Wrapper Learning

Kristina Lerman
University of Southern California

This presentation is based on slides prepared by Ion Muslea
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

• To integrate data from multiple sources, one must first _extract relevant information_ from Web pages

• Task: learn extraction rules based on labeled examples
  • Hand-writing rules is tedious, error prone, and time consuming
Example of Extraction Task

Wrapper is a procedure that translates a Web page to tuple(s)

NAME: Casablanca Restaurant
STREET: 220 Lincoln Boulevard
CITY: Venice
PHONE: (310) 392-5751
In this part of the lecture ...

- Wrapper Induction Systems
  - WIEN:
    - The rules
    - Learning WIEN rules
  - SoftMealy
- The STALKER approach to wrapper induction
  - The rules
  - The ECTs
  - Learning the rules
Assumes items are always in **fixed, known order**

... Name: J. Doe; **Address:** 1 Main; **Phone:** 111-1111. <p>
Name: E. Poe; **Address:** 10 Pico; **Phone:** 777-1111. <p> ...

Introduces several types of wrappers

- LR:
Wrapper Types

- **LR**
  - L and R delimit each of the k attributes

- **HLRT**
  - Two additional strings:
    - H marks the end of the header
    - T marks the beginning of the tail

- **OCLR**
  - O & C mark the open and close of each tuple
    (row of data in the page)
Rule Learning

- Goal: Find an instance of the given wrapper type that covers the given examples

INPUT:
- Labeled examples: training & testing data
- Admissible rules (hypotheses space)
- Search strategy

Desired output:
- Rule that performs well both on training and testing data

Termination
- Train on sufficient data to be provably approximately correct (PAC)
<html><b>Restaurants</b><p><ul>
<li><b>Kim’s</b> Phone: <i>(800) 757-1111</i></li>
<li><b>Joe’s</b> Phone: <i>(888) 111-1111</i></li>
</ul><hr><b>End</b></html>
Learning LR extraction rules

Restaurants

- **Kim's**
  - Phone: *(800) 757-1111*

- **Joe's**
  - Phone: *(888) 111-1111*

Admissible rules:
- prefixes & suffixes of items of to be extracted

Search strategy:
- start with shortest prefix & suffix, and expand until correct
Learning LR extraction rules

- **Admissible rules:**
  - prefixes & suffixes of items of to be extracted

- **Search strategy:**
  - start with shortest prefix & suffix, and expand until correct

```html
<html><b>Restaurants</b><p><ul>
<li><b>Kim’s</b> Phone: <i>(800) 757-1111</i></li>
<li><b>Joe’s</b> Phone: <i>(888) 111-1111</i></li>
</ul><hr><b>End</b></html>```
Learning LR extraction rules

- **Admissible rules:**
  - prefixes & suffixes of items to be extracted

- **Search strategy:**
  - start with shortest prefix & suffix, and expand until correct

