Chapter 4  Webpage Information Extraction

Li fang
Dept.of Computer Science
Shanghai Jiao Tong university

lecture of Internet-based IE technologies
Contents

• Overview of Information Extraction tools from Web pages
• Wrapper Induction
• Wrapper Maintenance
Two kinds of webpages

- **Multiple-record page extraction (left)**
- **One-record page extraction (right)**
<table>
<thead>
<tr>
<th>Tools</th>
<th>Degree of Automation</th>
<th>Support for Complex Objects</th>
<th>GUI</th>
<th>XML Output</th>
<th>Support for Non-HTML Sources</th>
<th>Type of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minerva</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
</tr>
<tr>
<td>TSIMMIS</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>Partial</td>
</tr>
<tr>
<td>Web-OQL</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>No</td>
<td>None</td>
<td>Semi-Structured Data</td>
</tr>
<tr>
<td>HTML-aware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4F</td>
<td>Semi-Automatic</td>
<td>Coding</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
</tr>
<tr>
<td>XWRAP</td>
<td>Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
<td>Semi-Structured Text</td>
</tr>
<tr>
<td>RoadRunner</td>
<td>Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
<td>Semi-Structured Data</td>
</tr>
<tr>
<td>NLP-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHISK</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Full</td>
</tr>
<tr>
<td>RAPIER</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Full</td>
</tr>
<tr>
<td>SRV</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Full</td>
</tr>
<tr>
<td>Induction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WIEN</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td>SoftMealy</td>
<td>Semi-Automatic</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td>STALKER</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Partial</td>
</tr>
<tr>
<td>Modeling-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoDoSE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
</tr>
<tr>
<td>DEByE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
</tr>
<tr>
<td>Ontology-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYU</td>
<td>Manual</td>
<td>Coding</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Full</td>
</tr>
</tbody>
</table>

Summary of the Qualitative Analysis:

Methods of extraction pages

ST: semi-structured text

SD: semi-structured data
Language for wrapper development—for manually constructed IE systems

**Minerva:** combines a declarative grammar-based approach with features typical of **procedural programming languages**.

**Tsimmis:** includes wrappers that can be configured through **specification files** written by the user.

**Web-OQL:** originally aimed at performing SQL-like queries over the Web.
Web-OQL

- Hypertrees are arc-labeled ordered trees.

Tag name
- The piece of HTML code
- Text excluding markup
Web-OQL (cont.)

• Query: extracts the reviewer names “Jeff” and “Jane” from page pe2.
Overview of Web data extraction tools (cont.)

- **HTML-aware Tools**
  - W4F (world wide web wrapper factory): a toolkit for building wrappers.
  - XWRAP: a component library that provides basic building blocks for wrapper development.

- **NLP-based tools**
  PAPIER (job posting), SRV, WHISK: suitable for Web pages consisting of grammatical text, such as job listings, apartment rental advertisements, seminar announcements.
NLP based tools: PAPIER

<table>
<thead>
<tr>
<th>BookTitle extraction rule-</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pre-filler pattern</strong></td>
<td><strong>Filler pattern</strong></td>
</tr>
<tr>
<td>(1) word: Book</td>
<td>list: len: 2</td>
</tr>
<tr>
<td>(2) word: Name</td>
<td>Tag: [nn, nns]</td>
</tr>
<tr>
<td>(3) word: &lt;/b&gt;</td>
<td></td>
</tr>
</tbody>
</table>

- Extraction rule for the **book title**: 

- Preceded by words “Book”, “Name”, and “</b>”
- Followed by the word “<b>”.
- The “Filler pattern” specifies that the title **consists of at most two words** that were labeled as “nn” or “nns” by the POS tagger (i.e., one or two singular or plural common common nouns).

lecture of Internet-based IE technologies
Overview of Web data extraction tools (cont.)

• **Modeling-based Tools**
  
  NoDoSE: an interactive tool for semi-automatically determining the structure of Web page.

  DEByE: an interactive tool to extract page contents based on a set of example objects.

