An Analysis of Duplicates in Web Extracted Objects

(Position Paper)

Stefano Ortona
Department of Computer Science, Oxford University
April 7th, 2014 @ DEOS 2014
Scenario

Web records
Web wrappers

(Semi) Supervised
(require human help)

* Lixto, WebHarvest, diffboat.com

Unsupervised
(do not require human help)

* DIADEM, Dalvi et al. (VLDB 2011), Crescenzi et al. (WWW’13)

In order to build a comprehensive database of entities of interest, one might have to extract from **thousands of sites**

(An analysis of structured data on the web, Dalvi et al. (Yahoo) VLDB 2012)
## From unstructured to structured Data

<table>
<thead>
<tr>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
<th>Bedroom</th>
<th>Bathroom</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>£104.95</td>
<td>Pretoria Road, Bordesley Green, Birmingham, B9 5LG</td>
<td>Houses</td>
<td>NULL</td>
<td>Pretoria Road, Bordesley Green, Birmingham, B9</td>
<td>2</td>
<td>1</td>
<td>NULL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
<th>Bedroom</th>
<th>Bathroom</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>£104.95</td>
<td>Pretoria Road, Bordesley Green B9</td>
<td>2 bedroom</td>
<td>for Sale</td>
<td>Pretoria Road, Bordesley Green B9</td>
<td>NULL</td>
<td>NULL</td>
<td>Pretoria Road, Bordesley Green B9</td>
</tr>
</tbody>
</table>
Current Status

Current solutions do not work

- Rule-based approaches do not capture the whole variety of the Web (and do not scale)
- Machine Learning techniques are inappropriate:
  - hard to engineer stable features
  - expensive to train and maintain models
  - hard to construct training data
- Classic record matching algorithms rely on the correctness of the Data
## The real scenario

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Description</th>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>Detached House</td>
<td>Offers in the region of £349,950</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham</td>
<td>NULL</td>
<td>NULL</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham</td>
</tr>
<tr>
<td>Barnt Green Road, Cofton Hackett, Birmingham B45</td>
<td>Added on 4th Oct 2013</td>
<td>£349.95</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham B45</td>
<td>4 bedroom detached new house for sale</td>
<td>for Sale</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**Notes:**
- **Noise:** NULL values in the table.
- **Null Values:** Misplaced attribute values in the table.
- **MISPLACED ATTRIBUTE VALUES:**
The Idea

Try to repair the broken records

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Description</th>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>Detached House</td>
<td>Offers in the region of £349,950</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham</td>
<td>Detached house</td>
<td>NULL</td>
<td>B451AG</td>
</tr>
<tr>
<td>B451AG</td>
<td>Detached House</td>
<td>£349,950</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham</td>
<td>Detached house</td>
<td>for Sale</td>
<td>Birmingham</td>
</tr>
</tbody>
</table>

The table above shows two property listings.

- The first listing has a NULL City value and is marked for repair as the Location details are incomplete.
- The second listing has a complete City value and is marked as sold.

The task is to repair the broken records and ensure all necessary information is filled in accurately.
Understand records structure

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Description</th>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>Detached House</td>
<td>Offers in the region of £349,950</td>
<td>Barnt Green Road, Cofton Hackett, Birmingham</td>
<td>Detached house for Sale</td>
<td>NULL</td>
<td>B451AG</td>
</tr>
<tr>
<td>NULL</td>
<td>Studio flat</td>
<td>Offers in excess of £200,000</td>
<td>47 St. Clement's, Oxford</td>
<td>Studio for Rent</td>
<td>NULL</td>
<td>OX41AG</td>
</tr>
<tr>
<td>NULL</td>
<td>Apartment</td>
<td>£130,000</td>
<td>Euston Square, London</td>
<td>2 Bedroom House for Sale</td>
<td>NULL</td>
<td>NW45</td>
</tr>
</tbody>
</table>

The Approach ➔ Part 1
The Oracle

A combination of two entity extraction systems

General entity extraction system
(Person, Organization, City) +
Manually engineered entity recognisers
(postcode, property_type, address)

Re-align the records

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Description</th>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>String</td>
<td>String</td>
<td>Street</td>
<td>County</td>
<td>Type</td>
<td>Status</td>
</tr>
</tbody>
</table>
Re-align the records

<table>
<thead>
<tr>
<th>Postcode</th>
<th>Description</th>
<th>Price</th>
<th>Location</th>
<th>Type</th>
<th>Status</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Postcode</strong></td>
<td><strong>String</strong></td>
<td>+</td>
<td><strong>String</strong></td>
<td><strong>Price</strong></td>
<td><strong>Street, County</strong></td>
<td><strong>Type</strong></td>
</tr>
</tbody>
</table>

- Compute a **score** for each possible realignment (the better the values are aligned to the schema, the highest is the score)
- Choose the realignment that **maximizes** the score
- Rather than a single realignment, compute **clusters** of different realignment
Identifies the repair operations

<table>
<thead>
<tr>
<th>Price</th>
<th>County</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detached House for £350,000</td>
<td>Detached House, Oxfordshire</td>
</tr>
<tr>
<td></td>
<td>Buckinghamshire, Apartment Flat</td>
</tr>
</tbody>
</table>

$previous_xpath$ = 
//li[@name='summary-list-item']
//p[@class='price']/span/text()  

substring-after(‘£’, $previous_xpath$)

Not enough evidence to generalize an XPath expression that matches both cases -> can’t repair wrapper
Repair the wrapper and the records

1. Repair the wrapper only with those changes that fall within the expressive power of the wrapping language.

2. Re-run the new wrapper and check whether the new extracted data are similar to the expected data (after the repair actions) - **Iterative**?

3. Clean directly the records with the *other* repair operations (the ones which cannot be translated into repair operations for the wrapper).
The oracle: what if it is wrong?

- The oracle analyses an entire table (website), with all records extracted from the website.
- The high number of examples allow the oracle to be wrong in some cases.
- Differently from the wrapper generation process (where few record examples are analysed), here we inspect every single record.

- Other than just align the attribute boundaries, move pieces of the record around
- Single source at time (less complexity)
- The main target is to repair the wrapper rather than individual records

Repair the wrapper without re-run the whole analysis

Where are we now?

- Ready to run our approach on DIADEM dataset (10K real-estate websites with 800K extracted records)
- Exploit the possibility of extending classic XPath functionalities (semantic wrapper)
- Building a suitable web record matching strategy to be applied after the repair step
Questions

stefano.ortona@cs.ox.ac.uk

http://diadem.cs.ox.ac.uk/