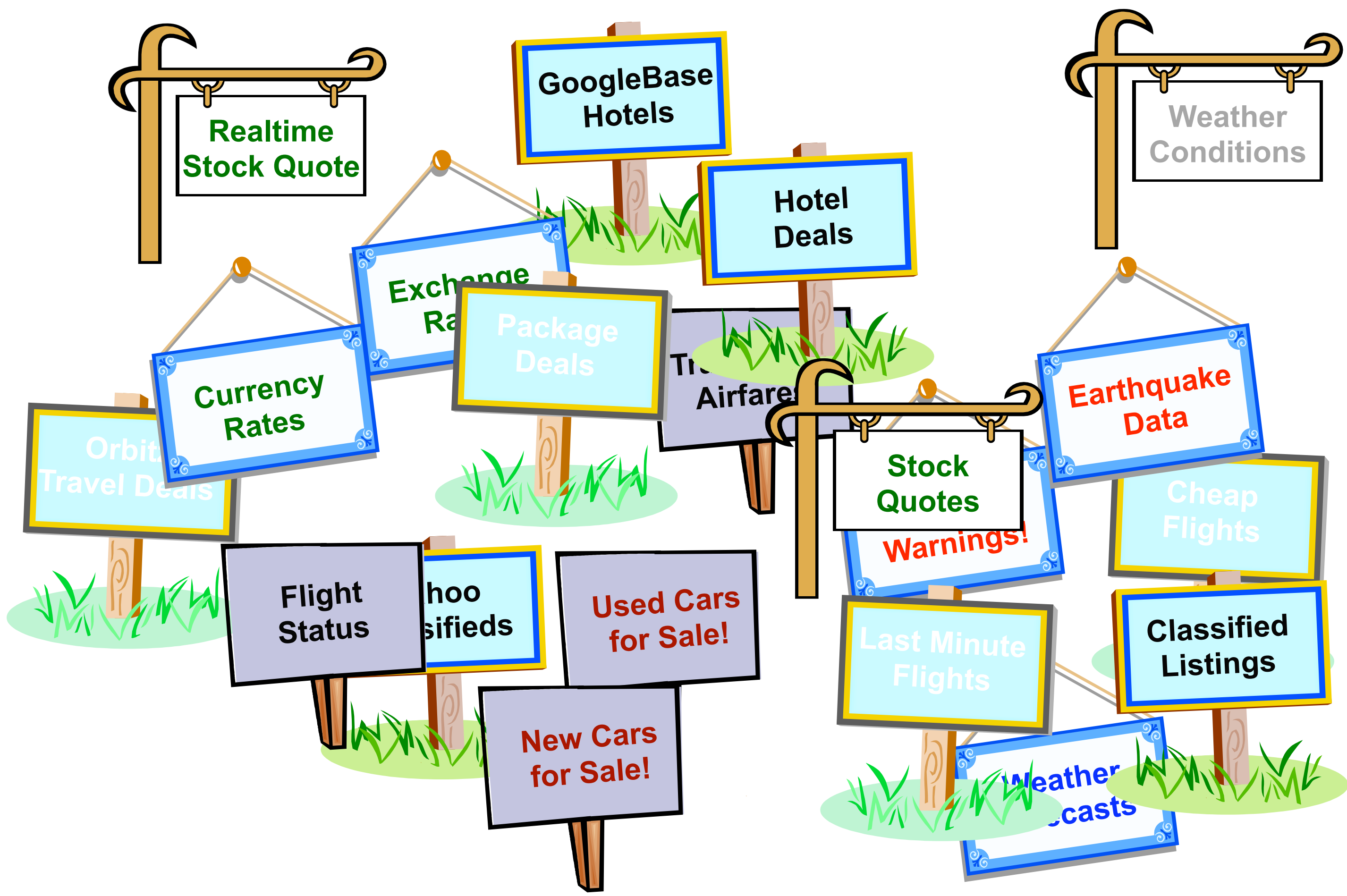
Automatically Discovering, Extracting and Modeling Web Sources for Information Integration

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*

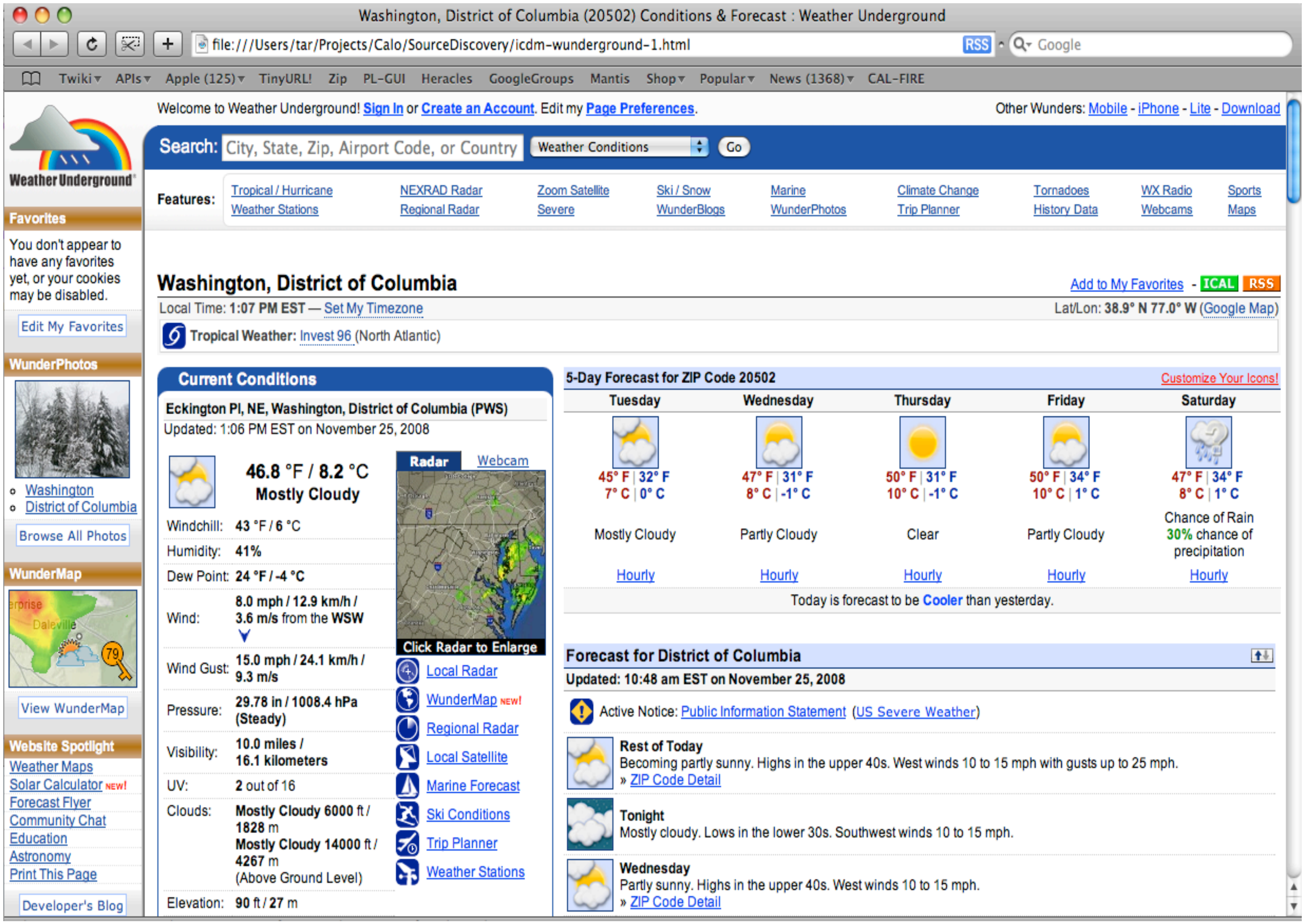
Abundance of Data, Limited Knowledge



Motivation

- 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*
    - *Determine how to invoke the sources*
    - *Learn the syntactic structure of the sources*
    - *Build semantic models of the source*
    - *Validate the correctness of the results*

