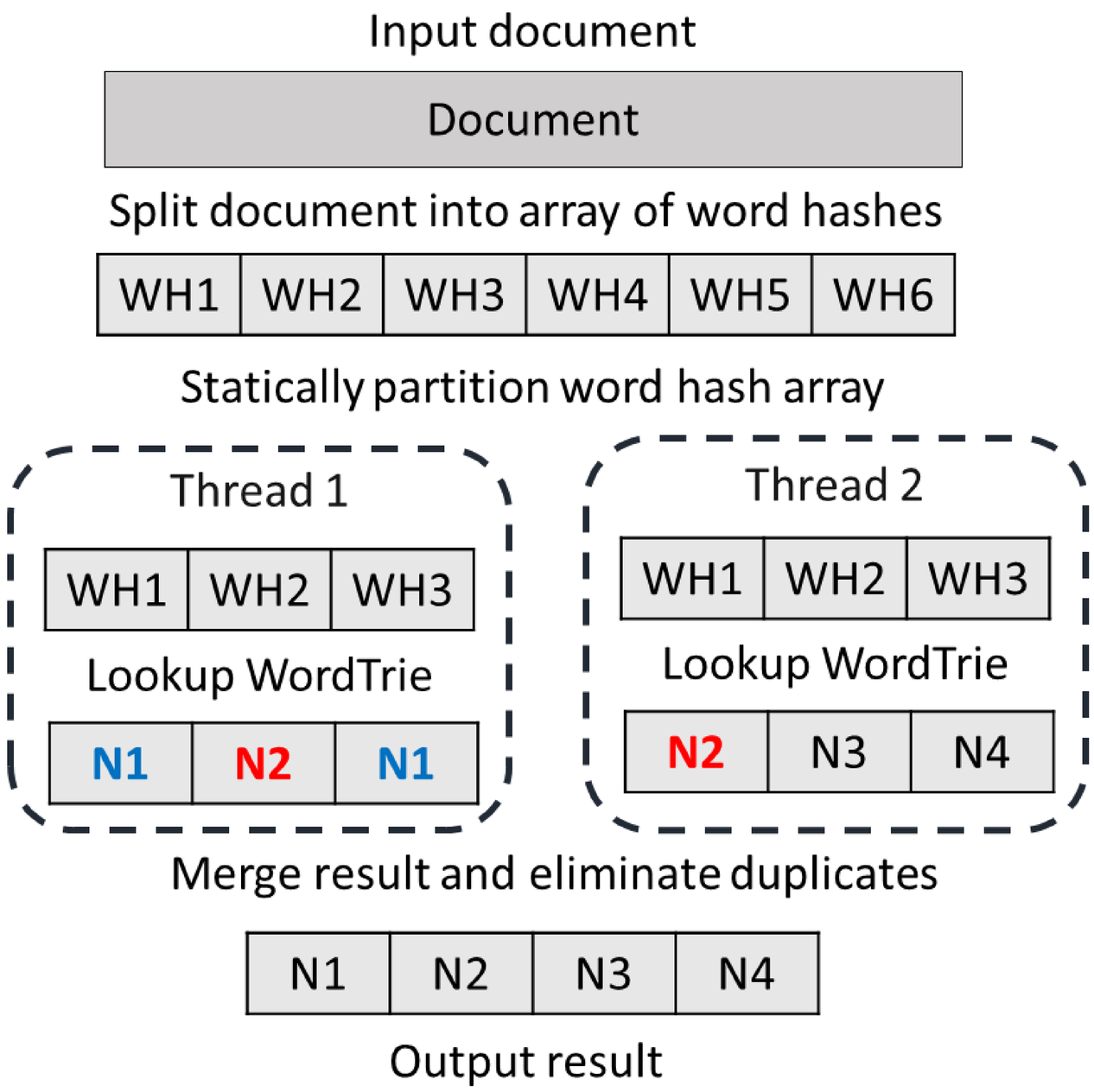
WordTrie

- Compact storage
- No redundant prefixes
- Word granularity
- Fast early exit



Document

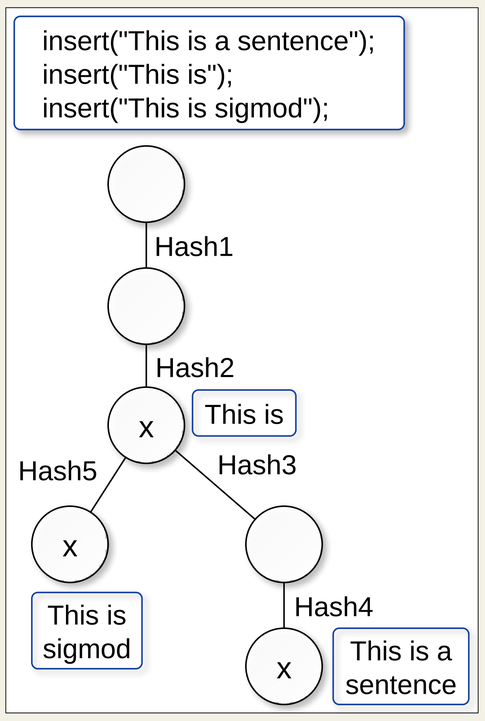
- For each document word, WordTrie returns all n-grams starting with this word
- Remove duplicates using Set
- Order is preserved with static partitioning



Optimizations

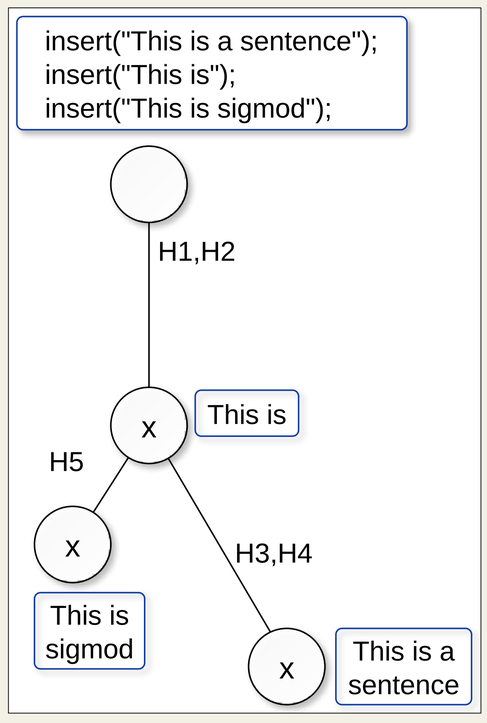
Hashing

- Replace words by hash
- Faster comparisons, smaller memory footprint
- Hash each word only once, reuse in hashmaps



Path Compression & Lazy Expansion

- Collapse chains in tree
- Reduced tree height
- Reduced memory footprint
- Reduced indirections



More: parallelization, custom memory allocator, … ask us!

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

ACM SIGMOD Programming Contest 2017