Extracting and Structuring Web Data

D.W. Embley*, D.M Campbell†, Y.S. Jiang, Y.-K. Ng, R.D. Smith, Li Xu
Department of Computer Science

S.W. Liddle‡
School of Accountancy and Information Systems
Marriott School of Management

D.W. Lonsdale
Department of Linguistics

Brigham Young University
Provo, UT, USA

Funded in part by *Novell, Inc., †Ancestry.com, Inc., ‡Faneuil Research, and the National Science Foundation (NSF).
Information Exchange
**GOAL**

Query the Web like we query a database

Example: Get the year, make, model, and price for 1987 or later cars that are red or white.

<table>
<thead>
<tr>
<th>Year</th>
<th>Make</th>
<th>Model</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>CHEVY</td>
<td>Cavalier</td>
<td>11,995</td>
</tr>
<tr>
<td>94</td>
<td>DODGE</td>
<td>Intrepid</td>
<td>10,000</td>
</tr>
<tr>
<td>94</td>
<td>DODGE</td>
<td>Intrepid</td>
<td>10,000</td>
</tr>
<tr>
<td>91</td>
<td>FORD</td>
<td>Taurus</td>
<td>3,500</td>
</tr>
<tr>
<td>90</td>
<td>FORD</td>
<td>Probe</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>FORD</td>
<td>Escort</td>
<td>1,000</td>
</tr>
</tbody>
</table>
PROBLEM
The Web is not structured like a database.

Example:
<html>
<head>
<title>The Salt Lake Tribune Classifieds</title>
</head>

... 
<hr>
<h4>‘97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her. Previous owner heart broken! Asking only $11,995. #1415 JERRY SEINER MIDVALE, 566-3800 or 566-3888</h4>
<hr>
... 
</html>
Making the Web Look Like a Database

• Web Query Languages
  – Treat the Web as a graph (pages = nodes, links = edges).
  – Query the graph (e.g., Find all pages within one hop of pages with the words “Cars for Sale”).

• Wrappers
  – Parse page to extract attribute-value pairs, form records, and either insert them into a database or filter them wrt a query.
    • Write parser by hand.
    • Use machine learning to discover how to parse a site.
    • Develop an application-specific, site-independent ontology to parse a site.
  – Query the database or present the filtered result.
for unstructured record documents, rich in data and narrow in ontological breadth.
Application Ontology: Object-Relationship Model Instance

Car [-> object];
Car [0..1] has Model [1..*];
Car [0..1] has Make [1..*];
Car [0..1] has Year [1..*];
Car [0..1] has Price [1..*];
Car [0..1] has Mileage [1..*];
PhoneNr [1..*] is for Car [0..1];
PhoneNr [0..1] has Extension [1..*];
Car [0..*] has Feature [1..*];
Application Ontology: Data Frames

Make matches [10] case insensitive
constant
   { extract “chev”; }, { extract “chevy”; }, { extract “dodge”; },
   ...
end;

Model matches [16] case insensitive
constant
   { extract “88”; context “\bolds\S*\s*88\b”; },
   ...
end;

Mileage matches [7] case insensitive
constant
   { extract “[1-9]\d{0,2}k”; substitute “k” -> “,000”; },
   ...
   keyword
      “\bmi\b”, “\bmi\b “\bmi.\b”;
end;

...
Ontology Parser

Make: chevy
... KEYWORD(Mileage): \bmb\iles\b ...
...

Object: Car;
...
Car: Year [0..1];
Car: Make [0..1];
...
CarFeature: Car [0..*] has Feature [1..*];

create table Car (Car integer, Year varchar(2), ...);
create table CarFeature (Car integer, Feature varchar(10));
...

