Advanced IR Seminar 2007, LTI

Structured Querying of Web Text Data

Ni Lao, Le Zhao

2007.11.5

Web Scale IE

- IE has becomes unsupervised, domain-independent, and scalable
  - DIRT(01)
    - Given a predicate
      - *X* manufactures *Y*
    - Automatically extract its synomyns
      - 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

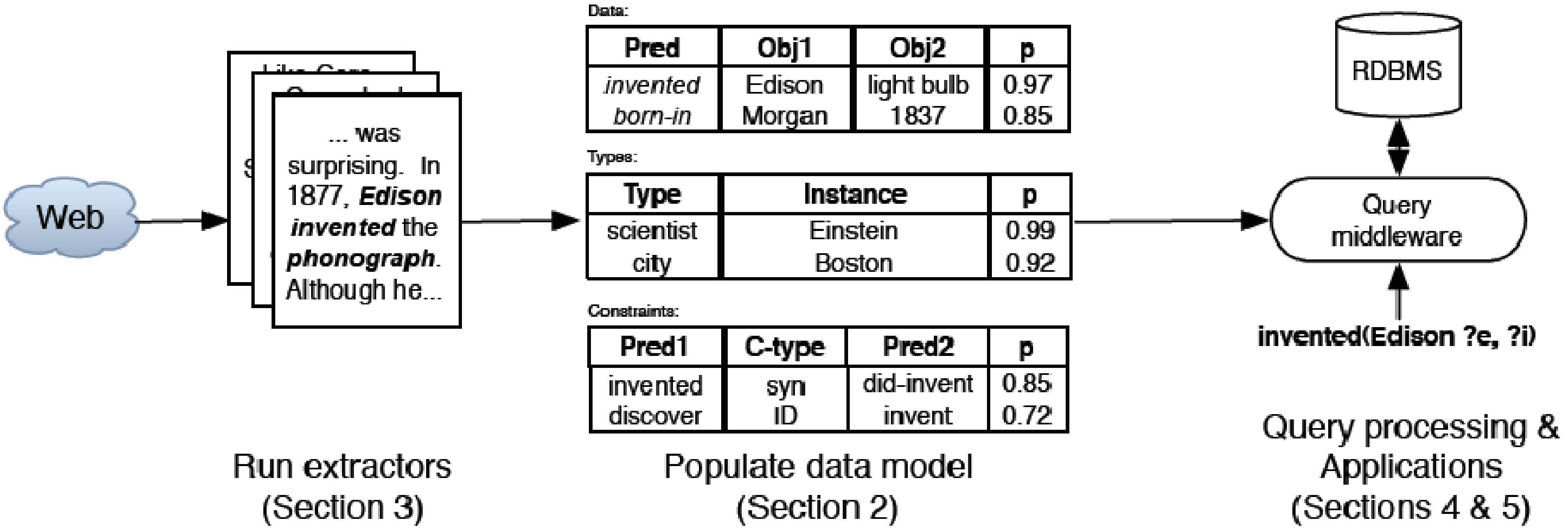
Structured Access to The Web

- 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



... was surprising. In 1877, ***Edison invented*** the ***phonograph***. Although he...

Data:

Pred	Obj1	Obj2	p
invented	Edison	light bulb	0.97
born-in	Morgan	1837	0.85

Types:

Type	Instance	p
scientist	Einstein	0.99
city	Boston	0.92

Constraints:

Pred1	C-type	Pred2	p
invented	syn	did-invent	0.85
discover	ID	invent	0.72

invented(Edison ?e, ?i)

Run extractors
(Section 3)