• **Ontology-based Tools**

  Ontologies are previously constructed to describe the data of interest, including relationships, lexical appearance and context keywords.
# Overview of Web data extraction tools

<table>
<thead>
<tr>
<th>Tools</th>
<th>Degree of Automation</th>
<th>Support for Complex Objects</th>
<th>GUI</th>
<th>XML Output</th>
<th>Support for Non-HTML Sources</th>
<th>Type of Page Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minerva</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>TSIMMIS</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Web-OQL</td>
<td>Manual</td>
<td>Coding</td>
<td>No</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>HTML-aware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4F</td>
<td>Semi-Automatic</td>
<td>Coding</td>
<td>No</td>
<td>Yes</td>
<td>Full</td>
<td>ST</td>
</tr>
<tr>
<td>XWRAP</td>
<td>Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>RoadRunner</td>
<td>Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>NLP-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WHISK</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Full</td>
<td>ST</td>
</tr>
<tr>
<td>RAPIER</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>SRV</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Induction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WIEN</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>SoftMealy</td>
<td>Semi-Automatic</td>
<td>Partial</td>
<td>Yes</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>STALKER</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Modeling-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoDoSE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>DEByE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Ontology-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYU</td>
<td>Manual</td>
<td>Coding</td>
<td>Yes</td>
<td>No</td>
<td>Full</td>
<td>ST/SD</td>
</tr>
</tbody>
</table>

**Table 1: Summary of the Qualitative Analysis**

**WRAPPER induction tools**: WIEN, SoftMealy, Stalker
Wrapper Technologies

- What is wrapper
- Wrapper induction
- Wrapper maintenance
What is Wrapper?

• For information integration
  A procedure that is designed for extracting content of a particular information source and delivering the content of interesting in a self-describing representation (eg. XML)

• For Web application
  – An extracting program to extract desired information from Web pages.

Semi-Structure Doc. → wrapper → Structure Info.
For Web Applications:

- Given a Web page $S$ containing a set of implicit objects, determine a mapping $W$ that populates a data repository $R$ with the objects in $S$. 

![Diagram showing the process: Page S → Mapping W → Repository R with a Wrapper]
An example for a wrapper

procedure \texttt{ccwrap}_{LR}(page \texttt{P})
while there are more occurrences in \texttt{P} of \texttt{<B>''}
for each \((\ell_k, r_k) \in \{(\texttt{<B>''}, \texttt{</B>''}, \texttt{<I>''}, \texttt{</I>''})\})
scan in \texttt{P} to next occurrence of \(\ell_k\); save position as start of \(k\)th attribute
scan in \texttt{P} to next occurrence of \(r_k\); save position as end of \(k\)th attribute
return extracted \{\ldots, \texttt{(country, code), \ldots}\} pairs
Wrapper Induction

- Web wrappers wrap...
  - “Query-able” or “Search-able” Web sites
  - Web pages with large itemized lists
- The primary issues are:
  - How to build the extractor quickly?
  - Wrapper induction algorithms search a hypothesis space of possible wrapper programs for a wrapper that has high extraction accuracy on a set of training pages.
Wrapper Induction: Methods

- **Manually writing wrappers**
  - Tedious, time consuming task, eg. TSIMMIS, Minerva, ...

- **Wrapper programming languages**
  - Florid (a logic-programming formalism), pillow (an HTML/XML programming library for logic programming systems) ...

- **Machine learning methods**
  - Stalker, Softmealy, WIEN ...

- **Supervised interactive wrapper**
  - W4F (uses an SQL-like query called HEL), Xwrap (uses a procedural rule system), ...
Wrapper Induction Tools

• WIEN:
  • Input: a set of pages where data of interest is labeled to serve as examples
  • Output: a wrapper that is consistent with each labeled page.

• SoftMealy
  • Using finite-state transducers (FST) which takes a sequence of tokens as input and matches the context separators with contextual rules to determine state transitions

• Stalker
  • The wrapper induction techniques used in WIEN and SoftMealy are further developed in Stalker
Wrapper Induction: machine learning methods (Stalker)

Our focus here

Figure 1: The Lifecycle of a Wrapper
Learning Extraction Rules
---from pages

• Aim:
Defining a set of extraction rules that precisely define how to locate the information on the page.