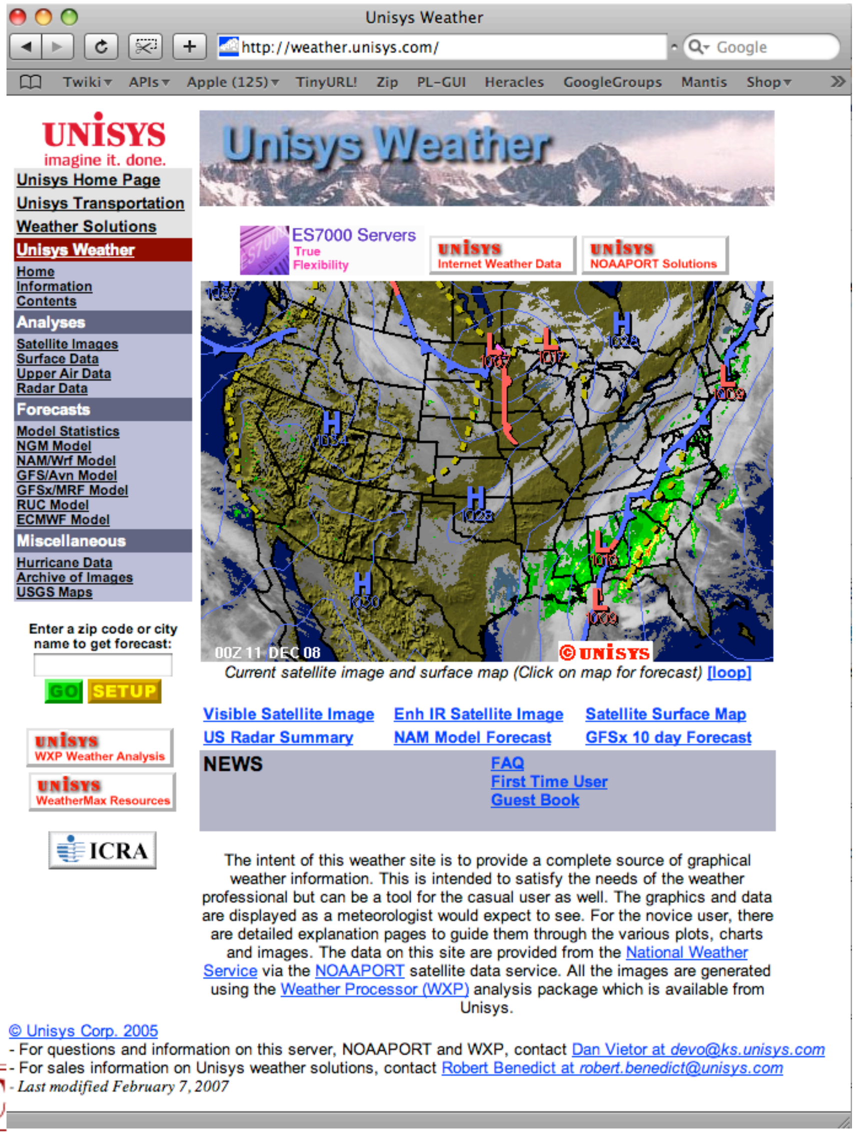
Seed Source

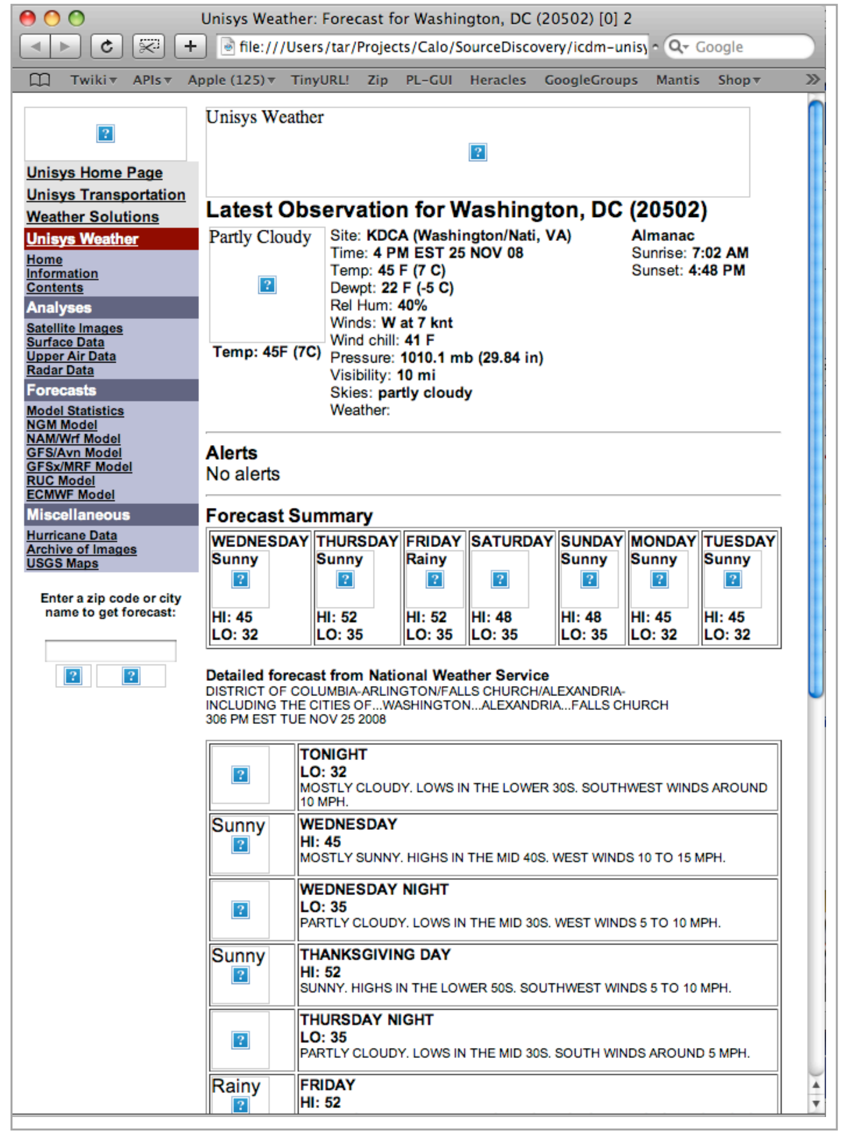


Washington, District of Columbia (20502) Conditions & Forecast : Weather Underground
file:///Users/tar/Projects/Calo/SourceDiscovery/icdm-wunderground-1.html
Welcome to Weather Underground! Sign In or Create an Account. Edit my Page Preferences.
Search:
Features:
Washington, District of Columbia
Local Time: 1:07 PM EST — Set My Timezone
Tropical Weather: Invest 96 (North Atlantic)
Current Conditions
Eckington Pl, NE, Washington, District of Columbia (PWS)
Updated: 1:06 PM EST on November 25, 2008
46.8 °F / 8.2 °C
Mostly Cloudy
Windchill: 43 °F / 6 °C
Humidity: 41%
Dew Point: 24 °F / -4 °C
Wind: 8.0 mph / 12.9 km/h / 3.6 m/s from the WSW
Wind Gust: 15.0 mph / 24.1 km/h / 9.3 m/s
Pressure: 29.78 in / 1008.4 hPa (Steady)
Visibility: 10.0 miles / 16.1 kilometers
UV: 2 out of 16
Clouds: Mostly Cloudy 6000 ft / 1828 m Mostly Cloudy 14000 ft / 4267 m (Above Ground Level)
Elevation: 90 ft / 27 m
