Wrapper Learning

Kristina Lerman
University of Southern California

This presentation is based on slides prepared by Craig Knoblock
GIVE ME:
Thai food < $20
“A”-rated

Thai < $20

“A” rated
Wrapper Induction

Problem description:
• Web sources present data in *human-readable format*
  • take user query
  • apply it to data base
  • present results in “template” HTML page

• Wrappers learn extraction rules that correctly extract data from Web pages
  • User supplies labeled examples → system learns correct extraction rules
  • Extraction rules are based on landmarks – tokens on HTML page
<html><b>Restaurants</b><p><ul>
<li><b>Kim’s</b> Phone: <i>(800) 757-1111</i> Review: ...
<li><b>John’s</b> Phone: <i>(888) 111-1111</i> Review: ...
</ul></html>
Wrapper’s Extraction Rules

<html><b>Restaurants</b><p><ul>
<li><b>Kim’s</b> Phone: <i>(800) 757-1111</i> Review: ...
<li><b>John’s</b> Phone: <i>(888) 111-1111</i> Review: ...

• Extraction rules
  • For Name ➔ SkipTo(<b>)
  • For Phone ➔ SkipTo(<i>)

• Extract (Name, Phone) pairs
  • (Kim’s, (800) 757-1111)
  • (John’s, (888) 111-1111)
Wrapper’s Extraction Rules

html><b>Restaurants</b><p><ul>
<li><i>Kim’s</i> Phone: (800) 757-1111. Review: …
<li><i>John’s</i> Phone: (888) 111-1111. Review: …

• Extraction rules
  • For **Name** → SkipTo(<b>)
  • For **Phone** → SkipTo(<i>)

• Extract (Name, Phone) pairs
  • (; Kim’s)
  • (; John’s)
Wrapper Maintenance

Problem

• Landmark-based extraction rules are fast and efficient…but they rely on stable Web Page layout.
• If the page layout changes, the wrapper fails!
• Unfortunately, the average site on the Web changes layout more than twice a year.
• Requirement: Need to detect changes and automatically re-induce extraction rules when layout changes
Learning Regular Expressions
[Goan, Benson, & Etzioni, 1996]

• Character level description of extracted data
• Based on ALERGIA [Carrasco and Oncina, 1994]
  • Stochastic grammar induction algorithm
  • Merges too many states resulting in over-general grammar
• WIL reduced faulty merges by imposing syntactic categories:
  • Number, lower upper, and delim
• Only merges when nodes contain the same syntactic category
• Requires large number of examples to learn
• Computationally expensive
Learning Global Properties for Wrapper Verification [Kushmerick, 1999]

• Each data field described by a numeric features
  • Word count
  • Average word length
  • HTML density
  • Alphabetic density

• Computationally efficient
  • Features are global (computed over all fields)
Learned Features

- Extracted data (wrapper working correctly)
  - (Kim’s, (800) 757-1111)
  - (John’s, (888) 111-1111)
- Name field
  - Average word count = 1
  - Average word length = 5.5
  - Alphabetic density \( =\frac{4/5 + 5/6}{2} = 0.82 \)
- Phone field
  - Average word count = 2 (depends on tokenization)
  - Average word length = 6.5
  - Alphabetic density = 0.74
Using Learned Features to Verify Wrappers

- Learn features on one set of data extracted when wrapper is working correctly
- See if learned features apply to new data
Learned Features

- Extracted data (page changed)
  - (, Kim’s)
  - (, John’s)

- Name field
  - Average word count = 0 (was 1)
  - Average word length = 0 (was 5.5)
  - Alphabetic density = 0 (was 0.82)

- Phone field
  - Average word count = 1 (was 2)
  - Average word length = 4 (was 6.5)
  - Alphabetic density = 0.8 (was 0.74)
Learned Features

- Extracted data → wrapper not working correctly
  - (, Kim’s)
  - (, John’s)
- Name field
  - Average word count = 0 (was 1)
  - Average word length = 0 (was 5.5)
  - Alphabetic density = 0 (was 0.82)
- Phone field
  - Average word count = 1 (was 2)
  - Average word length = 4 (was 6.5)
  - Alphabetic density = 0.8 (was 0.74)
Using Learned Features to Verify Wrappers

- Evaluated on a data set consisting of pages which have changed
- HTML density alone could account for many changes in the test set
- Large number of false negatives on real changes to web sources [Lerman, Knoblock, Minton, 2002]
Learning Data Prototypes
[Lerman & Minton, 2000]

- Approach to learning the structure of data
- Token level syntactic description
  - descriptive but compact
  - computationally efficient
- Structure is described by a sequence of general and specific tokens – *pattern*

Phone
(310) 448-8714
(310) 448-8775
(424) 555-1212

start with:
(Number)

end with:
Number-Number
Learning Data Prototypes
[Lerman & Minton, 2000]