→ How to describe the content of a page?
Describing the content of a page: Embedded Catalog Tree

- Embedded catalog (EC): a tree-like structure to represent a Web page.
- Leaves: items of interest for the user
- Internal nodes: lists of k-tuples where each item in the k-tuple can be either a leaf or another list L.
Embedded Catalog Tree (for example)

A list of five tuples

Figure 2: EC description of LA-Weekly pages.

LIST (LA-Weekly restaurants)

TUPLE (LA-Weekly restaurant)

name address phone review

LIST (CreditCards)

credit_card

LA Restaurants

Search Criteria: Name: Shrimp Latest: Any Cuisine: Any

KILLER SHRIMP
523 Washington Blvd., Mar Vista del Rey
(310) 876-2239

Find us on the grid in West LA and your eyes can’t help but notice this little building housing a heavenly eatery. Here’s the view: Killer Shrimp is a popular hot spot and has become one of LA’s hottest dining experiences. Come and check it out and see if you know that this is the place... seriously serious for the price. Indoor and outdoor dining. Lunch and dinner seven days. Beer and wine. Sunday brunch. MC, V.

KILLER SHRIMP
419 N. Pacific Coast Hwy., Redondo Beach
(310) 790-9080

Find us on the grid in West LA and your eyes can’t help but notice this little building housing a heavenly eatery. Here’s the view: Killer Shrimp is a popular hot spot and has become one of LA’s hottest dining experiences. Come and check it out and see if you know that this is the place... seriously serious for the price. Indoor and outdoor dining. Lunch and dinner seven days. Beer and wine. Sunday brunch. MC, V.

KILLER SHRIMP
4088 Collins Ave., Studio City
(310) 502-8570

Find us on the grid in West LA and your eyes can’t help but notice this little building housing a heavenly eatery. Here’s the view: Killer Shrimp is a popular hot spot and has become one of LA’s hottest dining experiences. Come and check it out and see if you know that this is the place... seriously serious for the price. Indoor and outdoor dining. Lunch and dinner seven days. Beer and wine. Sunday brunch. MC, V.
Extracting Rule based on EC

• **A rule:** for each node $x$ in the EC Tree, the wrapper needs a rule $r$ that extracts that particular node from its root, $p$ is a path from the root to the leaf.

• **A list iteration rule:** decomposes $p$ into individual tuples, and then apply $r$ to each extracted tuple for each list node.
Example for extraction rules

R1=SkipTo(Address) SkipTo(I)
R2=SkipTo(Address : <I>)
R3=SkipTo(Cuisine : <I>) SkipTo(Address : <I>)

...Cuisine:<i>Seafood</i><p>Address:<i> 12 Pico St. </i><p>Phone:<i>...</p>...Cuisine:<i>Thai</i>  
   </i><p>Address:<i> 512 Oak Blvd. </i><p>Phone:<i>...</p>...Cuisine:<i>Burgers</i>  
   </i><p>Address:<i> 416 Main St. </i><p>Phone:<i>...</p>...Cuisine:<i>Pizza</i>  
   </i><p>Address:<i>b 97 Adams Blvd. </i><p>Phone:<i>...
Example for extraction rules (cont.)

R4=SkipTo(Cuisine : <I> _Capitalized_</I> <p> Address : <I>)

R4 is defined based on a 9-token landmark that uses the wildcard_Capitalized_, which is a placeholder for any capitalized alphabetic string.

Disjunctive rules: either R1 or R2

To deal with variations in the format of the documents, disjunctions are allowed to use.
Extraction Rules as Finite Automata

- **Landmarks:** each argument of a `skipTo()`
  - A sequence of tokens and wildcards
- **Landmark automata**
  - A non-deterministic finite automata

\[ l_{i,j} \rightarrow S_{i,j} \]

The transition takes place if the automaton is in the state \( S_i \) and the landmark \( l_{i,j} \) matches the sequence of tokens at the input.
Landmark Automata (linear LA)

- A linear LA has **one accepting state**.
- From each non-accepting state, there are exactly **two possible transitions**: a loop to itself, and a transition to the next state;
- Each non-looping transition is labeled by a **landmarks**;
- All looping transitions have the meaning "consume all tokens until you encounter the landmark that leads to the next state".
Rules and its automaton

R1::=skipTo((),
R2::=skipTo(phone)
skipTo(<b>).

