LASH: Large-Scale Sequence Mining with Hierarchies

Kaustubh Beedkar and **Rainer Gemulla**
Data and Web Science Group
University of Mannheim, Germany
{kbeedkar,rgemulla}@uni-mannheim.de

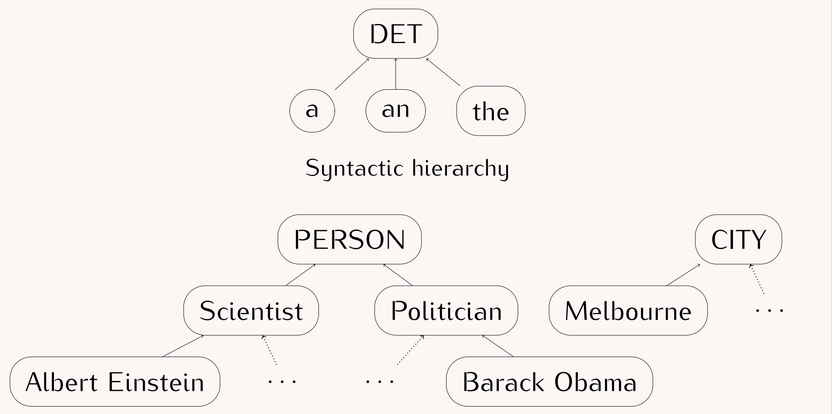
Sequence Mining

- Goal: Discover subsequences as patterns in sequence data
- Input: Collection of sequences of **items**, e.g.,
  → Text collection (sequence of words)
  → Customer transactions (sequence of products)
- Output: Subsequences that
  → occur in $\sigma > 0$ input sequences (frequency threshold)
  → have length at most $\lambda > 0$ (length threshold)
  → have gap $\gamma \geq 0$ (contiguous or non-contiguous subsequences)

Example:
S_1: Anna lives in Melbourne
S_2: Bob lives in the city of Berlin
S_3: Charlie likes London
lives in
$(\sigma = 2, \gamma = 0, \lambda = 2)$

Hierarchies

Items can be naturally arranged in a hierarchy:



Sequence Mining with Hierarchies

- Item hierarchies are specifically taken into account
- Items in output sequences may belong to different levels in the hierarchy

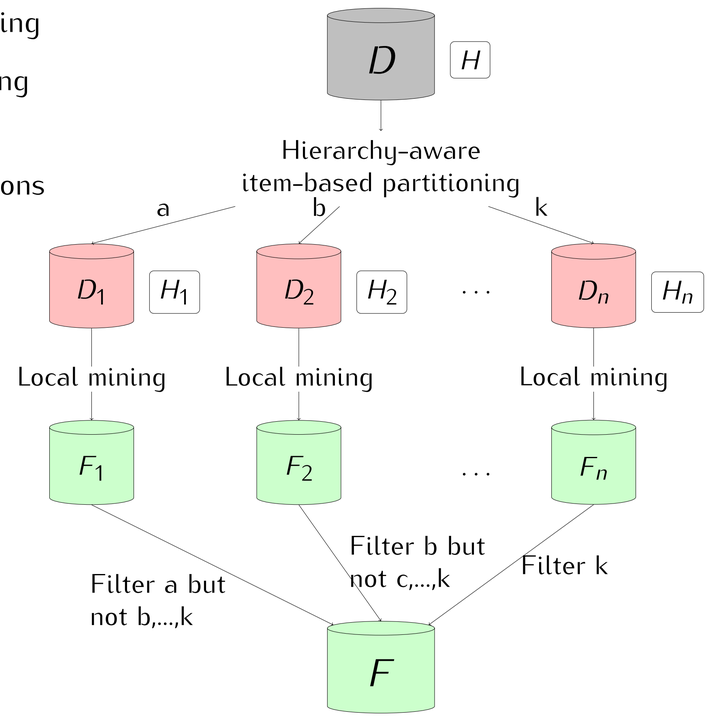
Example:
S_1: Anna lives in Melbourne
S_2: Bob lives in the city of Berlin
S_3: Charlie likes London
PERSON lives in CITY
$(\sigma = 2, \gamma = 3, \lambda = 4)$

- Applications:
  → Linguistic patterns: read **DET** book
  → Information extraction: **PERSON** lives in **CITY**
  → Market-basket analysis: buy **DSLR Camera** → **Photography book** → **flash**
  → Web-usage mining
  → ...

