Web-Based Learning

Craig A. Knoblock
University of Southern California

Joint work with
J. L. Ambite, K. Lerman, A. Plangprasopchok, and T. Russ, USC
C. Gazen and S. Minton, Fetch Technologies
M. Carman, University of Lugano
Introduction

• Problem
  • Web sources and services are designed for people, not machines
  • Limited or no description of the information provided by these sources
  • This makes it hard, if not impossible to find, retrieve and integrate the vast amount of structured data available
    • Weather sources, geocoders, stock information, currency converters, online stores, etc.

• Approach
  • Start with an some initial knowledge of a domain
    • Sources and semantic descriptions of those sources
  • Automatically
    • Discover related sources
    • Learn the syntactic structure of the sources
    • Build semantic models of the source
    • Validate the correctness of the results
Automatically Discover and Model a Source in the Same Domain
Approach

discovery

unisys

invocation &
extraction

Background knowledge

http://wunderground.com

unisys(Zip,Temp,…) :- weather(Zip, Temp, Hi, Lo)

• Seed URL

source modeling

semantic typing

unisys(Zip,Temp,…) unisys(Zip,Temp,Humidity,…) • sample input values • sample values • definition of known sources • patterns • domain types
• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
“Black” + “Jaguar” = ?

- Animal
- Car
Goal

Grouping semantically related tags and content

A group ~ A concept
A stochastic process of tag generation

PLSA (Hofmann99);
LDA (Blei03+)

A data point (tuple) <r, t, z>
Exploiting Social Annotations for Resource Discovery

- Simplified resource discovery task: "given a seed source, find other most similar sources"
Outline

• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
Discovering Web Structure

• **Goal:**
  • Model Web sources that generate pages dynamically in response to a query
    – Find the relational data underlying a semi-structured web site
  • **Generate a page template that can be used to extract data on new pages**

• **Approach**
  • **Site extraction**
    – Exploit the common structure within a web site
  • Take advantage of multiple structures
    – HTML structure, page layout, links, data formats, etc.
Overview

Web Site

Page & Data Hypotheses

Page & Data Clusters

Site and Page Structure

Homepage

AutoFeedWeather

States

<table>
<thead>
<tr>
<th>0</th>
<th>California</th>
<th>CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pennsylvania</td>
<td>PA</td>
</tr>
</tbody>
</table>

CityWeather

<table>
<thead>
<tr>
<th>0</th>
<th>Los Angeles</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>San Francisco</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>San Diego</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>Pittsburgh</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>Philadelphia</td>
<td>55</td>
</tr>
</tbody>
</table>
• Page Templates
  • Similar pages contain common sequences of substrings

• HTML Structure
  • List rows are represented as repeating HTML structures
Extracting Data

Pages

Extracted Data

<table>
<thead>
<tr>
<th></th>
<th>FRIDAY</th>
<th>SATURDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>Sunny</td>
<td>Sun</td>
</tr>
<tr>
<td>Hi: 65</td>
<td>LO: 52</td>
<td>Hi: 60</td>
</tr>
<tr>
<td>Rain</td>
<td>Rainy</td>
<td>LO: 48</td>
</tr>
</tbody>
</table>

Hypotheses

- **group_member** (FRIDAY, SATURDAY)
- **group_member** (Sunny, Rainy)
- **same_html_context** (65, 60)
- **vertically_aligned** (Sun, Rain)
- **two_digit_number** (65, 52, 60, 48)
- ...

Clusters

- FRIDAY 65 52
- SATURDAY 60 48
• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
Learning Patterns to Recognize Semantic Types

- Domain-independent token-level language to represent the structure of data as patterns
  - Token is a string or a general type
    - 90202 is a specific token
    - 5DIGIT number is a general type
  - Pattern is a sequence of tokens
    - E.g., Phone numbers

<table>
<thead>
<tr>
<th>Sample values</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>310 448–8714</td>
<td>[ 310 448 – 4DIGIT]</td>
</tr>
<tr>
<td>310 448–8775</td>
<td></td>
</tr>
<tr>
<td>212 555–1212</td>
<td>[ 3DIGIT 3DIGIT – 4DIGIT]</td>
</tr>
</tbody>
</table>

- Efficiently learn patterns from examples of semantic types
- Score the match between a type (patterns) and data
Weather Data Types

Sample values
• PR-TempF
  88 F
  57°F
  82 F ...

• PR-Visibility
  8.0 miles
  10.0 miles
  4.0 miles
  7.00 mi
  10.00 mi

• PR-Zip
  07036
  97459
  02102

Patterns
• PR-TempF
  [88, F]
  [2DIGIT, F]
  [2DIGIT, °, F]

• PR-Visibility
  [10, ., 0, miles]
  [10, ., 00, mi]
  [10, ., 00, mi, .]
  [1DIGIT, ., 00, mi]
  [1DIGIT, ., 0, miles]

• PR-Zip
  [5DIGIT]
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Score</th>
<th>Values</th>
<th>Score</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>PR-Zip</td>
<td>0.333</td>
<td>20502</td>
<td>0.68</td>
<td>45F</td>
</tr>
<tr>
<td>18</td>
<td>PR-TempF</td>
<td>0.68</td>
<td>40% Partly Cloudy</td>
<td>1.0</td>
<td>40% Partly Cloudy</td>
</tr>
<tr>
<td>25</td>
<td>PR-Humidity</td>
<td>1.0</td>
<td>32399</td>
<td>0.325</td>
<td>63F Sunny</td>
</tr>
<tr>
<td>15</td>
<td>PR-Sky</td>
<td>0.325</td>
<td>Partly Cloudy</td>
<td>0.375</td>
<td>Sunny</td>
</tr>
<tr>
<td>87</td>
<td>PR-Sky</td>
<td>0.375</td>
<td>Sunny</td>
<td>Partly Cloudy</td>
<td>Rainy</td>
</tr>
<tr>
<td>33040</td>
<td>63F</td>
<td>0.73</td>
<td>73F Sunny</td>
<td>Rainy</td>
<td>Rainy</td>
</tr>
<tr>
<td>90292</td>
<td>66F</td>
<td>0.59</td>
<td>Partly Cloudy</td>
<td>Sunny</td>
<td>Sunny</td>
</tr>
<tr>
<td>36130</td>
<td>62F</td>
<td>0.24</td>
<td>Sunny</td>
<td>Partly Cloudy</td>
<td>Partly Cloudy</td>
</tr>
</tbody>
</table>
Outline

• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).
source1($zip, lat, long) :-
   centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
   greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
   convertKm2Mi(dist1, dist2).

source4( $startZip, $endZip, separation)
Inducing Source Definitions

- Step 1: classify input & output semantic types

source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

source4($startZip, $endZip, separation)
Inducing Source Definitions

- Step 1: classify input & output semantic types

source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

$zip

source4( $startZip, $endZip, separation)
Inducing Source Definitions

- Step 1: classify input & output semantic types

source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

source4( $startZip, $endZip, separation)
Generating Plausible Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions

```
source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

source4( $zip1, $zip2, dist)
```
Generating Plausible Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions

source1($zip, lat, long) :-
centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
convertKm2Mi(dist1, dist2).

source4($zip1, $zip2, dist) :-
source1(zip1, lat1, long1),
source1(zip2, lat2, long2),
source2(lat1, long1, lat2, long2, dist2),
source3(dist2, dist).
Generating Plausible Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions

source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

source4($zip1, $zip2, dist):-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).

source4($zip1, $zip2, dist):-
    centroid(zip1, lat1, long1),
    centroid(zip2, lat2, long2),
    greatCircleDist(lat1, long1, lat2, long2, dist2),
    convertKm2Mi(dist1, dist2).
 Invoke and Compare the Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions
- Step 3: invoke service & compare output

<table>
<thead>
<tr>
<th>$zip1</th>
<th>$zip2</th>
<th>dist (actual)</th>
<th>dist (predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80210</td>
<td>90266</td>
<td>842.37</td>
<td>843.65</td>
</tr>
<tr>
<td>60601</td>
<td>15201</td>
<td>410.31</td>
<td>410.83</td>
</tr>
<tr>
<td>10005</td>
<td>35555</td>
<td>899.50</td>
<td>899.21</td>
</tr>
</tbody>
</table>

source4($zip1, $zip2, dist):-
source1(zip1, lat1, long1),
source1(zip2, lat2, long2),
source2(lat1, long1, lat2, long2, dist2),
source3(dist2, dist).

match

source4($zip1, $zip2, dist):-
centroid(zip1, lat1, long1),
centroid(zip2, lat2, long2),
greatCircleDist(lat1, long1, lat2, long2,dist2),
convertKm2Mi(dist1, dist2).
• Given a set of known sources and their descriptions
  • `wunderground($Z,CS,T,F0,S0,Hu0,WS0,WD0,P0,V0) :- weather(0,Z,CS,D,T,F0,_,_,S0,Hu0,P0,WS0,WD0,V0)``
  • `convertC2F(C,F) :- centigrade2farenheit(C,F)``
• Learn a description of a new source in terms of the known sources
  • `unisys($Z,CS,T,F0,C0,S0,Hu0,WS0,WD0,P0,V0) :- wunderground(Z,CS,T,F0,S0,Hu0,WS0,WD0,P0,V0), convertC2F(C0,F0)``
Outline

- Discovering sources using social annotations
- Discovering the structure of sources
- Learning semantic types of the source data
- Learning semantic models of the sources
- Experimental Results
- Discussion
Experimental Evaluation

- Experiments in 3 domains
  - Geospatial
    - *Geocoder that maps street addresses into lat/long coordinates*
  - Weather
    - *Produces current and forecasted weather*
  - Flight Status
    - *Current status for a given airline and flight*

- Evaluation:
  - 1) Can we correctly learn a model for those sources that perform the same task
  - 2) What is the precision and recall of the attributes in the model
Candidate Sources after Each Step

URL Filtering by Module

Number of URLs

<table>
<thead>
<tr>
<th>Step</th>
<th>Flight</th>
<th>Geospatial</th>
<th>Weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Invocation</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Source Typing</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Source Modeling</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
# Evaluation of the Models

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>geospatial</td>
<td>86</td>
<td>100</td>
<td>92</td>
</tr>
<tr>
<td>weather</td>
<td>29</td>
<td>64</td>
<td>39</td>
</tr>
<tr>
<td>flight</td>
<td>35</td>
<td>69</td>
<td>46</td>
</tr>
</tbody>
</table>
• Discovering sources using social annotations
• Discovering the structure of sources
• Learning semantic types of the source data
• Learning semantic models of the sources
• Experimental Results
• Discussion
Related Work

- **ILA & Category Translation** (Perkowitz & Etzioni 1995)
  - Learn functions describing operations on internet
- **iMAP** (Dhamanka et al. 2004)
  - Discovers complex (many-to-1) mappings between DB schemas
- **Metadata-based classification of data types used by Web services and HTML forms** (Hess & Kushmerick, 2003)
  - Naïve Bayes classifier
- **Woogle**: Metadata-based clustering of data and operations used by Web services (Dong et al, 2004)
  - Groups similar types together: Zipcode, City, State
• Integrated a diverse set of learning and reasoning techniques
  • *Discover new sources*
  • *Discover the template for a source*
  • *Find the semantic types of source data*
  • *Learn a definition of what a source does*

• Provides an end-to-end completely automatic approach to discover and build models of sources