5-Day Forecast for ZIP Code 20502
Tuesday 45° F | 32° F 7° C | 0° C Mostly Cloudy
Wednesday 47° F | 31° F 8° C | -1° C Partly Cloudy
Thursday 50° F | 31° F 10° C | -1° C Clear
Friday 50° F | 34° F 10° C | 1° C Partly Cloudy
Saturday 47° F | 34° F 8° C | 1° C Chance of Rain 30% chance of precipitation
Today is forecast to be Cooler than yesterday.
Forecast for District of Columbia
Updated: 10:48 am EST on November 25, 2008
Rest of Today
Becoming partly sunny. Highs in the upper 40s. West winds 10 to 15 mph with gusts up to 25 mph.
Tonight
Mostly cloudy. Lows in the lower 30s. Southwest winds 10 to 15 mph.
Wednesday
Partly sunny. Highs in the upper 40s. West winds 10 to 15 mph.

Automatically Discover and Model a Source in the Same Domain

USC Viterbi
School of Engineering

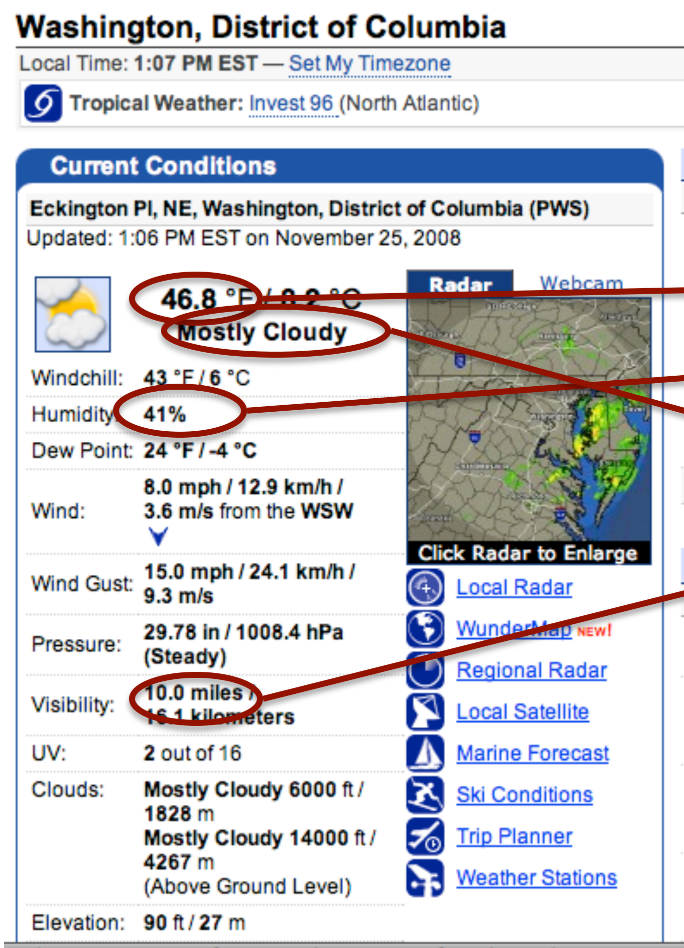




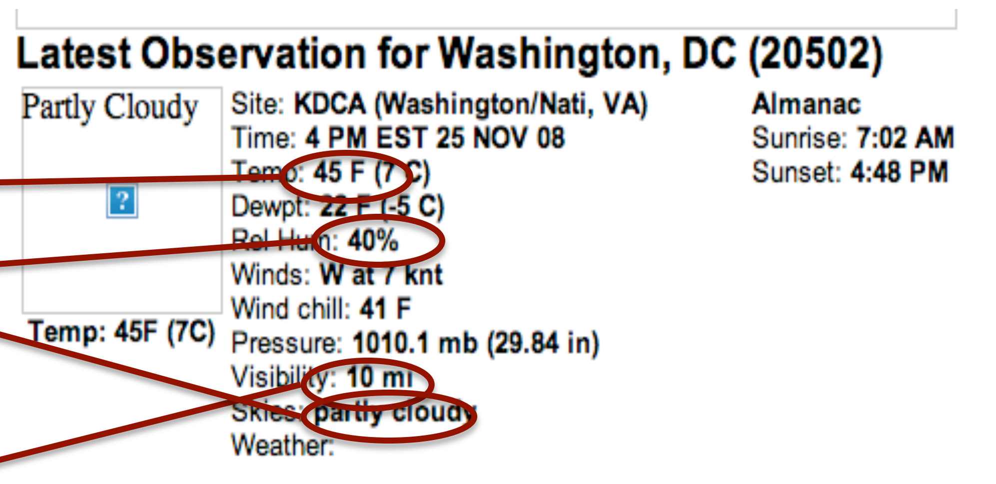
Current Conditions Data

USC Viterbi School of Engineering

Seed (wunderground.com)



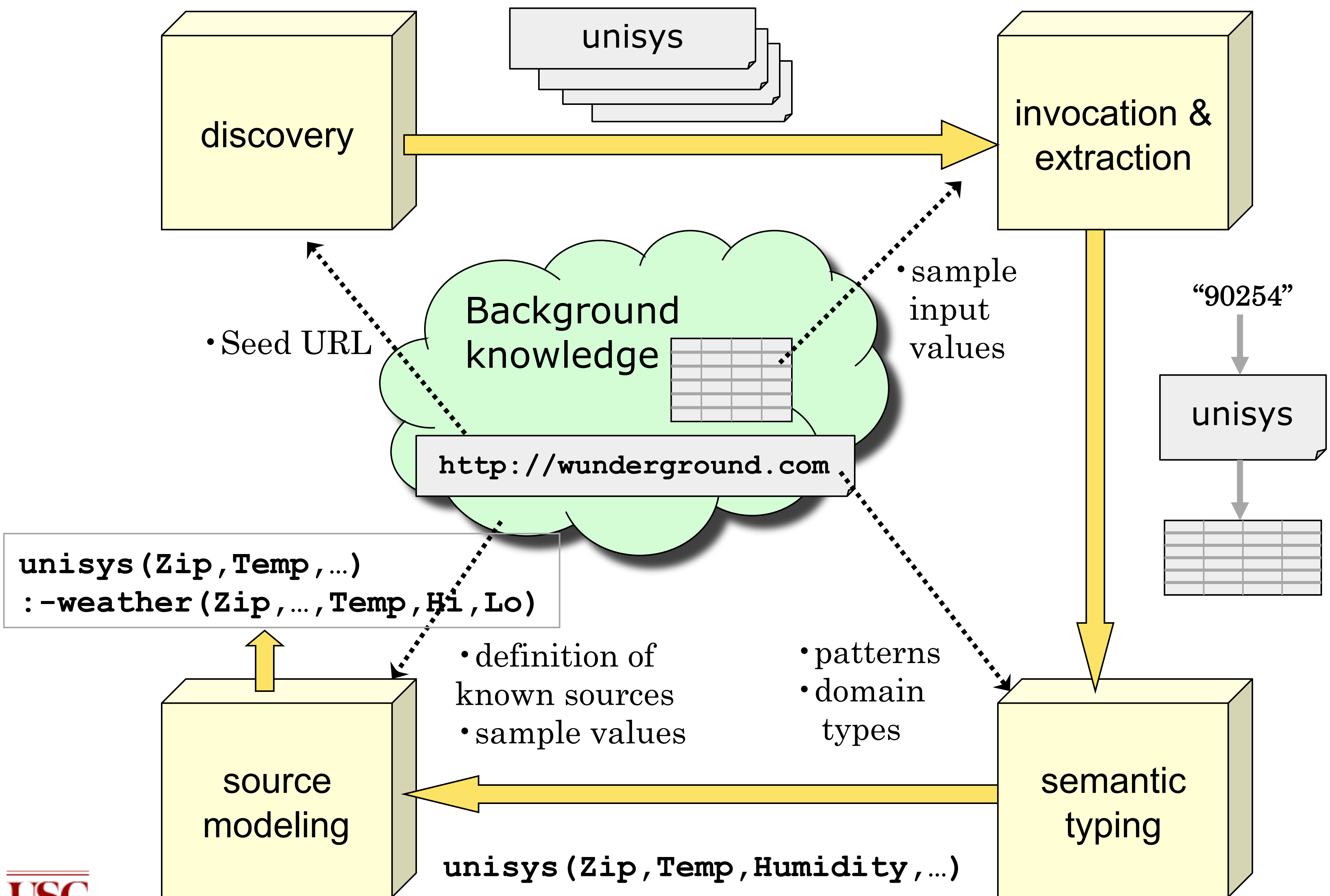
Target (unisys.com)