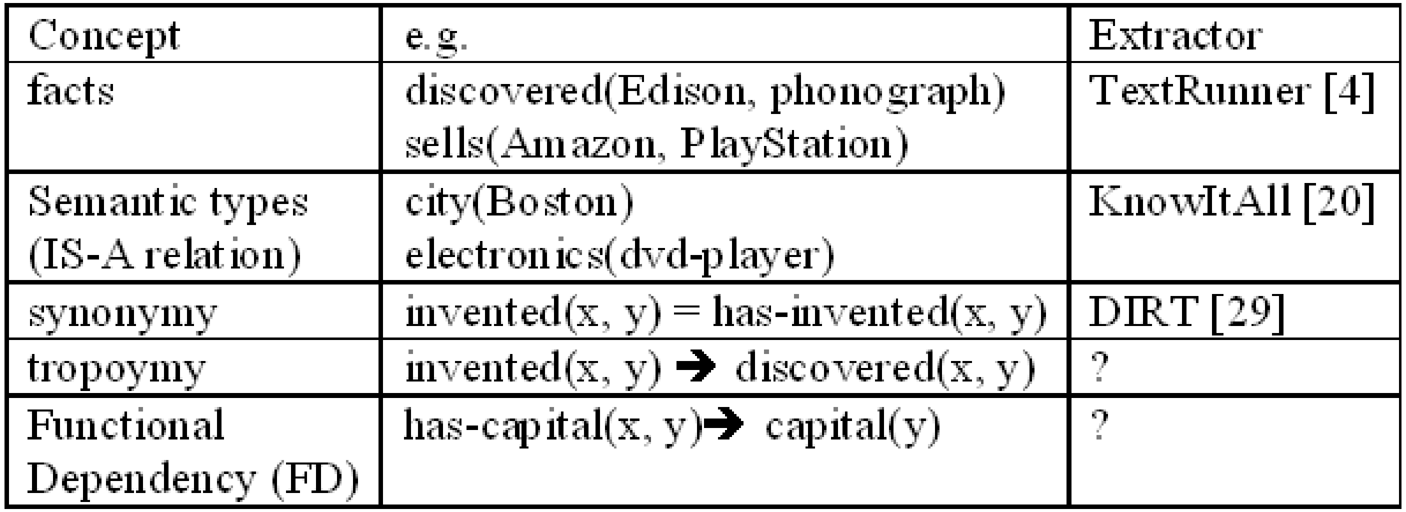
Populate data model
(Section 2)

Query processing &
Applications
(Sections 4 & 5)

- 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)



Concept	e.g.	Extractor

facts	discovered(Edison, phonograph)
	sells(Amazon, PlayStation)
		TextRunner [4]

Semantic types (IS-A relation)	city(Boston)
	electronics(dvd-player)
		KnowItAll [20]

synonymy	invented(x, y) = has-invented(x, y)	DIRT [29]
tropoymy	invented(x, y) ➔ discovered(x, y)	?
Functional Dependency (FD)	has-capital(x, y)➔ capital(y)	?

- 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 sumed 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

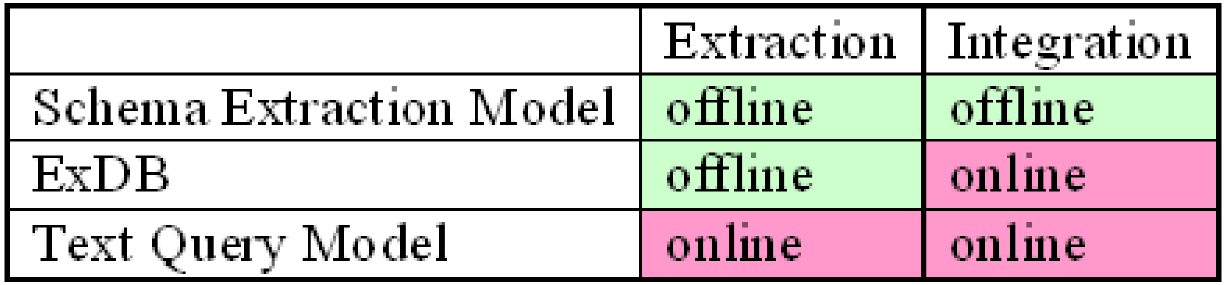
Experiment Result

- 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.