```
<html><b>Restaurants</b></html>
<ul>
  <li><b>Kim’s</b> Phone: <i>(800) 757-1111</i></li>
  <li><b>Joe’s</b> Phone: <i>(888) 111-1111</i></li>
</ul><hr><b>End</b></html>
```
Extraction using LR wrapper

<html><b>Restaurants</b></html>

<ul>
<li><b>Kim’s</b> Phone: <i>(800) 757-1111</i></li>
<li><b>Joe’s</b> Phone: <i>(888) 111-1111</i></li>
</ul>

<hr>

<b>End</b>

- LR wrapper \{ (b>, <), (<i>, <) \} extracts tuples
  - (Kim’s, (800) 757-1111)
  - (Joe’s, (888) 111-1111)
Extraction using LR wrapper

- If an attribute is missing, LR extracts
  - (Kim’s, (888) 111-1111)

- Not correct!
OCLR wrapper

- More expressive class of wrappers
- Extract tuples
  - O – open tuple delimiter
  - C – close tuple delimiter
- Then attributes of tuples
  - L – left field delimiter
  - R – right field delimiter
- Candidates for delimiters must be considered together
Extracting with OCLR wrapper

\[
\begin{html}
<b>Restaurants</b> <p> <ul>
<li><b>Kim’s</b> Address: <br>
<li><b>Joe’s</b> Phone: <i>(888) 111-1111</i>
</ul>
<hr>
<b>End</b> 
\end{html}
\]

- OCLR wrapper \{ \texttt{li}, \texttt{br}, (\texttt{b},,), (\texttt{i},,)} \}
  extracts tuples
  
  - (Kim’s)
  - (Joe’s, (888) 111-1111)

Instead of labeling all of the data, use recognizers to find instances of a particular attribute.

Recognizers may be:
- Human labeler
- Regular expressions for Phone numbers
- Company name recognizer (based on Fortune 500 list)
- Etc

Types of recognizers:
- Perfect: Accept all positive instances and reject all negatives
- Incomplete: Reject all negative instances but reject some positives
- Unsound: Accept all positive, but accept some negatives
- Unreliable: Reject some positive instances and accept some negatives
Automatic Wrapper learning

• Types of errors recognizers make
  • Perfect
    • Accept all positive instances and reject all negatives
    • Human labeler
  • Incomplete
    • Reject all negative instances but reject some positives
    • Fortune 500 company name recognizer
  • Unsound
    • Accept all positive, but accept some negatives
    • Phone number recognizers accepts Fax numbers
  • Unreliable
    • Reject some positive instances and accept some negatives
Labeling Data

- Combine imperfect recognizers to accurately label instances for wrapper learning
  - Must specify what type of error recognizer makes
- Combine the constraints on the ordering of attributes with the information on the type of recognizer
  - E.g., If a perfect recognizer says that position 15-19 is the year and an unsound recognizer says that 18-19 is the age, then the later information would be considered a false positive
Summary

- **Advantages:**
  - Fast to learn & extract
  - Some sources could be labeled automatically given an appropriate set of recognizers

- **Drawbacks:**
  - Cannot handle permutations of attributes
  - Entire page must be labeled
  - Requires large number of examples (if automatic labeling does not work)
In this part of the lecture...

- Wrapper Induction Systems
  - WIEN:
    - The rules
    - Learning WIEN rules
  - The STALKER approach to wrapper induction
    - The rules
    - The ECTs
    - Learning the rules
- Wrapper validation and maintenance
Hierarchical wrapper induction
- Decomposes a hard problem in several easier ones
- Extracts items independently of each other
- Each extraction rule is a finite automaton

Benefits
- Can handle pages with many different structures
  - Lists, embedded lists
- Can efficiently learn wrappers from few labeled examples
STALKER: The Wrapper Architecture

Query ─► Data

Information Extractor

EC Tree

Extraction Rules
Extraction Rules

Extraction rule: sequence of landmarks
Landmarks are tokens that help locate information on the page

SkipTo(Phone) SkipTo(<i>) SkipTo(</i>)

Name: Joel’s  Phone: <i>(310) 777-1111</i>  Review: …
More about Extraction Rules

Extraction rules can handle variability on pages

Name: Joel’s  <p> Phone:  <i>(310) 777-1111</i>  </p> Review: ...

Name: Kim’s  <p> Phone (toll free) :  <b>(800) 757-1111</b> </p> ...

Name: Kim’s  <p> Phone: <b>(888) 111-1111</b> </p> Review: ...

Start: EITHER SkipTo( Phone : <i> ) OR SkipTo( Phone ) SkipTo( : <b> )
The Embedded Catalog Tree (ECT)

ECT describes the structure of the page

| Name:     | KFC                  |
| Cuisine:  | Fast Food            |
| Locations:|
| Venice    | (310) 123-4567,     |
|          | (800) 888-4412.     |
| L.A.      | (213) 987-6543.     |
| Encino    | (818) 999-4567,     |
|          | (888) 727-3131.     |

<table>
<thead>
<tr>
<th>RESTAURANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>List (Locations)</td>
</tr>
<tr>
<td>Cuisine</td>
</tr>
<tr>
<td>City</td>
</tr>
<tr>
<td>List (PhoneNumbers)</td>
</tr>
<tr>
<td>AreaCode</td>
</tr>
<tr>
<td>Phone</td>
</tr>
</tbody>
</table>
Learning the Extraction Rules

EC Tree

Labeled Pages

Inductive Learning System

Extraction Rules

GUI
Example of Rule Induction

Training Examples:

Name: Del Taco <p> Phone (toll free): <b>(800) 123-4567</b><p>Cuisine ...

Name: Burger King <p> Phone: (310) 987-9876 <p>Cuisine: ...
Example of Rule Induction

Training Examples:

Name: Del Taco <p> Phone (toll free): <b>(800) 123-4567</b> <p>Cuisine ...

Name: Burger King <p> Phone: <b>(310) 987-9876</b> <p>Cuisine: ...