- Also can apply to data with less structure

**STREET_ADDRESS**
220 Lincoln Blvd
420 S Fairview Ave
2040 Sawtelle Blvd

**start with:**
[NUM _CAPS
**end with:**
_CAPS Blvd
_CAPS _CAPS
Token Syntactic Hierarchy

- Tokens = words
- Syntactic types
  - e.g., NUMBER, ALPHA
- Hierarchy of types
  - allows generalization
- Extensible
  - new types
  - domain-specific information
Prototype Learning Algorithm

• No explicit negative examples
• Learn from positive examples of data
• Find patterns that
  • describe many of the positive examples of data
  • highly unlikely to describe a random token sequence (implicit negative examples)
• are statistically significant patterns at $\alpha=0.05$ significance level
• **DataPro** – efficient (greedy) algorithm
DataPro Algorithm

- Process examples
- Seed patterns
- Specialize patterns loop
  - Extend the pattern
    - find a more specific description
    - is the longer pattern significant given the shorter pattern?
  - Prune generalizations
    - is the pattern ending with general type significant given the patterns ending with specific tokens

Examples:
220 Lincoln Blvd
420 S Fairview Ave
2040 Sawtelle Blvd
Examples: PHONE

( 310 ) 577 - 8182
( 310 ) 652 - 9770
( 310 ) 396 - 1179
( 310 ) 477 - 7242
( 626 ) 792 - 9779
( 310 ) 823 - 4446
( 323 ) 870 - 2872
( 310 ) 855 - 9380
( 310 ) 578 - 2293
( 310 ) 392 - 5751
( 805 ) 683 - 8864
( 310 ) 301 - 1004
( 626 ) 793 - 8123
( 310 ) 822 - 1511

• starting patterns:
  ( __NUM ) __NUM -- __NUM
  ( 310 ) __NUM -- __NUM

• ending patterns:
  ( __NUM ) __NUM -- __NUM
Example: STREET_ADDRESS

13455 Maxella Ave
903 N La Cienega Blvd
110 Navy St
2040 Sawtelle Blvd
87 E Colorado Blvd
4325 Glencoe Ave
2525 S Robertson Blvd
998 S Robertson Blvd
523 Washington Blvd
220 Lincoln Blvd
420 S Fairview Ave
13490 Maxella Ave
363 S Fair Oaks Ave
4676 Admiralty Way

• starting patterns:
  _NUM S _CAPS Blvd
  _NUM _CAPS Ave
  _NUM _CAPS

• ending patterns:
  _NUM _CAPS _CAPS _NUM S _CAPS Blvd
  _NUM _CAPS Ave
  _NUM _CAPS Blvd
Wrapper Verification

Learned patterns can be used for web wrapper maintenance applications.

- Automatically detect when the wrapper is no longer correctly extracting data from an information source
  - (Kushmerick 1999)
Going back to our old

• **Extracted data**
  • (Kim’s, (800) 757-1111)
  • (John’s, (888) 111-1111)

• **Patterns**
  • Name: _CAPS
    • Describes 2 examples of Name
  • Phone: ( _NUM ) _NUM - _NUM
    • Describes 2 examples of Phone

• **New data extracted by wrapper**
  • ( , Kim’s)
  • ( , John’s)
Wrapper Verification

Given

- Set of correct old examples of data
- Set of new examples
- Do the patterns describe the same proportions of new examples as old examples?
Wrapper Verification

Results

- Monitored 27 wrappers (23 distinct sources)
- There were 37 changes over ~ 1 year
- Algorithm discovered 35/37 changes with 15 mistakes
  - 13 false positives
- Overall:
  - Average precision = 73%
  - Average recall = 95%
  - Average accuracy = 97%
Wrapper Reinduction

- Rebuild the wrapper automatically if it is not extracting data correctly from new pages
- Data extraction step
  - Identify correct examples of data on new pages
- Wrapper induction step
  - Feed the examples, along with the new pages, to the wrapper induction algorithm to learn new extraction rules
The Lifecycle of A Wrapper

- GUI
- To be labeled
- Wrapper Induction System
- Web pages
- Wrapper
- Extracted data
- Automatic Re-labeling
- Wrapper Verification

USC Information Sciences
Automatic Relabeling Algorithm

1. Given a set of examples of a field, learn patterns describing it
2. Used learned patterns to extract all matching strings on the new pages (possible candidates of a field)
3. Group candidates by context
   - Context = neighboring landmarks
   - Candidates surrounded by the same landmarks grouped together
4. Score the group by degree of overlap with old examples
   - High score if contains some of the same strings as the old examples
### Phone Search Results

**Showing 1 - 1 of 1**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Phone (click to call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>600 S Curson Ave, Los Angeles, CA 90036-3666</td>
<td>(323)936-5549</td>
</tr>
</tbody>
</table>

