Efficient Taxonomic Similarity Joins with Adaptive Overlap Constraint

Pengfei Xu and Jiaheng Lu
Department of Computer Science, University of Helsinki, Finland

ABSTRACT
Established similarity join approaches usually deal with synthetic differences like typos and abbreviations, but neglect the semantic relations between words. Such relations, however, are helpful for obtaining high-quality joining results. In this paper, we leverage the taxonomy knowledge (i.e., a set of IS-A hierarchical relations) to define a similarity measure which finds semantic-similar records from two datasets. Based on this measure, we develop a similarity join algorithm with a prefix filtering framework to prune away irrelevant pairs effectively. Our technical contribution here is an algorithm that judiciously selects critical parameters in a prefix filter to maximise its filtering power, supported by an estimation technique and Monte Carlo simulation process.

SIMILARITY MEASURE
Let $S = \{s_1, \ldots, s_s\}$ and $T = \{t_1, \ldots, t_t\}$ be two sets of nodes from a hierarchical taxonomy. Let $s \in S$ and $t \in T$ be two nodes.

- **Similarity between two nodes** is defined based on their lowest common ancestor (LCA):
  \[
  TS(s, t) = \frac{|\text{LCA}(s, t)|}{\max(|s|, |t|)}
  \]
- **Based on node similarity**, set similarity aggregates all distinct node similarities:
  \[
  GTS(S, T) = \frac{W(S, T)}{\max(|S|, |T|)} = \max_{s \in S} \sum_{t \in T} f_{pq} TS(s_p, t_q)
  \]
  where $S$ and $T$ contain $p$ and $q$ nodes, $f_{pq}$ is an indicator variable (i) controlling whether to select the edge $(s_p, t_q)$, and (ii) ensures any of $s_p$ or $t_q$ is used at most once.

Solving for the value of $W$ in $GTS$ is to find the maximum weight matching in a bipartite graph. This can be done in polynomial time using Hungarian algorithm [1].

Example
Take two strings in Figure 1 as an example. Since the three most-similar node pairs are ("coffeehouse", "bar"), ("latte", "espresso"), and ("Turin", "Via Nizza"), the $GTS$ similarity between two strings becomes $0.717$ ($= (0.75 + 0.8 + 0.6) / 3 = 0.717$). Note that the distinctness forbid any node from being selected more than once, e.g., selecting both ("latte", "espresso") and ("latte", "Turin") are not allowed.

ACKNOWLEDGEMENT
This work is supported by the Academy of Finland (310321). Contact author and email: jiaheng.lu@helsinki.fi.

REFERENCES