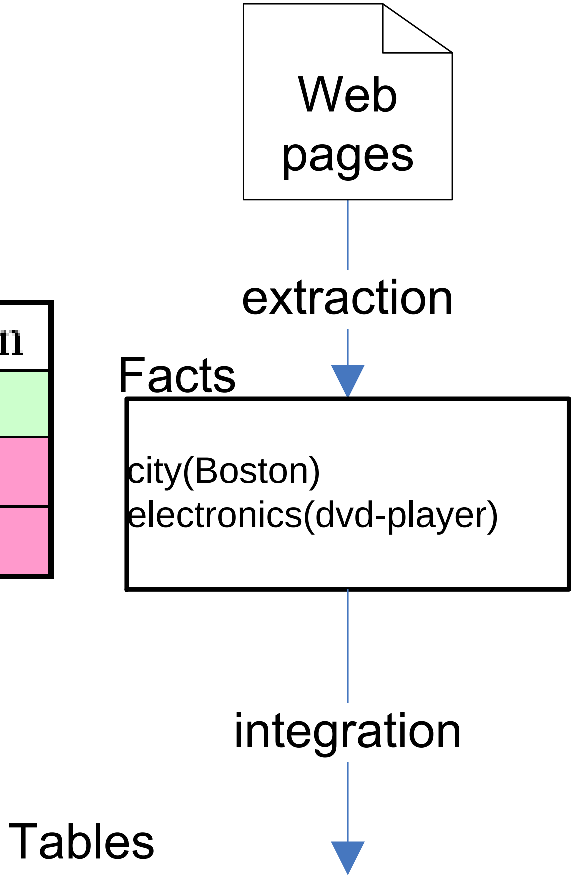
- Should define concrete tasks for more objective evaluation
  - QA tasks
  - Information distillation tasks
  - ..

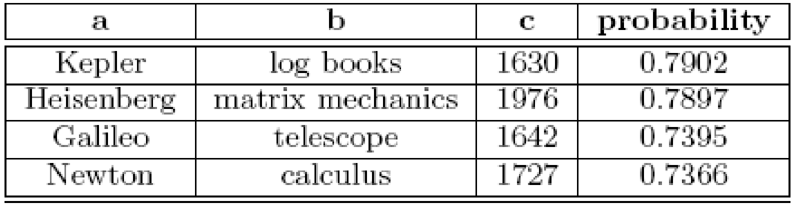
Alternative Models

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



	Extraction	Integration
Schema Extraction Model	offline	offline
ExDB	offline	online
Text Query Model	online	online





a	b	c	probability
Kepler	log books	1630	0.7902
Heisenberg	matrix mechanics	1976	0.7897
Galileo	telescope	1642	0.7395
Newton	calculus	1727	0.7366

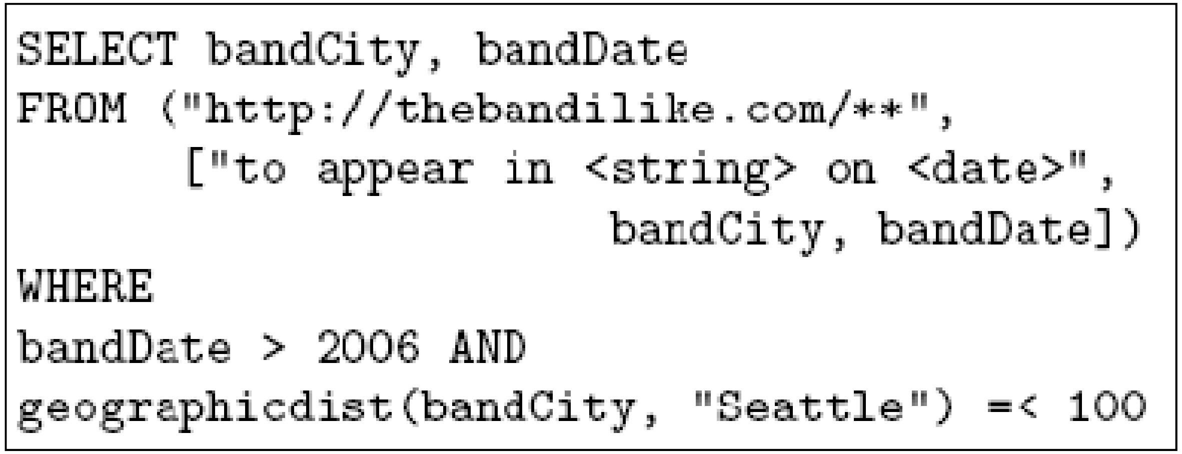
Schema Extraction Model

- 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: scalabity
  - Challenge: uncertainty at the schema level
- Going Probabilistic
  - Markov Networks

- THE END
- THANKS

Challenges

- Ambiguity
  - “Java”, “John Smith”, “develop”