Constant/Keyword Matching Rules

Ontology Parser

Record-Level Objects, Relationships, and Constraints

Database Scheme

Application Ontology
Record Extractor

<html>
...
<h4>‘97 CHEVY Cavalier, Red, 5 spd, …</h4>
<hr>
<h4>‘89 CHEVY Corsica Sdn teal, auto, …</h4>
<hr>
....
</html>

...  
#####
‘97 CHEVY Cavalier, Red, 5 spd, …
#####
‘89 CHEVY Corsica Sdn teal, auto, …
#####
...
‘97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her.
Previous owner heart broken! Asking only $11,995. #1415
JERRY SEINER MIDVALE, 566-3800 or 566-3888

<table>
<thead>
<tr>
<th>Year</th>
<th>97</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make</td>
<td>CHEV</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Make</td>
<td>CHEVY</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Model</td>
<td>Cavalier</td>
<td>11</td>
<td>18</td>
</tr>
<tr>
<td>Feature</td>
<td>Red</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>Feature</td>
<td>5 spd</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>Mileage</td>
<td>7,000</td>
<td>38</td>
<td>42</td>
</tr>
<tr>
<td>KEYWORD(Mileage)</td>
<td>miles</td>
<td>44</td>
<td>48</td>
</tr>
<tr>
<td>Price</td>
<td>11,995</td>
<td>100</td>
<td>105</td>
</tr>
<tr>
<td>Mileage</td>
<td>11,995</td>
<td>100</td>
<td>105</td>
</tr>
<tr>
<td>PhoneNr</td>
<td>566-3800</td>
<td>136</td>
<td>143</td>
</tr>
<tr>
<td>PhoneNr</td>
<td>566-3888</td>
<td>148</td>
<td>155</td>
</tr>
</tbody>
</table>
Heuristics

- Keyword proximity
- Subsumed and overlapping constants
- Functional relationships
- Nonfunctional relationships
- First occurrence without constraint violation
'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her. Previous owner heart broken! Asking only $11,995. #1415.
JERRY SEINER MIDVALE, 566-3800 or 566-3888
'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles. Previous owner heart broken! Asking only $11,995. #1415. JERRY SEINER MIDVALE, 566-3800 or 566-3888
## Functional Relationships

<table>
<thead>
<tr>
<th>Year</th>
<th>97</th>
<th>Mileage</th>
<th>7,000</th>
<th>Price</th>
<th>11,995</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make</td>
<td>CHEV</td>
<td>Model</td>
<td>Cavalier</td>
<td>Feature</td>
<td>Red</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her. Previous owner heart broken! Asking only $11,995. #1415. JERRY SEINER MIDVALE, 566-3800 or 566-3888
Nonfunctional Relationships

'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her. Previous owner heart broken! Asking only $11,995. #1415.

JERRY SEINER MIDVALE, 566-3800 or 566-3888
'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her. Previous owner heart broken! Asking only $11,995. #1415. JERRY SEINER MIDVALE, 566-3800 or 566-3888
Database-Instance Generator

