hPathSim: User-Guided Meta-Path Framework in Heterogeneous Networks

Hilfi Alkaff (alkaff2)
Efe Karakus (karakus1)

Outline

- Problem
- Related Work
  - PathSim
  - OLAP operations: roll-up, drill-down, slice
- Dataset
  - Network Statistics
- Contribution
  - Better Storage
  - Handling Hierarchies
  - User-specified constraints
- Results
  - hPathSim run-time and memory
  - OLAP operations run-time and memory
- Live Demo
- Future Work
- Discussion

Problem

- Similarity search using PathSim on large networks is expensive.
- We can not apply similarity search under specific context. (e.g. applying an OLAP operation first)
- PathSim does not allow for user-constrained search. Hard to find meaning by looking at meta-path.

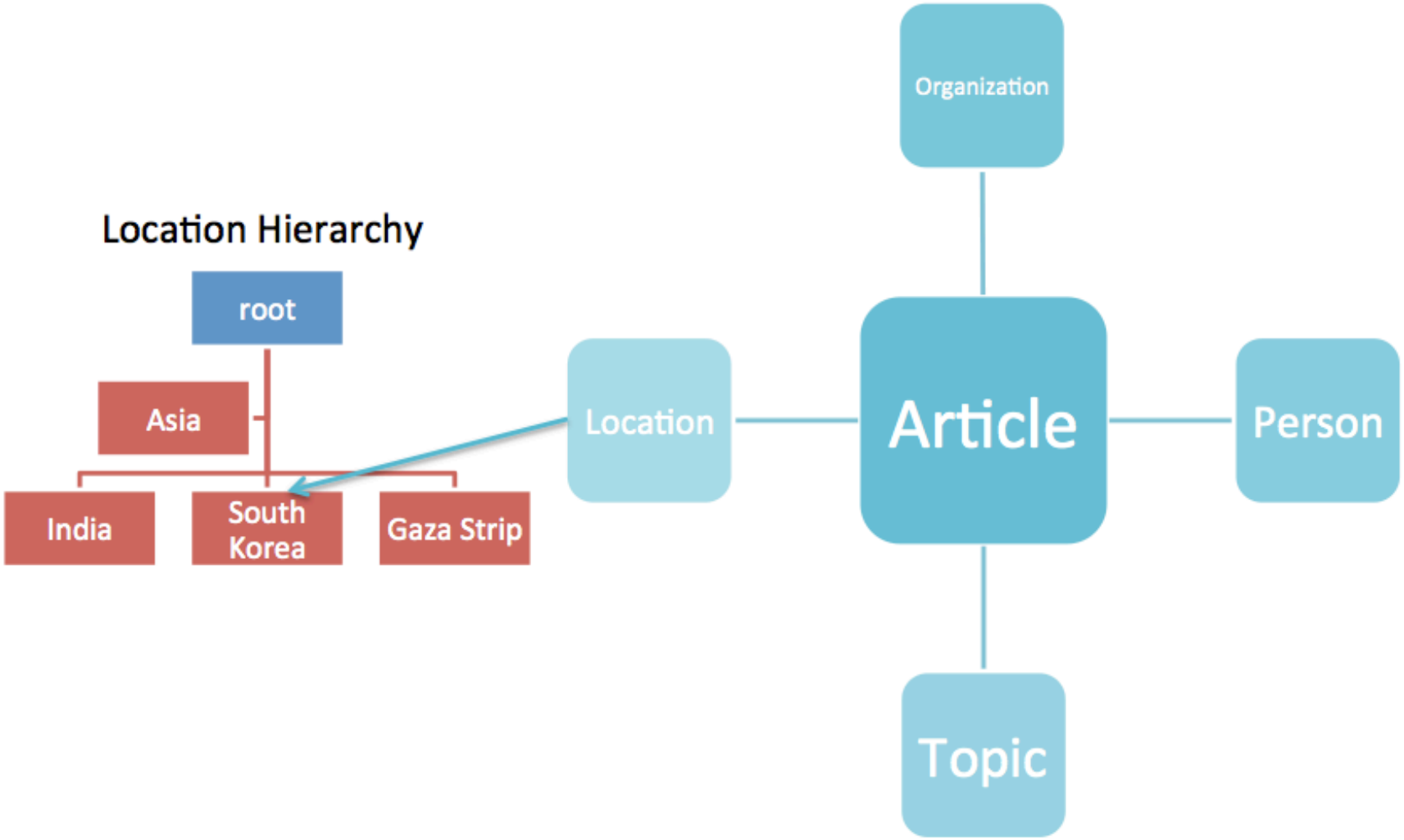
Related Work: PathSim

- Computing similarity in heterogeneous networks.
- Favor “peers”- objects with strong connectivity and similar visibility under a meta-path.
- Example meta-paths: author-paper-author (co-authors), author-paper-venue-paper-author (submitted to same conference).

Related Work: OLAP Operations

- Roll-up: Summarization of data along a dimension.
- Drill-down: Navigate to more detailed levels of the dimension.
- Slice: Pick a subset of the data cube.

Dataset: New York Times



Network Statistics

- DBLB
  - 70,536 vertices
  - Average degree 9.0, std: 74.8
  - Average path length: 4.0, std: 1.1
- NYT
  - 181,865 vertices
  - Average degree 10.0, std: 73.7
  - Average path length: 4.5, std: 0.97

Contribution: Better Storage

- Caches the similarity meta-paths between vertices (v1, v2).
- Every vertex is indexed by a unique id, allowing fast look-up.
- Vertices only store meta-information, content of articles are fetched later on the disk.

Contribution: Handling Hierarchies (1)

- Build a forest of hierarchical trees
  - –Hashtable to allow fast lookup of a category in the tree.
  - –Iterative DFS for roll-up and drill-down operations on the tree.
- 5 operations, *N* is number of nodes in tree:
  - –is_member() O(N)
  - –get_categories() O(N)
  - –get_parent() O(1)
  - - is_slice() O(1)
  - - get_children() O(1)

Contribution: Handling Hierarchies (2)

- Drill-down(category_name, val)
  - For each node, check if its category is a member of category_name
  - If it is not, delete node from current graph
- Roll-up(name, val)
  - Keep track of all leaves in hierarchies and the vertices which have that category
  - Add these vertices back to the current graph

Contribution: User-Specified Constraints

- Two types of constraints: must-link and cannot link
- User could specify these constraints at runtime
- Modified PathSim to handle constraints
  - For must-link: Enforced every time a complete path is added to the set of meta-paths
  - For cannot-link: Enforced every time a partial meta-path is about to be explored

Results: Similarity Search

Measurements were done between nodes that meta-paths of length 4, ran 5 times

Graph	Average (s)	Std
DBLP Full	5.26	0.042
NYT Full	5.12	0.157
DBLP DB	1.41	0.03
NYT Jordan	1.70	0.04

Results: OLAP Operations

Graph	Average (s)	Std
DBLP Drill-Down	0.055	0.024
DBLP Roll-Up	0.0058	0.00093
NYT Drill-Down	0.52	0.06
NYT Roll-Up	19.22	0.91

It should be noted that the drill-down and roll-up operations were done on different size subgraphs.

The NYT subgraph was 8x bigger than DBLP subgraph.

Live Demo

Future Work

- Adding more operations such as RankClus on network.
- Better performing roll-up operation.
- Handling higher scalability.
- Providing a better interface.