Structured Querying of Web Text Data

Ni Lao, Le Zhao
2007.11.5
Web Scale IE

• IE has become unsupervised, domain-independent, and scalable
  – DIRT(01)
    • Given a predicate
      – $X$ manufactures $Y$
    • Automatically extract its synonyms
      – $X$ produces $Y$; $X$ markets $Y$; $X$ develops $Y$; $X$ is supplier of $Y$; $X$ ships $Y$; etc.
  – KNOWITALL(05)
    • Given a set of universal patterns for extraction
      – NP “and other” <class1>
      – NP “is a” <class1>
    • Given a set of predicates
      – “scientist”, “invented”
    • Automatically extract facts of these predicates
      – scientist(Einstein), invented(Edison, light bulb)
  – TEXTRUNNER(07)
    • Extract all facts in one pass of the corpus,
    • without any kind of human input

• Trend
  – No human labeling
  – No predefined schema
Structured Access to The Web

• What is the opportunity?
• Observation
  – Some information need can be better fulfilled by structured query
    • List output is preferred
    • Constrained by some semantics
    • Need indication of popularity for each answer
  – “list all countries that have donated money to the Gujarati earth quake, how much they donated, and when”

• The semantic web
  – A vision of information that is understandable by computers, so that they can perform more of the tedious work involved in finding, sharing and combining information on the web [wikipedia]
    • “list the prices of flat screen HDTVs larger than 40 inches with 1080p resolution at shops in the nearest town that are open until 8pm on Tuesday evenings”
  – (tried but with no success yet)
    to provides a standard (like RDF) for websites to publish information

• The OIE paradigm
  – instead of publishing standard
  – Achieve semantic web by unsupervised extraction and Structured Access
Contributions (of This Work)

• A new paradigm of structured access to the web
• A data model and query scheme
• Some preliminary experiment results
The Big Picture

• The dream of a DB people
  – The information need of users can be satisfied by a RDB
  – And the structural data can be extracted from the web
Web Data Model

- Base-level concepts (with probabilities)

<table>
<thead>
<tr>
<th>Concept</th>
<th>e.g.</th>
<th>Extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>facts</td>
<td>discovered(Edison, phonograph)</td>
<td>TextRunner [4]</td>
</tr>
<tr>
<td></td>
<td>sells(Amazon, PlayStation)</td>
<td></td>
</tr>
<tr>
<td>Semantic types (IS-A relation)</td>
<td>city(Boston)</td>
<td>KnowItAll [20]</td>
</tr>
<tr>
<td></td>
<td>electronics(dvd-player)</td>
<td></td>
</tr>
<tr>
<td>synonymy</td>
<td>invented(x, y) = has-invented(x, y)</td>
<td>DIRT [29]</td>
</tr>
<tr>
<td>tropeomy</td>
<td>invented(x, y) ⇒ discovered(x, y)</td>
<td>?</td>
</tr>
<tr>
<td>Functional Dependency (FD)</td>
<td>has-capital(x, y) ⇒ capital(y)</td>
<td>?</td>
</tr>
</tbody>
</table>

- Query Scheme
  - Use Select-Project-Join (SPJ) queries
    - SPJ is single Block SQL with no “Group By”
    - E.g. q(?x, ?y) :- died-in(<scientist> ?x, 1955 ?y)
  - Result is a synthetic table
Query Processing

• For non-projecting queries
  – A proximate top-k ranking algorithm similar to [Theobald, et al 2004]

• For projecting queries (need aggregation)
  – q(?s) :- invented(<scientist> ?s, ?i)
    • Probability of inventions need to be summed out for each scientist

  – Challenges
    • Performance: potentially large number of item to sum over
    • Large number of low-quality tuples boost a poor answer

  – Solution
    • A panel of Experts: sum only the top k tuples (k=5)
    • An expert is a tuple with a score
      – e.g. invented(Tesla, Fluorescent-Lighting),0.95
Experiment Result

• Results of two queries are compared
  – q(?s) :- invented(hscientisti ?s, ?x)
  – Goolge result of “scientist invented”
    • “scientist” is a misleading word. These people are usually call physicist, chemist archeologist etc.

• Should define concrete tasks for more objective evaluation
  – QA tasks
  – Information distillation tasks
  – ..
Alternative Models

- Three (structural access) models differ at how much work is done offline

<table>
<thead>
<tr>
<th></th>
<th>Extraction</th>
<th>Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Extraction Model</td>
<td>offline</td>
<td>offline</td>
</tr>
<tr>
<td>ExDB</td>
<td>offline</td>
<td>online</td>
</tr>
<tr>
<td>Text Query Model</td>
<td>online</td>
<td>online</td>
</tr>
</tbody>
</table>

- Web pages
- Extraction
- Facts
- City (Boston)
- Electronics (DVD-player)
- Integration

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kepler</td>
<td>log books</td>
<td>1630</td>
<td>0.7902</td>
</tr>
<tr>
<td>Heisenberg</td>
<td>matrix mechanics</td>
<td>1976</td>
<td>0.7897</td>
</tr>
<tr>
<td>Galileo</td>
<td>telescope</td>
<td>1642</td>
<td>0.7395</td>
</tr>
<tr>
<td>Newton</td>
<td>calculus</td>
<td>1727</td>
<td>0.7366</td>
</tr>
</tbody>
</table>
Schema Extraction Model

• IE system extract only one type of information
  – object-attribute-value (e.g. Edison, invention, phonograph)

• Try to derive a single best schema for the whole web by optimizing
  – completeness (all extractions from text appear in the output)
  – simplicity (the output has few tables),
  – fullness (the output database has no NULLs)

• Pros
  – No need to write SQL query!
  – For the user who are trying to make sense of a domain, the tables are already populated offline

• Cons
  – Not easy to optimize

• Solution
  – ?
Text Query Model

• No information extraction offline
• Instead Offers users a query language that does extraction online

```
SELECT bandCity, bandDate
FROM ("http://thebandilike.com/++",
    ["to appear in <string> on <date>",
    bandCity, bandDate])
WHERE
    bandDate > 2006 AND
    geographicdist(bandCity, "Seattle") <= 100
```

• Pros:
  – Flexibility of expressing information need
• Cons:
  – query time performance
• Solution:
  – text indexing techniques
  – e.g. neighbor index, multi-gram index [8, 11]
Trends

- The Pace of Web Scale IE Is Fast
- Going Beyond Keywords
  - Benefit: reduced the representation gap

- Going Web Scale
  - Need light weight methods

- Going Open Domain & Unsupervised
  - Benefit: scalability
  - Challenge: uncertainty at the schema level

- Going Probabilistic
  - Markov Networks
• THE END
• THANKS
Challenges

• Ambiguity
  – “Java”, “John Smith”, “develop”