First | Prev | Next | Last

Search Again

---

USC Information Sciences Institute

**Example Source Change**
### Phone Search Results

**Showing 1 - 2 of 2**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Phone (click to call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>Mar Vista Calif</td>
<td>(310)822-9994</td>
</tr>
<tr>
<td>Andrew Philpot</td>
<td>600 S Curson Ave</td>
<td>(323)936-5549</td>
</tr>
</tbody>
</table>

---

## Whitepages Wrapper

### Phone Search Results

**Showing 1 - 2 of 2**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Phone (click to call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>Mar Vista Calif</td>
<td>(310)822-9994</td>
</tr>
<tr>
<td>Andrew Philpot</td>
<td>600 S Curson Ave</td>
<td>(323)936-5549</td>
</tr>
</tbody>
</table>

---

### NAME item

Begin_Rule

SkipTo __*_

End_Rule

SkipTo </td> <td nowrap >

### ADDRESS item

Begin_Rule

SkipTo </td> <td nowrap >

End_Rule

SkipTo <br>

---

### Whitepages Wrapper

#### Phone Search Results

**Showing 1 - 2 of 2**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Phone (click to call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>Mar Vista Calif</td>
<td>(310)822-9994</td>
</tr>
<tr>
<td>Andrew Philpot</td>
<td>600 S Curson Ave</td>
<td>(323)936-5549</td>
</tr>
</tbody>
</table>
## Phone Search Results

**Showing 1 - 1 of 1**

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone (Click to Call)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>(323)936-5549</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First</th>
<th>Prev</th>
<th>Next</th>
<th>Last</th>
</tr>
</thead>
</table>

### NAME item

```
Begin_Rule
SkipTo _*_
End_Rule

SkipTo </td> <td nowrap >
```

### ADDRESS item

```
Begin_Rule
SkipTo </td> <td nowrap >
End_Rule
SkipTo <br>
```

### CITY

```
600 S Curson Ave<br> Los Angeles
```
<table>
<thead>
<tr>
<th>NAME</th>
<th>ADDRESS</th>
<th>CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Philpot</td>
<td>600 S Curson Ave</td>
<td>Los Angeles</td>
</tr>
</tbody>
</table>
Lindbergh
by A. Scott Berg

List Price: $30.00
Our Price: $21.00
You Save: $9.00 (30%)

Availability: This title usually ships within 2-3 days.

Need this by December 24? No problem. Select shipping method (U.S. addresses).

Click for larger picture

Hardcover - 628 pages (September 1998)
Putnam Pub Group (T); ISBN: 0399144498; Dimensions(in inches): 1.97 x 9.36 x 6.47
Other Editions: Paperback, Audio Cassette (Abridged)

Amazon.com Sales Rank: 3,539
Popular in: U.S. Senate (#5), Laguna Beach, CA (#12). See more
Avg. Customer Review: 
Number of Reviews: 80

<table>
<thead>
<tr>
<th>AUTHOR</th>
<th>TITLE</th>
<th>PRICE</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Scott Berg</td>
<td>Lindbergh</td>
<td>21.00</td>
<td>This title usually ships...</td>
</tr>
</tbody>
</table>
Author: NIL
Title: NIL
Price: 21.00
Availability: This title usually ships within 2-3 days.
After Reinduction

Lindbergh
by A. Scott Berg

List Price: $30.00
Our Price: $21.00
You Save: $9.00 (30%)

Availability: This title usually ships within 2-3 days.

Need this by December 24? Select Next Day shipping method (U.S. addresses).

See larger photo

Hardcover - 623 pages (September 1998)
Putnam Pub. Group (T); ISBN: 0399144468; Dimensions (in inches): 1.97 x 0.26 x 6.47
Other Editions: Paperback, Audio Cassette (Abridged)

Amazon.com Sales Rank: 3,711
Popular in: U.S. Senate (#5), Laguna Beach, CA (#12). See mc
Avg. Customer Review: 
Number of Reviews: 81

<table>
<thead>
<tr>
<th>AUTHOR</th>
<th>TITLE</th>
<th>PRICE</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Scott Berg</td>
<td>Lindbergh</td>
<td>21.00</td>
<td>This title usually ships...</td>
</tr>
</tbody>
</table>
Wrapper Reinduction

Results

• Monitored 10 distinct sources
• There were 8 changes over ~ 1 year
• Extracting examples:
  • 277/338 correct (82%)
  • 31 false positives/30 false negatives
• Reinduction:
  • Average recall = 90%
  • Average precision = 80%
Discussion

- Flexible data representation scheme
- Algorithm to learn description of data fields
- Used in wrapper maintenance applications

Limitations:
- Needs to be extended to lists and tables
- Excellent recall, but lower recall will precision in many false positives