Disjunctive rule either R1 or R2

An SLG for the start of the area code.

- The initial state $S_0$ has a branching-factor of k.
- It has exactly k accepting states. (one per branch)
- All k branches that leave the $S_0$ are sequential LA.
STALKER algorithm

- it accepts the positive examples in E2 and E4
- it rejects both E1 and E3 because R1 can not be matched on them. R2 can do.

R1::=skipTo(),
R2::=skipTo(-<b>)
Process of the example (STALKER)

1. Generating a linear LA that covers as many as possible of the four positive examples.
2. Create another linear LA for the remaining examples, and so on.
3. Once STALKER covers all examples. It returns the disjunction of all the induced LAs.
STALKER Algorithm

LearnRule( Examples )
- let RetVal be an empty rule
- WHILE Examples ≠ Ø
  - aDisjunct = LearnDisjunct(Examples)
  - remove all examples covered by aDisjunct
  - add aDisjunct to RetVal
- return OrderDisjuncts(RetVal)

LearnDisjunct( Examples )
- let Seed ∈ Examples be the shortest example
- Candidates = GetInitialCandidates( Seed )
- DO
  - BestRefiner = GetBestRefiner( Candidates )
  - BestSolution = GetBestSolution( Candidates ∪ {BestSolution} )
  - Candidates = Refine(BestRefiner, Seed)
  WHILE IsNotPerfect(BestSolution) AND BestRefiner ≠ Ø
- return PostProcess(BestSolution)

lecture of Internet-based IE technologies
STALKER Algorithm (cont.)

**BestRefiner()**
Prefer candidates that have:
- larger coverage
- more early matches
- more failed matches
- fewer wildcards
- shorter unconsumed prefixes
- fewer tokens in \textit{SkipUntil()}
- longer end-landmarks

**BestSolution()**
Prefer candidates that have:
- more correct matches
- more failures to match
- fewer tokens in \textit{SkipUntil()}
- fewer wildcards
- longer end-landmarks
- shorter unconsumed prefixes

---

Figure 6. The stalker heuristics.

lecture of Internet-based IE technologies
STALKER Algorithm (cont.)

- Refine() function: obtain better disjuncts either by making its landmarks more specific (landmark refinements), or by adding new states in the automaton (topology refinements).
- Landmark refinements
- Topology refinements
Landmark Refinement

- R4 = SkipTo<b>

Refine as:

\[
R7 = \text{SkipTo}( - <b>)
\]

\[
R8 = \text{SkipTo}(\text{Punctuation} <b>)
\]

\[
R9 = \text{SkipTo}(\text{Anything} <b>)
\]
Topology
Refinements

• $R4 = \text{skipTo}<b>$

Refine as:

R10: $\text{skipTo(Venice)}$ $\text{skipTo(<b>)}$
R11: $\text{skipTo(</b>)}$ $\text{skipTo(<b>)}$
R12: $\text{skipTo(:)}$ $\text{skipTo(<b>)}$
R13: $\text{skipTo(-)}$ $\text{skipTo(<b>)}$
R14: $\text{skipTo(,)}$ $\text{skipTo(<b>)}$
R15: $\text{skipTo(Phone)}$ $\text{skipTo(<b>)}$
R16: $\text{skipTo(1)}$ $\text{skipTo(<b>)}$
R17: $\text{skipTo(Numeric)}$ $\text{skipTo(<b>)}$
R18: $\text{skipTo(Punctuation)}$ $\text{skipTo(<b>)}$
R19: $\text{skipTo(HtmlTag)}$ $\text{skipTo(<b>)}$
R20: $\text{skipTo(AlphaNum)}$ $\text{skipTo(<b>)}$
R21: $\text{skipTo(Alphabetic)}$ $\text{skipTo(<b>)}$
R22: $\text{skipTo(Capitalized)}$ $\text{skipTo(<b>)}$
R23: $\text{skipTo(NonHtml)}$ $\text{skipTo(<b>)}$
R24: $\text{skipTo(Anything)}$ $\text{skipTo(<b>)}$

* Figure 4. Four examples of restaurant addresses. *
Each initial candidate is a 2-state landmark automaton that is either a token $t$ that ends one prefix $p$ or a wildcard that matches such a $t$. 

