ParisMatch
Ontology Matching at Scale

Antoine Amarilli  Pierre Senellart  Fabian M. Suchanek

Télécom ParisTech
Performance

- Millions of entities, tens of millions of facts.
- In-memory, multithreaded computation.

<table>
<thead>
<tr>
<th>BerkeleyDB, SSD</th>
<th>RAM, 1 thread</th>
<th>RAM, 4 threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>20h30</td>
<td>1h15</td>
<td>26m</td>
</tr>
</tbody>
</table>

Table: Running times for the DBpedia–YAGO alignment task. Intel Core i7-3820 CPU clocked at 3.60 Ghz, 48 GB of RAM.
Join Relations

- Simplest possible difference in structure between ontologies.
- Relations of one ontology correspond to join relations in the other ontology.
- Similar to the “join” operator of relational calculus.
Literal Similarity Functions

- The original PARIS uses an exact literal equality function.
- Ad-hoc refinements: adjust for case, strip special characters...
- More general: index the literals using shingling:
  - hash the shingle sets through random hash functions
  - keep the minimum values (MinHashing)
- Use the index to compute pairs of similar literals efficiently!