Year|97|2|3  
Make|CHEV|5|8  
Make|CHEVY|5|9  
Model|Cavalier|11|18  
Feature|Red|21|23  
Feature|5 spd|26|30  
Mileage|7,000|38|42  
PRICE|11,995|100|105  
Mileage|11,995|100|105  
PhoneNr|566-3800|136|143  
PhoneNr|566-3888|148|155

```
insert into Car values(1001, “97”, “CHEVY”, “Cavalier”, “7,000”, “11,995”, “566-3800”)  
insert into CarFeature values(1001, “Red”)  
insert into CarFeature values(1001, “5 spd”)```
Recall & Precision

\( N = \) number of facts in source
\( C = \) number of facts declared correctly
\( I = \) number of facts declared incorrectly

\[
\text{Recall} = \frac{C}{N} \quad \text{(of facts available, how many did we find?)}
\]

\[
\text{Precision} = \frac{C}{C + I} \quad \text{(of facts retrieved, how many were relevant?)}
\]
### Results: Car Ads

_Salt Lake Tribune_

<table>
<thead>
<tr>
<th>Feature</th>
<th>Recall %</th>
<th>Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Make</td>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>Model</td>
<td>82</td>
<td>100</td>
</tr>
<tr>
<td>Mileage</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>Price</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>PhoneNr</td>
<td>94</td>
<td>100</td>
</tr>
<tr>
<td>Extension</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Feature</td>
<td>91</td>
<td>99</td>
</tr>
</tbody>
</table>

Training set for tuning ontology: 100
Test set: 116
Car Ads: Comments

• Unbounded sets
  – missed: MERC, Town Car, 98 Royale
  – could use lexicon of makes and models
• Unspecified variation in lexical patterns
  – missed: 5 speed (instead of 5 spd), p.l (instead of p.l.)
  – could adjust lexical patterns
• Misidentification of attributes
  – classified AUTO in AUTO SALES as automatic transmission
  – could adjust exceptions in lexical patterns
• Typographical errors
  – “Chrystler”, “DODG ENeon”, “I-15566-2441”
  – could look for spelling variations and common typos
## Results: Computer Job Ads

*Los Angeles Times*

<table>
<thead>
<tr>
<th></th>
<th>Recall %</th>
<th>Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Skill</td>
<td>74</td>
<td>100</td>
</tr>
<tr>
<td>Email</td>
<td>91</td>
<td>83</td>
</tr>
<tr>
<td>Fax</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Voice</td>
<td>79</td>
<td>92</td>
</tr>
</tbody>
</table>

Training set for tuning ontology: 50
Test set: 50
Our beloved Brian Fielding Frost, age 41, passed away Saturday morning, March 7, 1998, due to injuries sustained in an automobile accident. He was born August 4, 1956 in Salt Lake City, to Donald Fielding and Helen Glade Frost. He married Susan Fox on June 1, 1981. He is survived by Susan; sons Jordan (9), Travis (8), Bryce (6); parents, three brothers, Donald Glade (Lynne), Kenneth Wesley (Ellen), ... Funeral services will be held at 12 noon Friday, March 13, 1998 in the Howard Stake Center, 350 South 1600 East. Friends may call 5-7 p.m. Thursday at Wasatch Lawn Mortuary, 3401 S. Highland Drive, and at the Stake Center from 10:45-11:45 a.m.
Obituary Ontology

(partial)
Data Frames
Lexicons & Specializations

Name matches [80] case sensitive
constant
{ extract First, "\s+", Last; },
...
{ extract "[A-Z][a-zA-Z]*\s+([A-Z]\.|\s+)?", Last; },
...
lexicon
{ First case insensitive; filename "first.dict"; },
{ Last case insensitive; filename "last.dict"; }
end;
Relative Name matches [80] case sensitive
constant
{ extract First, "\s+\(, First, \)\s+", Last; substitute "\s*\([^]\)*\)" -> ""; }
...
end;
...

Keyword Heuristics
Singleton Items

RelativeName|Brian Fielding Frost|16|35
DeceasedName|Brian Fielding Frost|16|35
KEYWORD(Age)|age|38|40
Age|41|42|43
KEYWORD(DeceasedName)|passed away|46|56
KEYWORD(DeathDate)|passed away|46|56
BirthDate|March 7, 1998|76|88
DeathDate|March 7, 1998|76|88
IntermentDate|March 7, 1998|76|98
FuneralDate|March 7, 1998|76|98
ViewingDate|March 7, 1998|76|98
...
Keyword Heuristics
Multiple Items

... KEYWORD(Relationship)|born ... to|152|192
    Relationship|parent|152|192
    KEYWORD(BirthDate)|born|152|156
    BirthDate|August 4, 1956|157|170
    DeathDate|August 4, 1956|157|170
    IntermentDate|August 4, 1956|157|170
    FuneralDate|August 4, 1956|157|170
    ViewingDate|August 4, 1956|157|170
    BirthDate|August 4, 1956|157|170
    RelativeName|Donald Fielding|194|208
    DeceasedName|Donald Fielding|194|208
    RelativeName|Helen Glade Frost|214|230
    DeceasedName|Helen Glade Frost|214|230
KEYWORD(Relationship)|married|237|243
...
Results: Obituaries

*Arizona Daily Star*

<table>
<thead>
<tr>
<th></th>
<th>Recall %</th>
<th>Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeceasedName*</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Age</td>
<td>86</td>
<td>98</td>
</tr>
<tr>
<td>BirthDate</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>DeathDate</td>
<td>84</td>
<td>99</td>
</tr>
<tr>
<td>FuneralDate</td>
<td>96</td>
<td>93</td>
</tr>
<tr>
<td>FuneralAddress</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>FuneralTime</td>
<td>92</td>
<td>87</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relationship</td>
<td>92</td>
<td>97</td>
</tr>
<tr>
<td>RelativeName*</td>
<td>95</td>
<td>74</td>
</tr>
</tbody>
</table>

*partial or full name

Training set for tuning ontology: ~ 24
Test set: 90
Results: Obituaries

*Salt Lake Tribune*

<table>
<thead>
<tr>
<th>Field</th>
<th>Recall %</th>
<th>Precision %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeceasedName*</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Age</td>
<td>91</td>
<td>95</td>
</tr>
<tr>
<td>BirthDate</td>
<td>100</td>
<td>97</td>
</tr>
<tr>
<td>DeathDate</td>
<td>94</td>
<td>100</td>
</tr>
<tr>
<td>FuneralDate</td>
<td>92</td>
<td>100</td>
</tr>
<tr>
<td>FuneralAddress</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>FuneralTime</td>
<td>97</td>
<td>100</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relationship</td>
<td>81</td>
<td>93</td>
</tr>
<tr>
<td>RelativeName*</td>
<td>88</td>
<td>71</td>
</tr>
</tbody>
</table>

*partial or full name

Training set for tuning ontology: ~ 12
Test set: 38
“Open” Problems

- Record-Boundary Detection
- Record Reconfiguration
- Page/Ontology Matching
- Form Interfaces
- Rapid Ontology Construction, Evolution, and Improvement
Record-Boundary Detection: High Fan-Out Heuristic

<html>
<head>
<title>The Salt Lake Tribune ... </title>
</head>
<body bgcolor="#FFFFFF">
<h1 align="left">Domestic Cars</h1>
...
<hr>
<h4> '97 CHEVY Cavalier, Red, ... </h4>
<hr>
<h4> '89 CHEVY Corsica Sdn ... </h4>
<hr>
...
</body>
</html>
Record-Boundary Detection: Record-Separator Heuristics

- Identifiable “separator” tags
- Highest-count tag(s)
- Interval standard deviation
- Ontological match
- Repeating tag patterns

Example:

```html
<hr>
<h4> '97 CHEVY Cavalier, Red, 5 spd, <i>only 7,000 miles</i> on her. Asking <i>only $11,995</i>. … </h4>
<hr>
<h4> '89 CHEV Corsica Sdn teal, auto, air, <i>trouble free</i>. Only $8,995 </h4>
<hr>
```

...
Record-Boundary Detection: Consensus Heuristic

Certainty is a generalization of: $C(E_1) + C(E_2) - C(E_1)C(E_2)$. $C$ denotes certainty and $E_i$ is the evidence for an observation.

Our certainties are based on observations from 10 different sites for 2 different applications (car ads and obituaries)

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Correct Tag Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>IT</td>
<td>96%</td>
</tr>
<tr>
<td>HT</td>
<td>49%</td>
</tr>
<tr>
<td>SD</td>
<td>66%</td>
</tr>
<tr>
<td>OM</td>
<td>85%</td>
</tr>
<tr>
<td>RP</td>
<td>78%</td>
</tr>
</tbody>
</table>
Record-Boundary Detection: Results

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT</td>
<td>95%</td>
</tr>
<tr>
<td>HT</td>
<td>45%</td>
</tr>
<tr>
<td>SD</td>
<td>65%</td>
</tr>
<tr>
<td>OM</td>
<td>80%</td>
</tr>
<tr>
<td>RP</td>
<td>75%</td>
</tr>
<tr>
<td>Consensus</td>
<td>100%</td>
</tr>
</tbody>
</table>

4 different applications (car ads, job ads, obituaries, university courses) with 5 new/different sites for each application
Record Reconfiguration: Problems Encountered

factored

joined

interspersed

off-page

split
Record Reconfiguration: Proposed Solution

- Maximize a Record-Recognition Measure
- Improvements:
  - Split joined records
  - Distribute factored values
  - Link off-page information
  - Join split records
  - Discard interspersed records
Record Reconfiguration: Use Record-Recognition Measures based on Vector Space Modeling

VSM

Ontology Vector $<f_1, \ldots, f_n>$

Document Vector $<f_1, \ldots, f_n>$

VSM Measures

Cosine

Vector Length
Record Reconfiguration:
Test Set Characteristics

- 30 pre-selected documents
- Characteristics
  - 8 contained only regular car ads.
  - 13 contained inside-boundary joined car ads all with inside-boundary factored values.
  - 1 contained outside-boundary factored values.
  - 13 contained interspersed non-car-ads.
  - None contained off-page or split ads.
Record Reconfiguration: Results

- Correctly reconfigured 91% (of 304)
  - 36 false drops
  - 11 car ads improperly discarded (value-recognition problem)
  - 25 car ads improperly reconfigured
    - 20 ads with identical phone numbers on every 5th
    - 5 inside-boundary ads not split (missing years & makes; not all models recognized)

- Correctly discarded 94% (of 47)
  - 3 false positives
  - All snowmobile ads

- Correctly produced 97% (of 1,077)
Page/Ontology Matching: Recognition Heuristics

• Density Heuristic
  – Lots of constant and keyword matches.
  – Total matched characters / total characters

• Expected-Values Heuristic
  – Recognized constants appear in expected frequencies.
  – VSM cosine measure

• Grouping Heuristic
  – Recognized constants are grouped as expected.
  – Number of distinct “one-max” values: ordered & grouped by expected size of group
Page/Ontology Matching: Machine-Learned Combined Heuristic

- A heuristic triple \((H_1, H_2, H_3)\) represents a document.
- Training set: 20 positive examples; 30 negative.
- C4.5 machine-learning algorithm produced decision trees.