Partial Mapping of Values

USC

Approach


discovery
unisys
invocation & extraction
Background knowledge
http://wunderground.com
• Seed URL
• sample input values
"90254"
unisys
unisys(Zip,Temp,…)
:-weather(Zip,…,Temp,Hi,Lo)
• definition of known sources
• sample values
• patterns
• domain types
source modeling
semantic typing
unisys(Zip,Temp,Humidity,…)

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

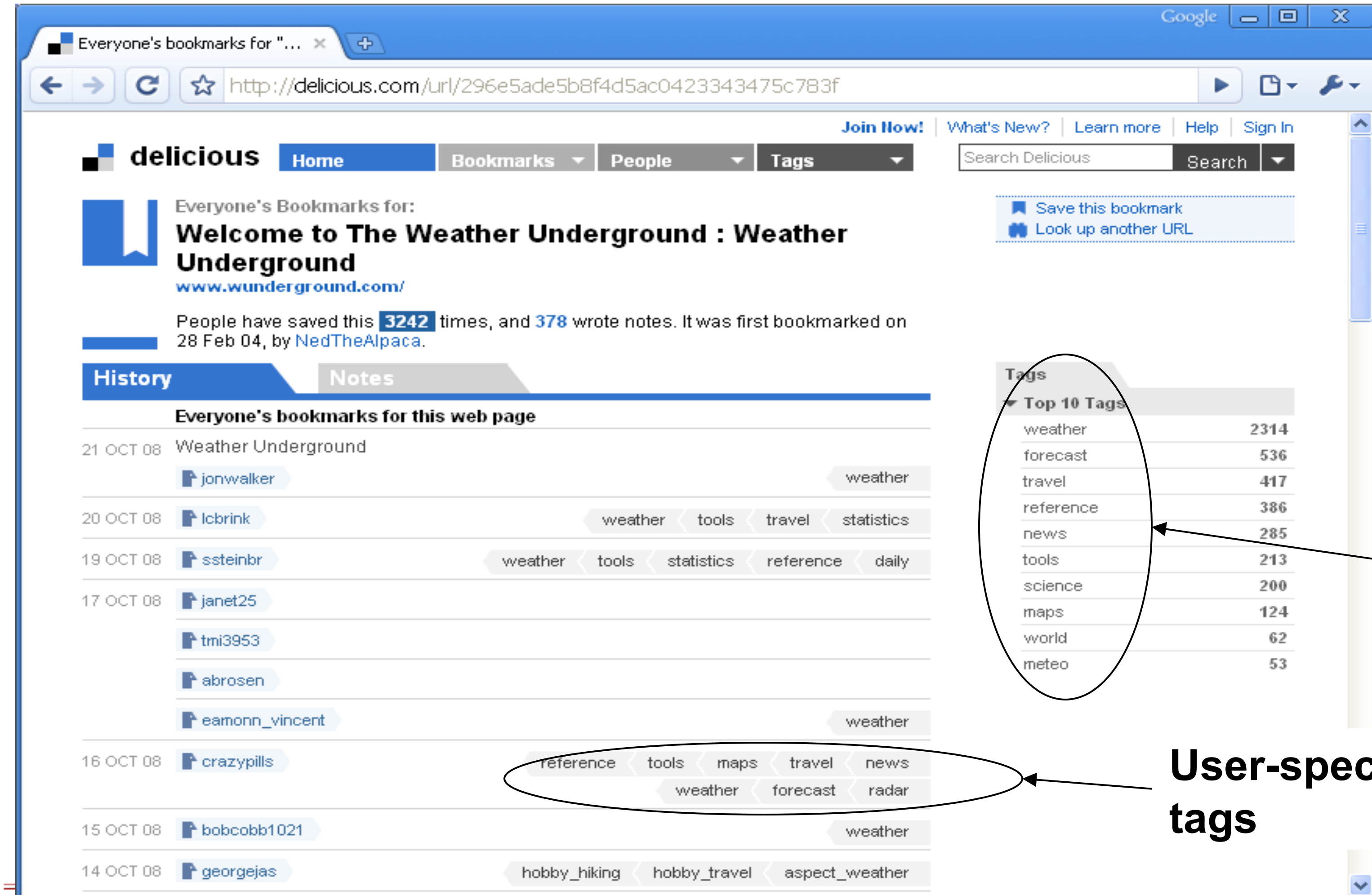
Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Source Discovery

USC Viterbi School of Engineering

- Sources providing similar functionality are annotated with "similar" tags on the social bookmarking site del.icio.us

