Cross-Fertilizing Deep Web Analysis and Ontology Enrichment

Marilena Oita, Antoine Amarilli, and Pierre Senellart

August 31, 2012

Telecom ParisTech, France
The Deep Web

dynamically-generated Web pages in response to a user query

HTML forms: intuitive to humans, but hardly understandable by search crawlers

challenging research topic: there are (still) no practical ways for search engine crawlers to explore this rich source of data in a meaningful way;
The Deep Web

**Apps:**

1. focused indexing (vertical search engines)
2. extensional crawling (Web archiving)
3. Semantic Web (ontology enrichment)

**Motivation:**

- **IN:** deep Web sources are vast repositories of *semi-structured data*
- **IDEA:** leverage the *Structured Web* for the expansion of the *Semantic Web*
- **OUT:** access to the deep Web data in a *fully automatic, domain-independent* manner
Outline

1. Context
Outline

1. Context
2. Envisioned Approach
Outline

1. Context
2. Envisioned Approach
3. Advantages
Outline

1. Context
2. Envisioned Approach
3. Advantages
4. Conclusions
Form Interface Understanding

ordered list of form elements

- labels
- constraints
- set values, for non-textual input elements

Understanding...

1. how form elements relate to each other
   extract an input schema →
   - syntactic parsing (as a tree)
   - visual segmentation, etc.

2. which type of input values are valid (e.g., gazetteer)
Domain Knowledge Related Work (1)

→ works rely on a domain knowledge, constructed:

1 manually
2 using machine learning
3 by mapping schemas of different form interfaces (pertaining to the same domain, though)

Shortcomings:

- is highly simplifying the real Web situation, in which a global virtual schema of deep Web entities cannot exist
- approach not scalable
- is segmenting even more the Semantic Web
Information Extraction from Result Pages

**valid form submission**: Web records
Information Extraction *Related Work (2)*

→ works suppose valid response pages and extract the data values from records through **IE processing** *Aim:*

1. building/enriching ontologies or gazetteers
2. expanding sets of entities

**Shortcomings:** isolated works that do not involve the form understanding
Holistic Approach

Motivation:

- (complementarity): the form interface and the response pages represent facets of the same conceptual object
- (interconnection): the output of each step is useful for the next;
- (late ontologic use): a source of knowledge is inevitable – relax the domain specificity constraint by adapting to the data context;
Domain-Agnostic Form Probing

Purpose: \(\rightarrow bootstrap\) some initial response pages

- fill out a textual input with a stop word or a contextual term (possibly, use the AJAX auto-completion facilities)
- select or check non-textual input elements
Record Identification

1. **The Adventures of Tom Sawyer (Dover Thrift Editions)** by Mark Twain (Jan 27, 1998)
   - **Formats**
     - **Paperback**
       - Usually ships in 1 to 4 weeks
       - Eligible for FREE Super Saver Shipping and 1 more promotion
       - Price: $3.50
     - **Kindle Edition**
       - Auto-delivered wirelessly
       - Price: $2.97
   - Other Formats: Hardcover; Paperback; Mass Market Paperback; Audio CD; See All.
   - Sell this back for an Amazon.com Gift Card

2. **Life on the Mississippi** by Mark Twain (Nov 5, 2011)
   - **Formats**
     - **Paperback**
       - Order in the next 27 hours to get it by Wednesday, May 30.
       - Price: $13.99
     - **Kindle Edition**
       - Auto-delivered wirelessly
       - Price: $0.00
   - Other Formats: Hardcover; Paperback; Mass Market Paperback; Audio CD; See All.
   - Sell this back for an Amazon.com Gift Card
Record Identification

typically, wrapper induction techniques

→ **FOREST**: identify the location of records using the keywords used during form submission to identify their *common XPath* in the DOM
Attribute Alignment

Web records = structurally-similar DOM subtrees:

1. extract the values of textual leaf nodes
2. group values based on their record internal path

Example

//[div[class="data"]/h3[class="title"]/a[class ="title"]  {The Adventures of Tom Sawyer (Dover Thrift Editions); Life on the Mississippi}

//[div[class="data"]/span[class="ptBrand"]/a[href=...]  {Mark Twain}

//[div[class="data"]/span[class="bindingAndRelease"]  {Jan 27, 1998; 2011}
Attribute Alignment

\[ \text{record feature} = \langle \text{record internal path, cumulated bag of instances} \rangle \]

Used for:

1. constructing the **output schema** (\(\text{:=} \) the ordered sequence of record features)
2. generation of RDF triples
Input-Output Schema Mapping

align input fields of the form with record features of response pages
Input and Output Schema Mapping

Idea: the form as an instrument of validating mapping hypothesis:

1. use extracted values as query instances
2. verify the record internal path where they will appear in the responses

→ the same values will appear consistently in all the records, under its expected record internal path

///[div[class="data"]/span[class="ptBrand"]/a[href=...]
{Mark Twain}
The following results were found for your search:

- *Great Expectations*
- Charles Dickens
- Dover Thrift Editions

- *David Copperfield*
- by Charles Dickens
- Penguin Classics

### Labeled graph

```
?class
  rdfs:type
  ?e1
  "Great Expectations"
  "Charles Dickens"
  "Dover Thrift Editions"

?e2
  rdfs:type
  "David Copperfield"
  "by Charles Dickens"
  "Penguin Books"
```

RDF triples generation

### List of records

The following results were found for your search:

- *Great Expectations*
- Charles Dickens
- Dover Thrift Editions

- *David Copperfield*
- by Charles Dickens
- Penguin Classics
Labeled Graph Construction

1. entities \(:=\) records
2. all records are of the same rdf:type
3. literals \(:=\) extracted data values
4. for each record feature, attribute values are of the same rdf:type
5. the relation (i.e., predicate) \(:=\) record internal path
Deep Web Data Alignment

Yago

- **Othello**
  - y:hasName: "Othello"
  - y:created: "Shakespeare"
- **Great Expectations**
  - y:hasName: "Great Expectations"
  - y:created: "Charles Dickens"
- **David Copperfield**
  - y:hasName: "David Copperfield"
  - y:created: "Shakespeare"

Book

Ontology alignment

Labeled graph

- ?e1
  - rdfs:type: "Great Expectations"
  - "Charles Dickens"
  - "Dover Thrift Editions"
- ?e2
  - rdfs:type: "David Copperfield"
  - "by Charles Dickens"
  - "Penguin Books"
Deep Web Data Alignment

Components:

1. labeled graph
2. generic reference ontology: YAGO
3. alignment system: PARIS (VLDB ’12) aligns both entities and relations by:
   - matching literals
   - propagating evidence based on relation functionalities

Purpose obtain the missing:

- relations
- the class of entities (e.g., book)
- the meaning of record attributes (data type, domain and range)
Preliminary Experiments using PARIS

approach prototyped for the Amazon advanced search form for books

1. similarity computation: Hamlet (French Edition) $\equiv$ Hamlet
2. compute the transitive closure of the ontology graph – to answer reachability questions regarding relation mappings

→ in practice: limit the exploration depth to 2

William Shakespeare $\text{y:created}$ Hamlet
William Shakespeare $\text{y:hasPreferredName}$ Shakespeare
Alignment Consequences

1. Propagate discovered knowledge back to the input schema:
   - Discovered relations are mapped to the record internal paths of attributes.
   - Attribute types propagate to form input fields.

2. Incrementally infer new representative instances to fill in the form.
New Probing Terms

### Form

- **Author:**
- **Title:**
- **Publisher:**

### Yago

- **Othello**
  - `y:created` by **Shakespeare**
  - **y:hasName** "Othello"
- **Great Expectations**
  - **y:created** by **Charles Dickens**
  - **y:hasName** "Great Expectations"
- **David Copperfield** (novel)
  - **y:created** by **Shakespeare**
  - **y:hasName** "David Copperfield"
Ontology Enrichment

possibilities

1. set of entities expansion
2. add facts (triples) that are missing in YAGO attribute values
3. add the relation types that did not align
Ontology Enrichment

possibilities

1. set of entities expansion
2. add **facts** (triples) that are missing in YAGO attribute values
3. add the **relation** types that did not align → more challenging
Ontology Enrichment

Yago

Book
  \[ rdfs:type \]
  \[ y:hasName \]
  \[ y:created \]

Othello
  \[ rdfs:type \]
  \[ y:hasName \]
  \[ y:created \]

Great Expectations
  \[ rdfs:type \]
  \[ y:hasName \]
  \[ y:created \]

David Copperfield (novel)
  \[ rdfs:type \]
  \[ y:hasName \]
  \[ y:created \]

Yago

Shakespeare
  \[ y:hasName \]
  \[ y:created \]

Charles Dickens
  \[ y:hasName \]
  \[ y:created \]

Labeled graph

?e1

?e2

?class

"Great Expectations"
"Charles Dickens"
"Dover Thrift Editions"
"David Copperfield"
"by Charles Dickens"
"Penguin Books"
Holistic Approach

Form

Author: 
Title: 
Publisher: 
Submit

Result page

The following results were found for your search:

Great Expectations
Charles Dickens
Dover Thrift Editions

David Copperfield
by Charles Dickens
Penguin Classics

List of records

The following results were found for your search:

Great Expectations
Charles Dickens
Dover Thrift Editions

David Copperfield
by Charles Dickens
Penguin Classics

Yago

rdfs:type
Othello

y:hasName
Shakespeare
y:hasName
"Shakespeare"

rdfs:type
Great Expectations

y:hasName
"Great Expectations"
y:created
Charles Dickens
y:hasName
"Charles Dickens"

rdfs:type
David Copperfield (novel)

y:hasName
"David Copperfield"
y:created
Charles Dickens
y:hasName
"Charles Dickens"

Labeled graph

rdfs:type
Great Expectations

y:hasName
"Great Expectations"
y:created
Dover Thrift Editions
"Dover Thrift Editions"

rdfs:type
David Copperfield

y:hasName
"David Copperfield"
y:created
by Charles Dickens
"by Charles Dickens"

y:hasName
Penguin Books
"Penguin Books"

ontology alignment

ontology enrichment

new probing terms

input and output schema mapping

form probing

wrapper induction

RDF triples generation

ontology alignment

ontology enrichment
Conclusions

**advantages**

1. fully automatic
2. domain-independent
3. focused on knowledge discovery

**further experiments:**

1. more sophisticated strategy for the I/O schema matching
2. test forms from various domains (YAGO coverage)
3. multiple settings for PARIS (e.g., vary the exploration depth)
Challenges

- identification of new relation types of interest among those extracted
- domain identification (through form object description)
- resilience to outliers and noise resulting from imperfect literal matching
- proper management of the confidence in the results of each automatic task (cascade behavior)
Thank You

Questions