Example of rule induction:

R1 is a perfect disjunct as a result for first iteration.
During the second iteration with E1 and E3 example, the initial candidate rules $R5$ and $R6$.

Refinement: tries to obtain better disjuncts either by marking its landmarks more specific (Landmark refinement) or by adding new states in the automaton (Topology refinements).

A perfect rule which matches examples.

Figure 7: Rule induction (second iteration).
Identifying highly informative examples

- The most **informative** examples illustrate exceptional cases
- **Active learning**: analyzes the set of unlabeled example to **automatically select examples** for the user to label
- forward and backward rules:
  
  $\text{Fwd R1}=\text{SkipTo(Address)}\text{SkipTo(<I>)}$
  $\text{Bwd R1}=\text{BackTo(Phone)}\text{BackTo(_Number_)}$

**If two rules disagree on the sample, which is selected for user to label** – **highly informative training example.**
Results reported from STALKER

• From 28 sources, 206 extraction rules: 182 rules (100% correct), 18 rules (>90%), 3% rules are <90%.

• Active learning:
Average accuracy from 85.7% → 94.2%
STALKER features

• the ability to wrap a larger variety of sources.
• capable of learning most of the extraction rules based on just a couple of examples.
• Using single-slot rules, keep high accuracy.
• improving the efficiency based on active learning for hardest items.
Other Wrappers

- **WIEN**: learns the landmarks by searching *common prefixes* at the *character level*, needs more training data.

- **SoftMealy**: its extraction rules are less expressive than STALKER, complex to deal with missing items and various orderings of items.
<table>
<thead>
<tr>
<th>TICK</th>
<th>LAST</th>
<th>CHG</th>
<th>%</th>
<th>VOL #TRDS</th>
<th>BID ASK</th>
<th>LOW HIGH</th>
<th>PREV OPEN</th>
<th>52LOW HIGH</th>
<th>EPS DIV</th>
<th>DATE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DJ</td>
<td>47.2500</td>
<td>+1.0000</td>
<td>+2.16</td>
<td>140,800</td>
<td>na</td>
<td>46.0000</td>
<td>46.2500</td>
<td>41.5625</td>
<td>0.09</td>
<td>02/22 16:02</td>
</tr>
<tr>
<td>DJM</td>
<td>11.1875</td>
<td>+0.1250</td>
<td>+1.13</td>
<td>28,600</td>
<td>na</td>
<td>11.1250</td>
<td>11.0625</td>
<td>9.5000</td>
<td>0.01</td>
<td>02/22 15:19</td>
</tr>
<tr>
<td>DJT</td>
<td>4.3125</td>
<td>−0.0625</td>
<td>−1.42</td>
<td>167,500</td>
<td>na</td>
<td>4.1250</td>
<td>4.3750</td>
<td>2.7500</td>
<td>−1.79</td>
<td>02/22 15:57</td>
</tr>
<tr>
<td>DK</td>
<td>6.0000</td>
<td>+0.1250</td>
<td>+2.13</td>
<td>21,500</td>
<td>na</td>
<td>5.8125</td>
<td>5.8750</td>
<td>5.0625</td>
<td>−3.03</td>
<td>02/22 15:54</td>
</tr>
<tr>
<td>DL</td>
<td>29.5000</td>
<td>+0.1875</td>
<td>+0.64</td>
<td>382,100</td>
<td>na</td>
<td>29.2500</td>
<td>29.3125</td>
<td>19.5000</td>
<td>1.02</td>
<td>02/22 16:01</td>
</tr>
</tbody>
</table>