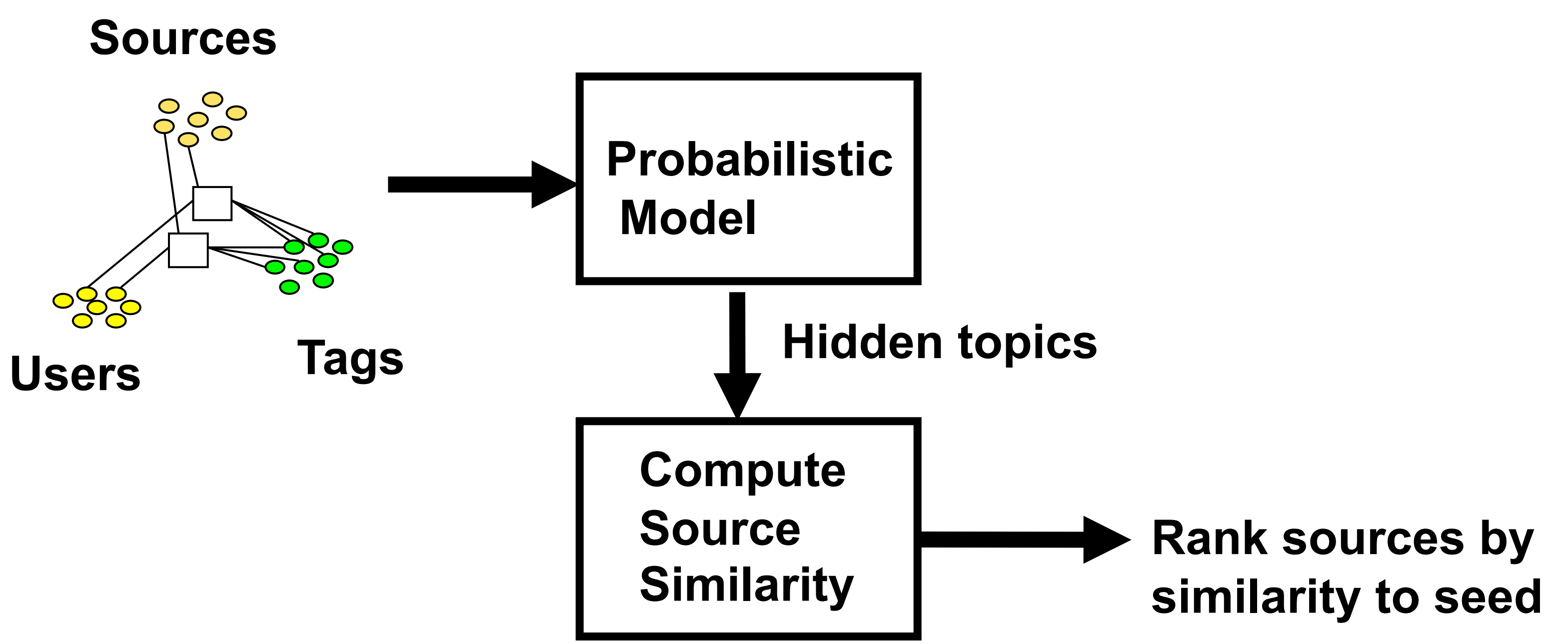


Most common tags

User-specified tags

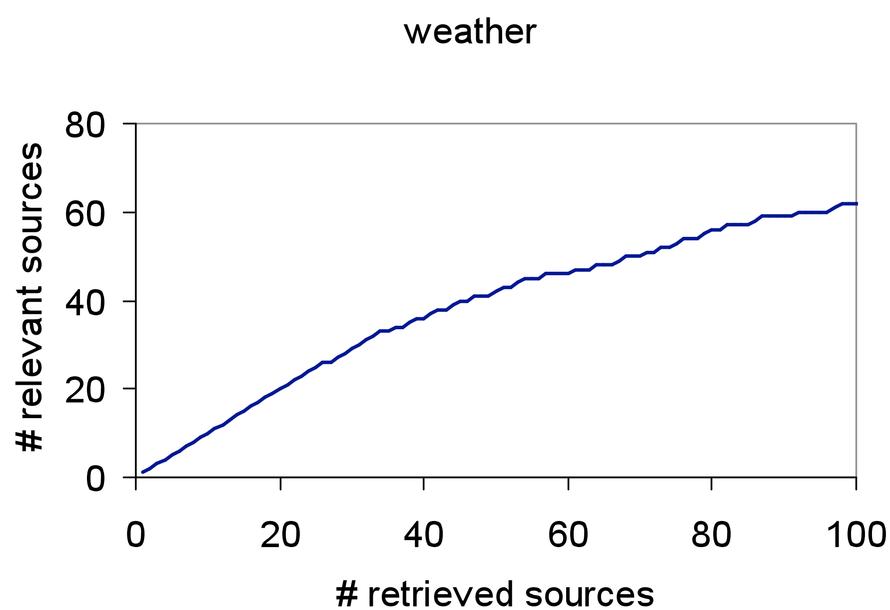
USC

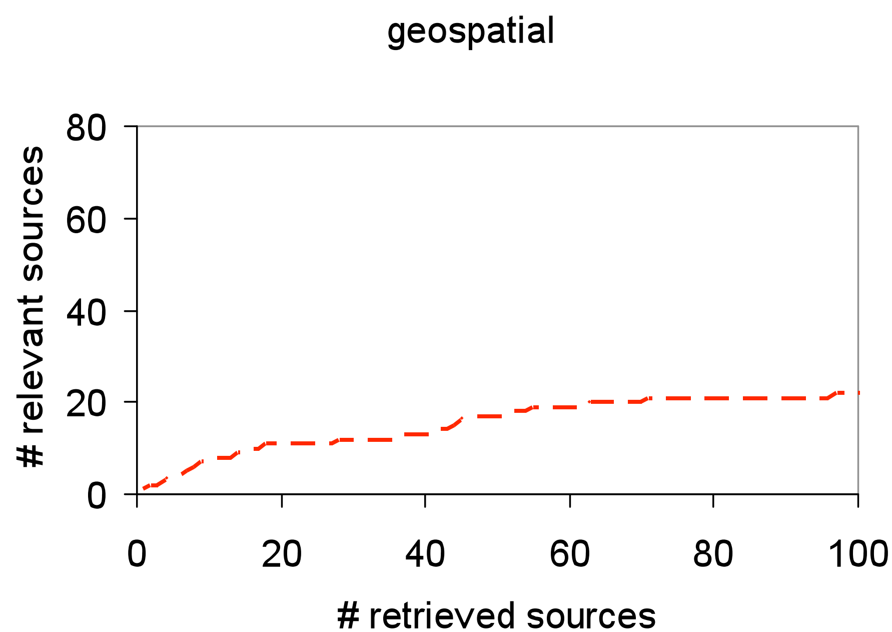
- Goal
  - Leverage user-generated tags on the social bookmarking site del.icio.us to discover sources similar to the seed
- Approach
  - Gather a corpus of <user, source, tag> bookmarks from del.icio.us
  - Use probabilistic modeling to find hidden topics in the corpus
  - Rank sources by similarity to the seed within topic space



Source Discovery Results

- Manually evaluated the top-ranked 100 sources
  - Number of relevant sources providing same functionality as the seed
    - *Weather domain: weather conditions (wunderground seed)*
    - *Geospatial domain: geocodes of addresses (geocode.us seed)*