```
Car Ads
H_2 >0.88
  no  yes

Obituary
H_2 >0.68
  no
H_1 >0.22
  no
  yes

Universal
H_3 >0.63
  yes
H_1 >0.37
  no
H_2 >0.63
  no
  yes
```
Page/Ontology Matching: Results

• Car Ads
  – Rule 1 correctly matched 97% (of 11 positive and 19 negative)
  – One false negative: “ROBERTS FORDCHRYSLERPLYMOUTHJEEPUSED CARS99
     Plymouth Breeze $12,99599 Plymouth Neon $11,99599
     Ford … theHomer Adams Pkwy., Alton466-7220”

• Obituaries
  – Rule 2 correctly matched 97% (of 10 positive and 20 negative)
  – One false positive: Missing People

• Singleton Obituaries – marginal for famous people
Form Interfaces
Form Interfaces: Questions

• What’s the best way to automate retrieval of data behind Web forms?
  – Can we match a form to an ontology?
  – Can we learn how to fill in a form for a given (ontology) query?
  – Is it reasonable to try to retrieve all the data behind a form?
• Can we automatically:
  – Fill in Web forms?
  – Extract information behind forms?
  – Screen out error messages and inapplicable Web pages?
  – Eliminate duplicate data?
Rapid Ontology Construction, Evolution, and Improvement

• Is it possible to (semi)automate the construction of an application ontology?
  – Can we build tools to help users create an ontology? (yes)
  – Can we assemble components from a knowledgebase of ontological components?
  – Can we use machine learning?

• Can we use automated construction techniques to aid in ontology evolution and improvement?
Conclusions

• Given an ontology and a Web page with multiple records, it is possible to extract and structure the data automatically.
• Recall and Precision results are encouraging.
  – Car Ads: ~ 94% recall and ~ 99% precision
  – Job Ads: ~ 84% recall and ~ 98% precision
  – Obituaries: ~ 90% recall and ~ 95% precision (except on names: ~ 73% precision)
• Resolution of Problems
  – Record-Boundary Detection – excellent if records nicely separated
  – Record Reconfiguration – excellent for known patterns
  – Page/Ontology matching – excellent for multiple-record documents
• Open Problems
  – Extraction of data behind forms
  – Rapid ontology construction

http://www.deg.byu.edu/