**Market Watch.** A detailed look at Market activity.

**APL – Ticker Search – Advertise on QS – Internet Stock Report – Questionnaire**
<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>E-Mail</th>
<th>Last Update</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'Lithium' J Smith</td>
<td><a href="mailto:aulmer@u.washington.edu">aulmer@u.washington.edu</a></td>
<td>08/01/95</td>
<td>University of Washington</td>
</tr>
<tr>
<td>2</td>
<td>'Sir Brand' Gregobin Smith</td>
<td><a href="mailto:sirbrand@u.washington.edu">sirbrand@u.washington.edu</a></td>
<td>06/21/96</td>
<td>University of Washington</td>
</tr>
<tr>
<td>3</td>
<td>Craig Smith</td>
<td><a href="mailto:chs@maxwell.cs.uoregon.edu">chs@maxwell.cs.uoregon.edu</a></td>
<td>08/01/94</td>
<td>University of Oregon</td>
</tr>
<tr>
<td>4</td>
<td>Richard Smith</td>
<td><a href="mailto:GBORDER@SFASU.EDU">GBORDER@SFASU.EDU</a></td>
<td>11/12/95</td>
<td>Stephen F. Austin State University</td>
</tr>
<tr>
<td>5</td>
<td>David S Smith</td>
<td><a href="mailto:dssmith@INDIANA.EDU">dssmith@INDIANA.EDU</a></td>
<td>11/16/95</td>
<td>Indiana University</td>
</tr>
</tbody>
</table>
Result Comparison

**Quote Server**
- **Stalker:** 10 example tuples, 79%, 500 test
- **WIEN:** the collection beyond learn’s capability
- **SoftMealy:** multi-pass 85%, single-pass 97%

**Internet Address Finder**
- **Stalker:** 85% ~ 100%, 500 test
- **WIEN:** the collection beyond learn’s capability
- **SoftMealy:** multi-pass 68%, single-pass 41%
Result Comparison (cont.)

◊ Okra (tabular pages)
  ■ Stalker: 97%, 1 example tuple
  ■ WIEN: 100%, 13 example tuples, 30 test
  ■ SoftMealy: single-pass 100%, 1 example tuple, 30 test

◊ Big-book (tagged-list pages)
  ■ Stalker: 97%, 8 example tuples
  ■ WIEN: perfect, 18 example tuples, 30 test
  ■ SoftMealy: single-pass 97%, 4 examples, 30 test
    multi-pass 100%, 6 examples, 30 test
A General View of Wrapper (as Summarization)

- Machine learning method for Wrapper Induction

Stalker, SoftMealy, Wien, Whisk
Overall Comparison

Three dimensions: the difficulty of IE task, the techniques used, the effort made by the user for the training process and necessity to port IE across different domains.

Conclusion:
- Template-based pages have high automation degree.
- IE cross-site pages and free texts, semantic features are required.
- Manual IE systems can be applied to all kinds of inputs.
- Semi-supervised and unsupervised IE systems can be applied only to template-based pages.
- Unsupervised systems usually apply superficial features.
Problem?

- The Web are very **dynamic**: contents, page structures
- Original wrappers can stop working: rely on Web page structures
- Re-generating wrappers is not easy: **heavy workload** to system developers

.changed documents

<table>
<thead>
<tr>
<th>Original Wrapper</th>
<th>Extract nothing …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Wrapper</td>
<td>Incomplete results</td>
</tr>
<tr>
<td>Original Wrapper</td>
<td>Incorrect results</td>
</tr>
</tbody>
</table>

lecture of Internet-based IE technologies
Example

The original wrapper fails due to the structure change. How to solve it? (discussion)

- Monitoring a set of generic features
- Machine learning techniques to learn a set of patterns that describe the information that is being extracted from each of the relevant fields.

How to solve it? (discussion)
Wrapper Maintenance

• **DataProg algorithm**, which learns structural information (patterns) about a data field from a set of examples of the field.

• **Wrapper verification**: Is a wrapper operating correctly?

• **Wrapper maintenance**: How to automatically modify a wrapper when the pages have changed?

Street address: 12 Pico St., 512 Oak Blvd, 416 Main St. and 97 Adams Blvd.→ (_Number_ _ _ capitalized_) (Blvd.) or (St.)

detecting when a wrapper stops extracting data correctly from a Web source.

identify new examples of the data field in order to rebuild the wrapper if it stops working.
Example for Wrapper Maintenance in STALKER

an example of the original site, the extracting rule for a book title and the extracted results from the example page.

The source and incorrectly extracted result after the titles’s font and color were changed.

Rule changed.
Wrapper Maintenance Methods (Kushmerick’s method)

• Each data field was described by a collection of global features, such as word count, average word length, and density of types.

• Calculated the mean and variance of each feature’s distribution over the training examples.

• Individual feature probabilities are then combined to produce a value.