The top-ranked 100 sources become the *target sources* we will try to model

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

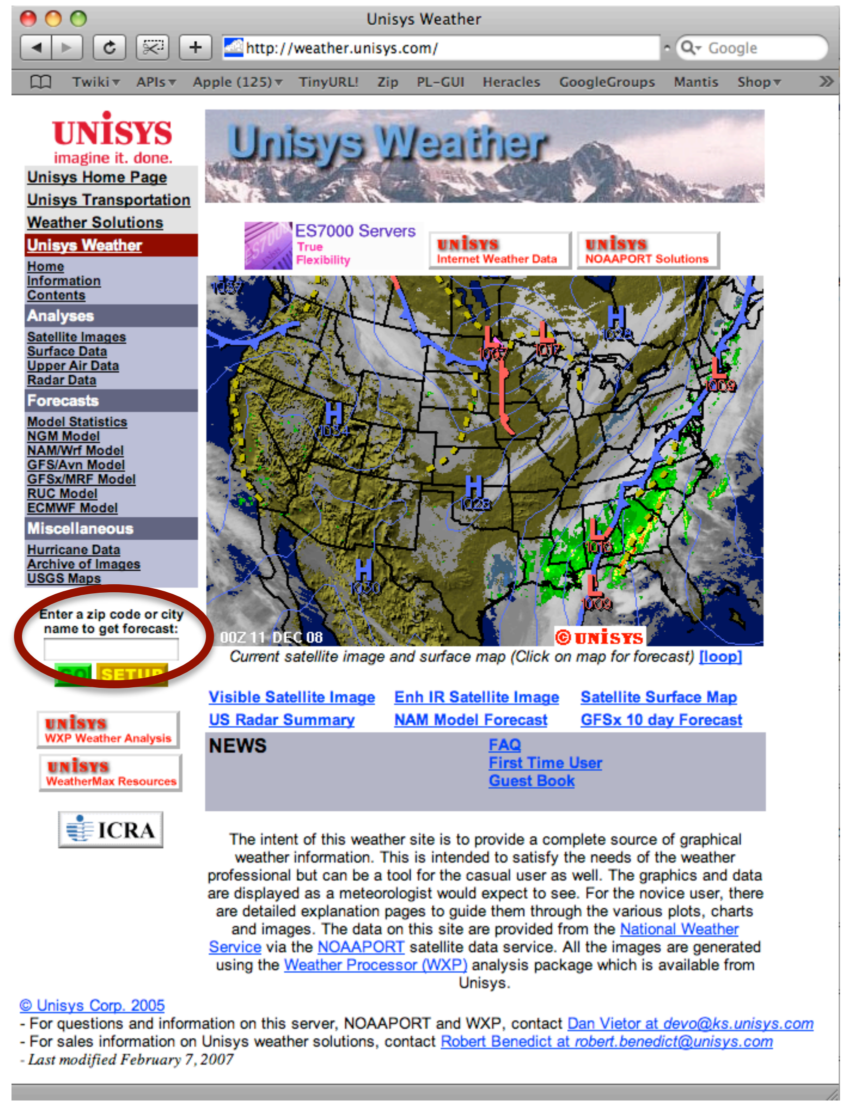
Target Source Invocation

USC Viterbi School of Engineering

- To invoke the target source, we need to locate the form and submit it with appropriate input values
  1. Locate the form
  2. Try different data type combinations as input
     - *For weather, only one input - location, which can be zipcode or city*
  3. Submit Form

  4. Keep successful invocations

Form Input

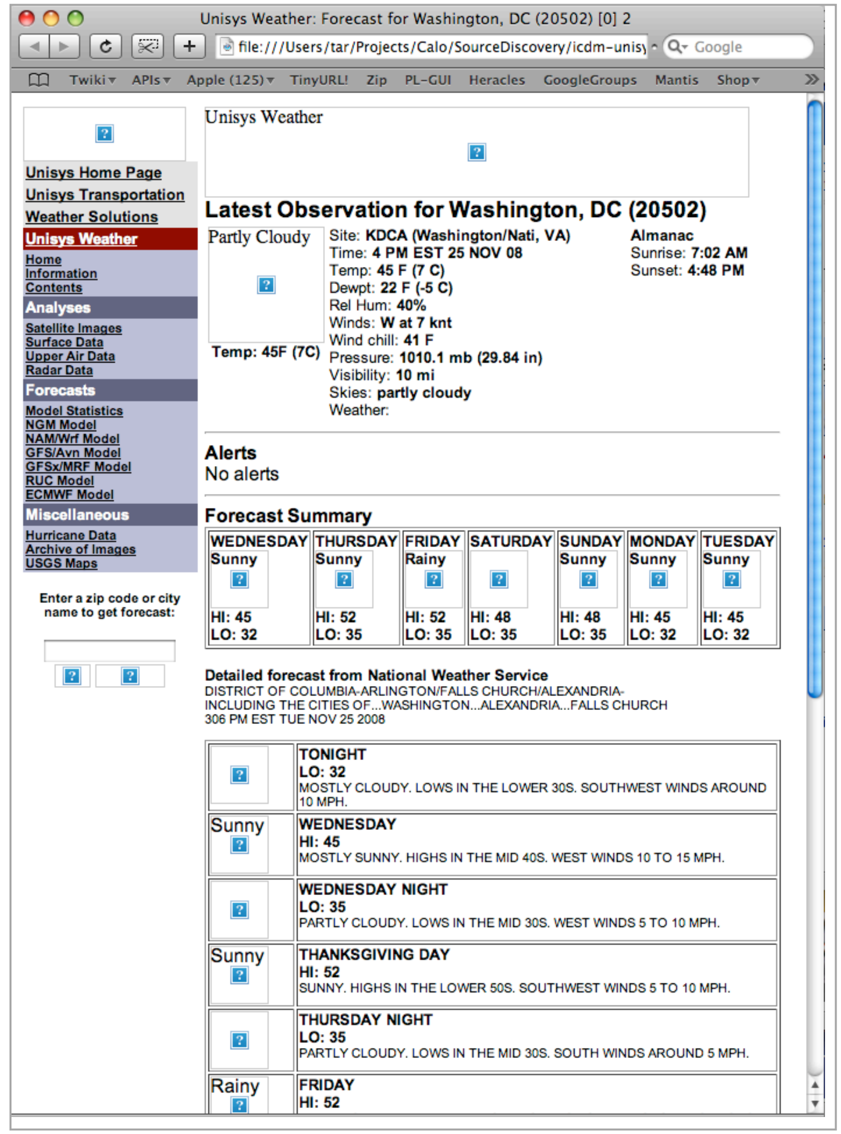


Invoke the Target Source with Possible Inputs

http://weather.unisys.com

input

Weather conditions for 20502



Form Input Data Model

- Each domain has an input data model
  - Derived from the seed sources
  - Alternate input groups
- Each domain has sample values for the input data types

domain name="weather

- input “zipcode” type PR-Zip
- input “cityState” type PR-CityState
- input “city” type PR-City
- input “stateAbbr” type PR-StateAbbr

PR-Zip	PR-CityState	PR-City	PR-StateAbbr
20502	Washington, DC	Washington	DC
32399	Tallahassee, FL	Tallahassee	FL
33040	Key West, FL	Key West	FL
90292	Marina del Rey, CA	Marina del Rey	CA
36130	Montgomery, AL	Montgomery	AL

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Constructing Syntactic Models of the Sources

- Goal:
  - Model Web sources that generate pages dynamically in response to a query
- Approach:
  - Given two or more sample pages, derive the page template
  - Use the template to extract data from the pages

Inducing Templates

- Template: a sequence of alternating slots and stripes
  - stripes are the common substrings among all pages
  - slots are the placeholders for data
- Induction: Stripes are discovered using the Longest Common Subsequence algorithm

Sample Page 1

```
<img src="images/Sun.png" alt="Sunny"><br>
<font face="Arial, Helvetica, sans-serif">
 <small><b>Temp: 72F (22C)</b></small></font>
<font face="Arial, Helvetica, sans-serif">
 <small>Site: <b>KSMO (Santa_Monica_Mu, CA)</b><br>
     Time: <b>11 AM PST 10 DEC 08</b>
```

Sample Page 2

```
<img src="images/Clouds.png" alt="Cloudy"><br>
<font face="Arial, Helvetica, sans-serif">
 <small><b>Temp: 37F (2C)</b></small></font>
<font face="Arial, Helvetica, sans-serif">
 <small>Site: <b>KAGC (Pittsburgh/Alle, PA)</b><br>
     Time: <b>2 PM EST 10 DEC 08</b>
```

Induction

Template

```
<img src="images/✱.png" alt="✱"><br>
<font face="Arial, Helvetica, sans-serif">
 <small><b>Temp: ✱ (✱)</b></small></font>
<font face="Arial, Helvetica, sans-serif">
 <small>Site: <b>✱ (✱, ✱)</b><br>
     Time: <b>✱ 10 DEC 08</b>
```

Slot

Stripe

Data Extraction with Templates

- To extract data: Find data in slots by locating the stripes of the template on unseen page:

Unseen Page

```
<img src="images/Sun.png" alt="Sunny"><br>
<font face="Arial, Helvetica, sans-serif">
 <small><b>Temp: 71F (21C)</b></small></font>
<font face="Arial, Helvetica, sans-serif">
 <small>Site: <b>KCQT (Los_Angeles_Dow, CA)</b><br>
      Time: <b>11 AM PST 10 DEC 08</b>
```