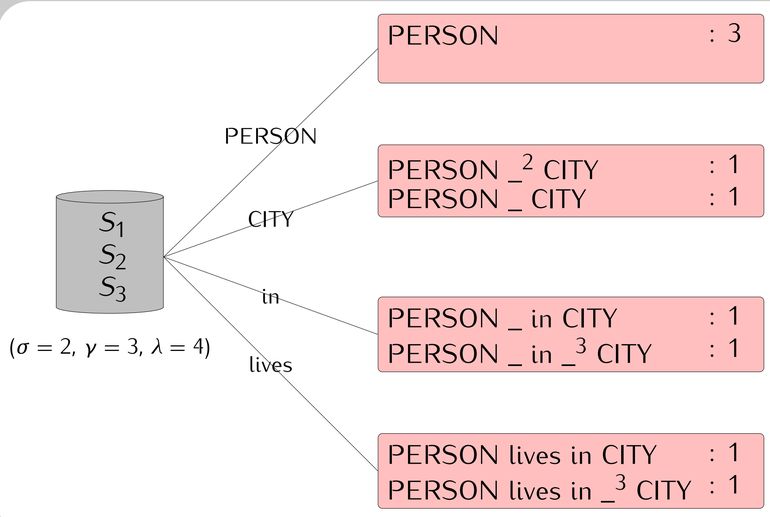
LASH

- Distributed framework for generalized sequence mining
- Build over MapReduce for large-scale data processing
- MAP (partitioning)
  → Data is divided into potentially overlapping partitions
- REDUCE (mining)
  → Partitions are mined independently

Key features
- Scales to very large datasets
- Novel hierarchy aware form of item-based partitioning
- Optimized partition construction
- Customized local mining
- No global post processing



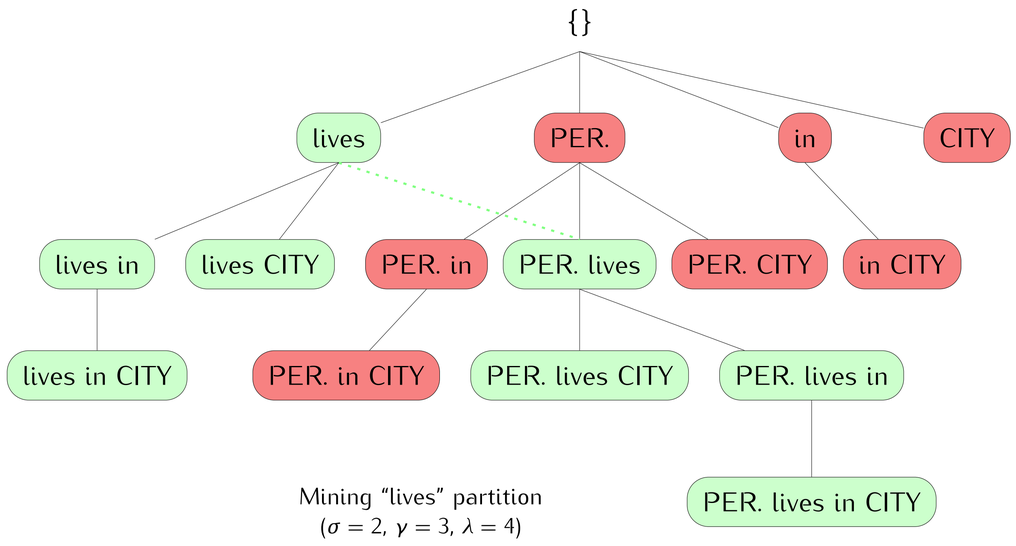
Partitioning



- Key idea: partition the output space
- Items are ordered by decreasing frequency
  e.g. PERSON < CITY < in < lives < ...
- Create a partition for each frequent item called pivot item
- Rewrite each input sequence for each partition
  → Fast rewrites (low overhead)
  → Makes partitions as small as possible
  → Reduces communication and skew

Mining

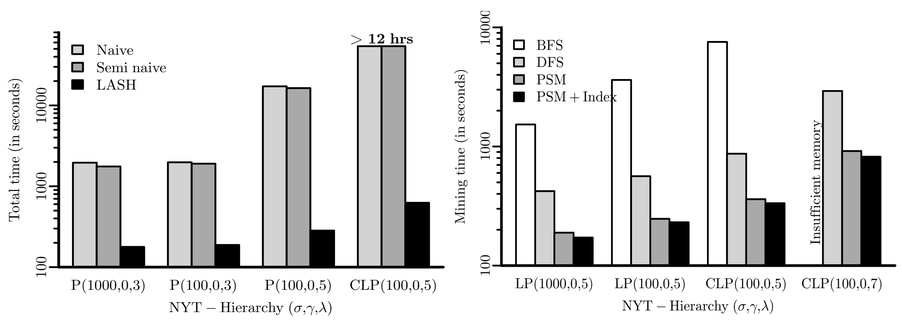
- Traditional approach
  → Mine using any GSM alg.
  → Filter non-pivot sequences
  → Inefficient
- LASH's PSM approach
  → Only mine pivot sequences
  → Requires no filtering



Mining "lives" partition
$(\sigma = 2, \gamma = 3, \lambda = 4)$

Experiments

The New York Times Corpus, syntactic hierarchy



Overall runtime

Local Mining

Highlights
- Multiple orders of magnitude faster
- PSM more than 3× faster than traditional sequence miners
- Good strong and weak scalability

UNIVERSITY OF MANNHEIM

http://uma-pi1.github.io/lash/