• If the value exceeds a threshold, the wrapper is correct, otherwise, it is failed.
A prototype for tracking changes to webpages – Microsoft Research

Diff-IE is a prototype Internet Explorer add-on that:

• Highlights the changes to a webpage since the last time you visited it.

• Enables you to view and compare previously cached version of a page.

→Tracking changes to webpages.
Download DIFF-IE

From: Microsoft research


• How it was implemented?
• Cache: stores the previous versions of the page, in order to highlight how a page has changed.

• **Comparison component**: is responsible for detecting and highlighting the changes.

• Toolbar component: is the portion of the application with which the user interacts.
Web page representation

- DiffIE identifies changes to text-based Web content at the Document Object Model (DOM) level. Pages are represented internally as a tree of hash values to support this DOM-level comparison of text across pages.
- The text nodes of a Web page: the leaves of the tree.
- The content of these nodes are hashed using the MD5 algorithm.

MD5:
A message of arbitrary length $\rightarrow$ 128-bit fingerprint

Two messages will not produce the same fingerprint.
Comparing Component (2)

Detecting Differences:

- Starting at the root node, DiffIE compares the pre-computed subtree hash of the live version and the cached version.
- If same, DiffIE terminates comparison of the corresponding subtree, since identical hashes implies the content must not have changed.
Comparison Component (3)

- **4 Types** of Differences: only addition and changes are highlighted.

**Addition**

**Deletion**

**Change**

**Movement**

Figure 3. An illustration of the types of changes that can occur at the DOM level of a Web page.
Application 1

• Monitoring a page for change, to keep track of the latest stock prices, or latest updates on the page.
Application 2

- See new or different search results.

lecture of Internet-based IE technologies
Application 3

Find changes in long lists of text

It can be hard to see changes in long lists of text, but Diff-IE identifies these automatically.

Workshops, Collaborations and Papers:

- Collaborator: "Collaborative Information Retrieval", a multidisciplinary research project to understand the social aspects of information retrieval in a variety of workplace settings. Collaboration with Raya Field and Harry Brance (U Washington iSchool), Steve Pollock (Boeing), A.M. Peterson (Riso National Laboratory), and Jonathan Grudin (Microsoft Research).
- Collaborator: "Keeping Found Things Found", a research project to understand the ways in which people manage information for subsequent re-access. In collaboration with William Jones, Harry Brance and Mike Bloomberg (U Washington iSchool).

Track price changes

We rarely remember prices, but Diff-IE does. Here, the prices of these HP workstations dropped.

» Personal workstations

» Affordable power

- HP Z400 Workstation NEW!
  Starting at: $929.00*
  As low as $27/mo.**
  - Up to 16 GB of system memory
  - Up to 4.5 TB of internal storage
  - Up to NVIDIA Quadro FX4800 or dual NVIDIA Quadro FX1800 graphics

- HP xw4600 Workstation
  Starting at: $679.00*
  As low as $20/mo.**

» Compact power

- HP Z600 Workstation NEW!
  Starting at: $1,589.00*
  As low as $46/mo.**
  - Eight core performance in a compact footprint
  - Up to 24 GB of system memory
  - Up to 4.5 TB of internal storage
  - Up to eight 2D displays
  - Up to dual NVIDIA Quadro FX1800 graphics

- HP xw6600 Workstation
  Starting at: $1,219.00*
  As low as $35/mo.**

» Extreme power

- HP Z800 Workstation NEW!
  Starting at: $1,839.00*
  As low as $53/mo.**
  - Ultimate performance with extreme expandability
  - Up to 192 GB of system memory
  - Up to 7.5 TB of internal storage
  - Up to dual Quadro FX5800 graphics

- HP xw8600 Workstation
  Starting at: $1,339.00*
  As low as $39/mo.**

- HP xw9400 Workstation
  Starting at: $2,599.00*
  As low as $74/mo.**
References


lecture of Internet-based IE technologies
References sites

- Repository of online information sources used in information extraction task: [http://www.isi.edu/info-agents/RISE/index.html](http://www.isi.edu/info-agents/RISE/index.html)
- Papers, tutorials, lectures, code
Summarization

• What is wrapper?
• How to do wrapper induction?
• How to maintenance wrapper?