+

Induced Template

```
<img src="images/*.png" alt="*"><br>
<font face="Arial, Helvetica, sans-serif">
 <small><b>Temp: * (*)</b></small></font>
<font face="Arial, Helvetica, sans-serif">
 <small>Site: <b>* (*, *)</b><br>
      Time: <b>* 10 DEC 08</b>
```

Extracted Data

Sun	Sunny	71F	21C	KCQT	Los_Angeles_Dow	CA	11 AM PST

Extracting Lists

- Approach:
  - Assume items in a list are formatted using an “item” template
  - Search for “item” templates, using the DOM structure to reduce complexity

Sample Page

```
<td valign="top" width="14%">
  <font face="Arial, Helvetica, sans-serif">
    <small><b>FRIDAY<br>
    <img src="images/Sun-s.png" alt="Sunny"><br>
    HI: 65<br>LO: 52<br></b></small></font></td>
<td valign="top" width="14%">
  <font face="Arial, Helvetica, sans-serif">
  <small><b>SATURDAY<br>
 <img src="images/Rain-s.png" alt="Rainy"><br>
  HI: 60<br>LO: 48<br></b></small></font></td>
```

Induction

Template

```
<td valign="top" width="14%">
  <font face="Arial, Helvetica, sans-serif">
  <small><b>✱<br>
 <img src="images/✱-s.png" alt="✱"><br>
  HI: ✱<br>LO: ✱<br></b></small></font></td>
```

Extraction

FRIDAY	Sun	Sunny	65	52
SATURDAY	Rain	Rainy	60	48

Raw Extracted Data from Unisys

Column	Invocation 1		Invocation 2	…
1	Unisys Weather: Forecast for Washington, DC (20502) [0] 2		Unisys Weather: Forecast for Tallahassee, FL (32399) [0] 2	
2	Washington,		Tallahassee,	
3	DC		FL	
4	20502	**Good Field**	32399	
5	20502)	**Extra Garbage**	32399)	
…				
14	Images/PartlyCloudy.png	**Image URL**	Images/Sun.png	
15	Partly Cloudy	**Good Field**	Sunny	
16	45	**Hard to Recognize**	63	
17	Temp: 45F (7C)	**Too Complex**	Temp: 63F (17C)	
18	45F	**Good Field**	63F	
…				
217	45		64	
218	MOSTLY SUNNY. HIGHS IN THE MID 40S.		PARTLY CLOUDY. HIGHS AROUND 64.	

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Semantic Typing of Extracted Data

- Goal:
  - Assign semantic types to extracted data
- Approach: Leverage background knowledge to semantically type extracted data
  - Learn models of content from samples of known semantic types
  - Use learned models to recognize semantic types of extracted data

Learning Patterns to Recognize Semantic Types

USC Viterbi
School of Engineering

- We developed a 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*

Sample values	Patterns
310 448-8714	
310 448-8775	[310 448 – 4DIGIT]
212 555-1212	[3DIGIT 3DIGIT – 4DIGIT]

- 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]

Using the Patterns for Semantic Labeling

USC Viterbi
School of Engineering

- Use learned patterns to map new data to types in the domain model
  - Score how well patterns associated with a semantic type describe a set of examples
    - *Scoring considers:*
      - *Number of matching patterns*
      - *How specific the matching patterns are*
      - *How many tokens of the example are left unmatched*
  - Output top-scoring types

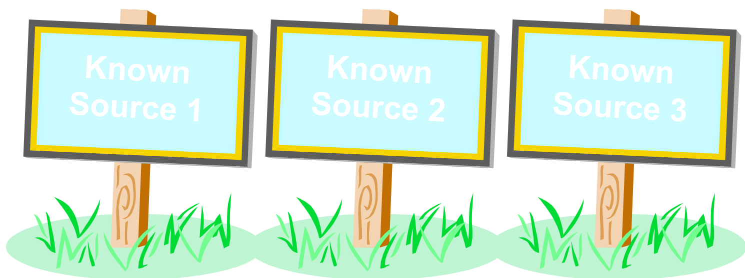
Labeled Columns of Target Source Unisys

Column	4	18	25	15	87
Type	**PR-Zip**	**PR-TempF**	**PR-Humidity**	**PR-Sky**	**PR-Sky**
Score	0.333	0.68	1.0	0.325	0.375
Values	20502	45F	40%	Partly Cloudy	Sunny
	32399	63F	23%	Sunny	Partly Cloudy
	33040	73F	73%	Sunny	Rainy
	90292	66F	59%	Partly Cloudy	Sunny
	36130	62F	24%	Sunny	Partly Cloudy

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Inducing Source Definitions



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

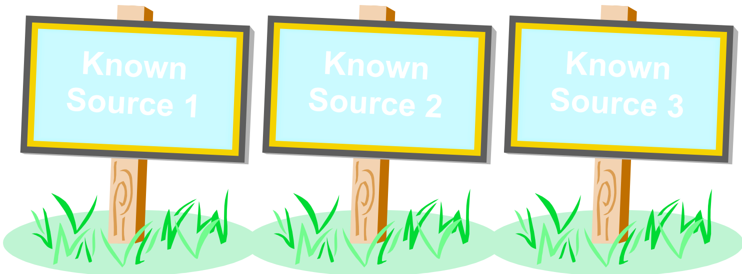
source2($lat1, $long1, $lat2, $long2, dist) :-
  greatCircleDist(lat1, long1, lat2, long2, dist).

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

- Step 1: classify input & output semantic types

source4($startZip, $endZip, separation)

Generating Plausible Definition



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).

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

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).

Top-down Generation of Candidates

USC Viterbi School of Engineering

Start with empty clause & generate specialisations by

- Adding one predicate at a time from set of sources
- Checking that each definition is:
  - Not logically redundant
  - Executable (binding constraints satisfied)


```
source5(_,_,_,_).
```

source5($zip1,$dist1,zip2,dist2)

Expand

```
source5(zip1,_,_,_)           :-  source4(zip1,zip1,_).
source5(zip1,_,zip2,dist2)  :-  source4(zip2,zip1,dist2).
source5(_,dist1,_,dist2)     :-  <(dist2,dist1).
…
```

Invoke and Compare the Definition

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

```
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),
```

match

$zip1	$zip2	dist *(actual)*	dist *(predicted)*
80210	90266	842.37	843.65
60601	15201	410.31	410.83
10005	35555	899.50	899.21

Approximating Equality

Allow flexibility in values from different sources

- Numeric Types like *distance*

  10.6 km ≈ 10.54 km

  Error Bounds (eg. +/- 1%)

- Nominal Types like *company*

  Google Inc. ≈ Google Incorporated

  String Distance Metrics
  (e.g. JaroWinkler Score > 0.9)

- Complex Types like *date*

  Mon, 31. July 2006 ≈ 7/31/06

  Hand-written equality checking procedures.

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Experiments: Source Discovery

- DEIMOS crawls social bookmarking site del.icio.us to discover sources similar to domain seeds:
  - Geospatial: geocoder.us
  - Weather: wunderground.com
- For each seed:
  - retrieve the 20 most popular tags users applied to this source.
  - retrieve other sources that users have annotated with that tags
  - ➢ 15 million source-user-tag triples for the domains.
- ➢ Compute similarity of resources to seed using model
- Evaluation:
  - Manually checked top-ranked 100 resources produced by model
    - *same functionality if same inputs and outputs as seed*
  - Among the 100 highest ranked URLs:
    - *20 relevant geospatial sources*
    - *70 relevant weather sources.*

Experiments: Source Invocation, Extraction and Semantic Typing

USC Viterbi School of Engineering

- **Invocation**: Recognize form input parameters and calling method
- **Extraction**: Learn extractor for resulting output

→ Then, DEIMOS can call websites programmatically as web services.