Initial candidate: 

SkipTo( )
Example of Rule Induction

Training Examples:

Name: Del Taco <p> Phone (toll free): <b>(800) 123-4567</b><p>Cuisine ...

Name: Burger King <p>Phone: (310) 987-9876 <p>Cuisine: ...

Initial candidate:

SkipTo( )

SkipTo(Phone) SkipTo( ) ...

SkipTo( ) SkipTo( )
Example of Rule Induction

Training Examples:

Name: Del Taco <p> Phone (toll free): <b>(800) 123-4567</b> <p>Cuisine: ...

Name: Burger King <p> Phone: (310) 987-9876 <p>Cuisine: ...

Initial candidate:

SkipTo( ( )... SkipTo(Phone) SkipTo( ( )... SkipTo(:) SkipTo( )... SkipTo(Phone) SkipTo(:) SkipTo( )...
Active Learning & Information Agents

- **Active Learning**
  - **Idea:** system selects most informative exs. to label
  - **Advantage:** fewer examples to reach same accuracy

- **Information Agents**
  - One agent may use hundreds of extraction rules
    - *Small reduction* of *examples per rule* $\Rightarrow$ *big impact* on user
  - Why stop at 95-99% accuracy?
    - Select most informative examples to get to 100% accuracy
### Training Examples

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel’s</td>
<td>(310) 777-1111</td>
<td>The chef…</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>Korean…</td>
</tr>
</tbody>
</table>
Which example should be labeled next?

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel’s</td>
<td>(310) 777-1111</td>
<td>The chef...</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>Korean...</td>
</tr>
</tbody>
</table>
Which example should be labeled next?

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel’s</td>
<td>(310) 777-1111</td>
<td>The chef...</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>Korean...</td>
</tr>
<tr>
<td>Chez Jean</td>
<td>(310) 666-1111</td>
<td>...</td>
</tr>
<tr>
<td>Burger King</td>
<td>(818) 789-1211</td>
<td>...</td>
</tr>
<tr>
<td>Café del Rey</td>
<td>(310) 111-1111</td>
<td>...</td>
</tr>
<tr>
<td>KFC</td>
<td>(800) 111-7171</td>
<td>...</td>
</tr>
</tbody>
</table>
Which example should be labeled next?

SkipTo(Phone:)

Training Examples

Name: Joel’s <p> Phone: <b>(310) 777-1111</b> <p> Review: The chef...

Name: Kim’s <p> Phone: <b>(213) 757-1111</b> <p> Review: Korean ...

Unlabeled Examples

Name: Chez Jean <p> Phone: <b>(310) 666-1111</b> <p> Review: ...

Name: Burger King <p> Phone: <b>(818) 789-1211</b> <p> Review: ...

Name: Café del Rey <p> Phone: <b>(310) 111-1111</b> <p> Review: ...

Name: KFC <p> Phone: <b>(800) 111-7171</b> <p> Review:...
Multi-view Learning

Two ways to find start of the phone number:

SkipTo( Phone: ) BackTo( Number )

Name: KFC Phone: (310) 111-1111 Review: Fried chicken …
Co-Testing

RULE 1

RULE 2

Labeled data

Unlabeled data

ISI

USC Information Sciences Institute
<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joel's</td>
<td>(310) 777-1111</td>
<td>...</td>
</tr>
<tr>
<td>Kim’s</td>
<td>(213) 757-1111</td>
<td>...</td>
</tr>
<tr>
<td>Chez Jean</td>
<td>(310) 666-1111</td>
<td>...</td>
</tr>
<tr>
<td>Burger King</td>
<td>(818) 789-1211</td>
<td>...</td>
</tr>
<tr>
<td>Café del Rey</td>
<td>(310) 111-1111</td>
<td>...</td>
</tr>
<tr>
<td>KFC</td>
<td>(800) 111-7171</td>
<td>...</td>
</tr>
</tbody>
</table>
Not all queries are equally informative
Weak Views

- Learn “content description” for item to be extracted
  - Too general for extraction
    - \((Nmb)\) \(Nmb - Nmb\) can’t tell a phone number from a fax number
  - Useful at \textit{discriminating} among \textit{query candidates}

- Learned field descriptions
  - Starts with: \((Nmb)\)
  - Ends with: \(Nmb - Nmb\)
  - Contains: \(Nmb\ Punct\)
  - Length: \([6,6]\)
Naïve & Aggressive Co-Testing

- **Naïve Co-Testing:**
  - Query: randomly chosen contention point
  - Output: extraction rule with fewest mistakes on queries

- **Aggressive Co-Testing:**
  - Query: contention point that most violates weak view
  - Output: committee vote
    - If extracted strings are same, new extraction rule is generated
    - Otherwise, rule that violates fewer constraints of weak view is kept
Empirical Results: 33 Difficult Tasks

- 140 extraction tasks
  - Number of labeled examples required to learn rules
- 33 most difficult of the 140 extraction tasks
  - Each view: > 7 labeled examples for best accuracy
  - At least 100 examples for task

![Extraction tasks vs. labeled examples graph]

USC Information Sciences Institute
Results in 33 Difficult Domains

Extraction Tasks

Examples to 100% accuracy
Results in 33 Difficult Domains

Extraction Tasks

Examples to 100% accuracy

Naïve Co-Testing vs Random sampling
Results in 33 Difficult Domains

Extraction Tasks

Examples to 100% accuracy

Aggressive Co-Testing  Naïve Co-Testing  Random sampling
Summary

• Advantages:
  • Powerful extraction language (eg, embedded list)
  • One hard-to-extract item does not affect others

• Disadvantage:
  • Does not exploit item order (sometimes may help)
Discussion

• Basic problem is to learn how to extract the data from a page

• Range of techniques that vary in the
  • Learning approach
  • Rules that can be learned
  • Efficiency of the learning
  • Number of examples required to learn

• Regardless, all approaches
  • Require labeled examples
  • Are sensitive to changes to sources