ProbKB: Managing Web-Scale Knowledge

Yang Chen§, Daisy Zhe Wang§
§CISE, University of Florida
{yang, daisyw}@cise.ufl.edu

Abstract

- We designed a relational model for MLN and pushed all extracted facts, and rules into the database. This allows grounding algorithms that apply rules in batches. We implemented this model on Greenplum, a massive parallel processing (MPP) framework.
- We identified six rules pattern in SHERLOCK dataset. Each rule type has a table M_i recording the predicates involved in the rules of that type.
- We have another table R for relationships. For each relationship p(x, y) that is stated in the text corpus, we have a tuple (p, x, y) in R.

Grounding

- The relation model allows us to apply rules in batches using existing database techniques.
- Assume rules of type 3 are stored in table M3 (p, q, r), and relationships p(x, y) are stored in R(p, x, y), then the following SQL query computes atoms that are activated during the grounding process, this process is repeated until convergence, resulting in an active closure. The following SQL query then computes active clauses given the active atoms:

  ```
  SELECT DISTINCT R1.id AS id1, R2.id AS id2, R3.id AS id3
  FROM M3 JOIN R ON M3.p = R.p
  JOIN R R1 ON M3.q = R1.p
  JOIN R R2 ON M3.r = R2.p
  WHERE R.x = R1.x AND R.y = R2.x AND R1.y = R2.y
  ```

- The result of grounding is a factor graph (Markov network). This graph encodes a probability distribution over its variable nodes, which can be used to answer user queries. We use TUFFY as comparison point. We tried to run ProbKB using the REVERB-SHERLOCK dataset and measured the grounding time in 85 seconds while TUFFY crashes before grounding.

Knowledge Integrity

- We designed a novel robust semi-naive evaluation algorithm to improve accuracy and efficiency. The basic idea is to promote most confident facts and avoid repeated rule applications by maintaining a delta relation between two iterations.

Algorithm 1 Robust Semi-Naive Evaluation

```
repeat
  upvotes ← infer(beliefs, delta)
  beliefs ← beliefs ∪ delta
  delta ← promote(upvotes, candidates)
until delta = ∅
```

Results

- The results show that ProbKB outperforms state-of-the-art.
- Learned 100,000 more facts using OpenIE extractions in 85 sec.
- Semantic constraints greatly improve quality.