- **Semantic Typing**: automatically assign semantic types to extracted data

Evaluation:

- Success if extractor produces output table *and* at least one output column not part of the input can be typed
- Given top-ranked 100 URLs, DEIMOS generated
  - 2 semantically-typed geospatial sources

    Ex: ontok($Address, Longitude, Latitude, Street, StateAbbr)
  - 6 semantically-typed weather sources

    Ex. unisys($Zip, Sky, TempF, TempC, _, _, _)

Experiments: Semantic Modeling

Semantic Modeling: learn formal (Datalog) source descriptions based on background knowledge (known sources and types)

- Geospatial Domain
  - Background knowledge (seed source description):

geocoder.us(Address, Street, City, StateAbbr, ZIP, Latitude, Longitude):-
Address(Address, Street, City, StateAbbr, State, ZIP, CountryAbbr, Country, Latitude, Longitude)

  - Learned source descriptions:

ontok($Address, Longitude, Latitude, _, _) :-
geocoder.us(Address, _, _, _, _, Latitude, Longitude)

geocoder.ca($Address, _, StateAbbr, Street, Latitude, _):-
geocoder.us(Address, Street, _, StateAbbr, _, Latitude, _)

Experiments: Semantic Modeling (Weather)

USC Viterbi School of Engineering

Given background source descriptions:

- wunderground(\$Zip, Humidity, $TempF_{hi}$, $TempF_{low}$, $TempF_{hinextday}$, Sky, PressureInches, WindDirection) :- weather(Zip, $TempF_{hi}$, $TempF_{low}$, $TempF_{hinextday}$, Humidity, Sky, PressureInches, WindDirection)
- convertC2F(\$TempC, TempF) :- convertTemp(TempC, TempF)

DEIMOS learned descriptions for 2 sources:

- unisys(\$Zip, Sky, $TempF_{hi}$, TempC, _, _, _) :-
  weather(Zip, $TempF_{hi}$, _, _, _, Sky, _, _),
  convertTemp(TempC, $TempF_{hi}$)

conjunctive source description!

- timetemperature(\$Zip, _, Sky, _, _, $TempF_{low}$, $TempF_{hinextday}$, _):-
  weather(Zip, _, $TempF_{low}$, $TempF_{hinextday}$, _, Sky, _, _)

USC

Experiments: Discussion (I)

+ Sound: only learned correct source descriptions
  - Using both type and value comparison make it very unlikely that an attribute would be modeled incorrectly

~ 60% attributes mapped (3/5, 4/6, 4/7, 4/8)

+ Expressive: learned conjunctive source descriptions
  - Unisys: DEIMOS uses Fahrenheit to Celsius translation function

– Can't learn attributes not present in background sources

– Dynamic sources: Rapidly changing values, update rates
  - cannot compare temperatures if seed, target invocations too distant
  - sites reported very different humidity values

Experiments: Discussion (II)

- Extraction errors => missed types
  - Ex: "<font size='-1'>FL"
    - *too many spurious tokens to be considered similar to "FL"*
  - Ex: 118.440470 vs. -118.440470:
    - *extractor missed – sign, not a longitude*
  - Mixed-value columns:
    - *variable number of data items returned for different inputs can sometimes fool extractor*
    - *Ex: weather advisory attribute appears for one input and not for others → shift in columns → mixed value columns*

- Semantic Typing errors
  - Ex: labeled time zone codes as WindDirection due to 3caps pattern learned (WSW vs PST)

➔ Overall, promising results

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Related Work

USC Viterbi
School of Engineering

ILA & Category Translation (Perkowitz & Etzioni 1995)
Learn functions describing operations on internet

- Our system learns *more complicated* definitions
  - Multiple attributes, Multiple output tuples, etc.

iMAP (Dhamanka et. al. 2004)
Discovers complex (many-to-1) mappings between DB schemas

- Our system learns *many-to-many* mappings
- Our approach is more general
- We deal with problem of invoking sources

Related Work

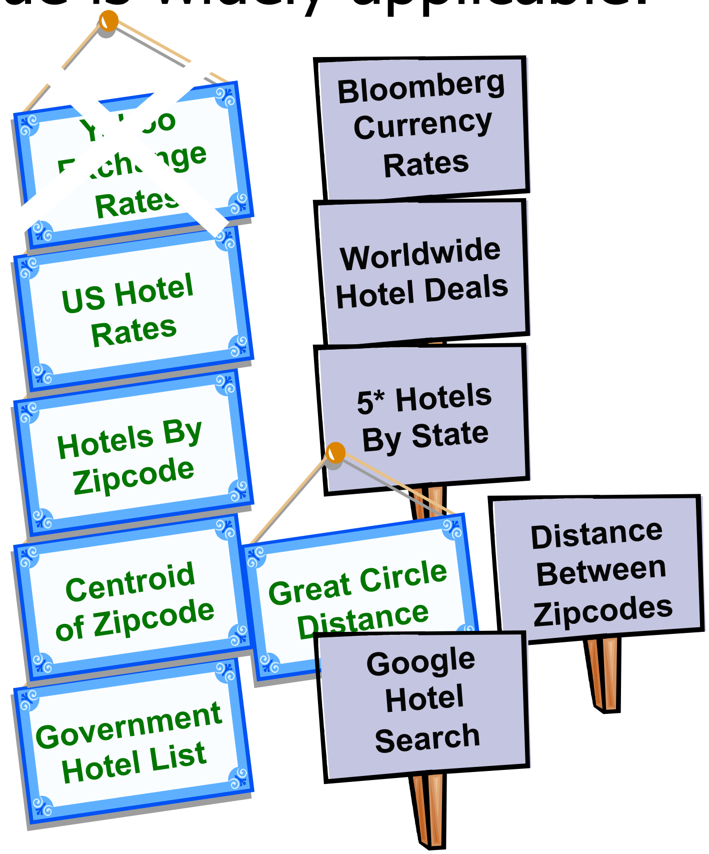
- Metadata-based classification of data types used by Web services and HTML forms (Hess & Kushmerick, 2003)
  - Naïve Bayes classifier
  - No invocation of services
- Woogle: Metadata-based clustering of data and operations used by Web services (Dong et al, 2004)
  - Groups similar types together: Zipcode, City, State
  - Cannot invoke services with this information

Outline

- Discovering related sources
- Automatically invoking the sources
- Constructing syntactic models of the sources
- Determining the semantic types of the data
- Building semantic models of the sources
- Experimental Results
- Related Work
- Conclusions

Coverage

- Assumption: overlap between new & known sources
- Nonetheless, the technique is widely applicable:
  - Redundancy
  - Scope or Completeness
  - Binding Constraints
  - Composed Functionality
  - Access Time



Discussion

- Integrated approach to discovering and modeling online sources and services:
  - *Discover new sources*
  - *How to invoke a source*
  - *Discovering the template for the source*
  - *Finding the semantic types of the output*
  - *Learning a definition of what the service does*

- Provides an approach to generate source descriptions for the Semantic Web
  - Little motivation for providers to annotate services
  - Instead we can generate metadata automatically

Future Work

USC Viterbi
School of Engineering

- Scalability!
  - Difficult to invoke sources with many inputs
    - *Hotel reservation sites*
  - Hard to learn sources that have many attributes
    - *Some weather sources could have 40 attributes*
- Learning beyond the domain model
  - Learn new semantic types
  - Learn new source attributes
  - Learn new source relations
  - Learn the domain and range of the sources

USC