Answering a 1000$ question...
John Turturro played this title screenwriter in a 1991 Coen Brothers film.

Two worlds of querying:

Structured
- + Better captures user intention and unambiguous
- Need expertise on language and schema

Unstructured
- + Easy to formulate (no joins)
- Misses user intention

Wiki-LOD Benchmark collection

<table>
<thead>
<tr>
<th>Property</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Documents</td>
<td>3,164,042</td>
</tr>
<tr>
<td>Wikipedia Category Articles</td>
<td>266,134</td>
</tr>
<tr>
<td>Wikipedia Entity Articles</td>
<td>2,053,050</td>
</tr>
<tr>
<td>Wikipedia Entity Articles with Infoboxes</td>
<td>907,304</td>
</tr>
<tr>
<td>Other Wikipedia Articles</td>
<td>844,857</td>
</tr>
<tr>
<td>Resolved DBpedia Links</td>
<td>36,941,795</td>
</tr>
<tr>
<td>Resolved YAGO2 Links</td>
<td>32,941,667</td>
</tr>
<tr>
<td>Intra-Wiki Links</td>
<td>22,235,754</td>
</tr>
<tr>
<td>External Web links</td>
<td>7,214,827</td>
</tr>
<tr>
<td>Imported DBpedia Properties</td>
<td>168,374,863</td>
</tr>
<tr>
<td>Imported YAGO2 Properties</td>
<td>23,634,511</td>
</tr>
</tbody>
</table>

Wiki-LOD Benchmark collection[1]

Query Benchmark[1]

90 handcrafted SPARQL queries with fulltext conditions:
- 50 jeopardy queries having one target entity as answer
- 40 natural language queries returning ranked list of one or more entities

Example query: John Turturro played this title screenwriter in a 1991 Coen Brothers film

select ?s where {
  FILTER FTContains(?)coen brethrens). 
  FILTER FTContains(?)John Turturro played 1991 Coen brothers film
}

Setup and Evaluation

- Manually translated 90 queries into SPARQL with fulltext conditions.
- Built a fulltext index (using BM25 ranking model) over the textual components of Wiki-LOD documents.
- Assessed 50 of the Jeopdary queries using Amazon Mechanical Turk

Novelties & Goals

- Benchmark collection and queries help to bring together two worlds (structured & unstructured) retrieval models and challenge new types of indexing and querying.
- Linked Data pointers (Yago and DBpedia) in Wiki-LOD allow people to extend into almost arbitrary RDF (LOD) collections.
- The new collection may be of high interest for other IR applications, like the clustering, classification of entities, summarization, taxonomy building, and many more.

References


A Sample “Albert Einstein” Wiki-LOD XML Document

Unify( DBpedia + YAGO, WIKIPEDIA )

FTContains operator for SPARQL
- to add keyword constraints to structured query