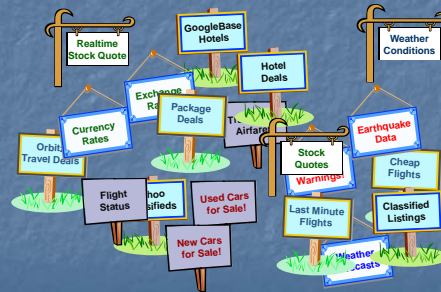


Learning Definitions of Online Sources for Information Integration

Craig Knoblock
University of Southern California

This is joint work with Mark Carman,
Kristina Lerman, and Anon Plangprasopschok

Abundance of Information Sources

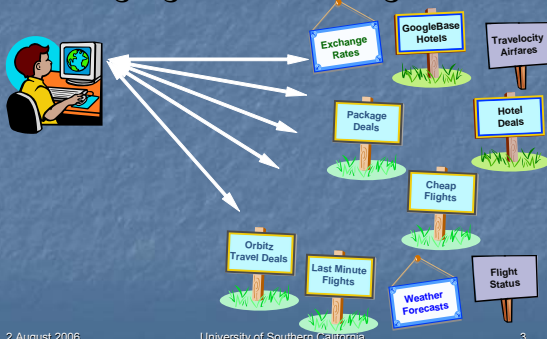


2 August 2006

University of Southern California

2

Bringing the Data Together

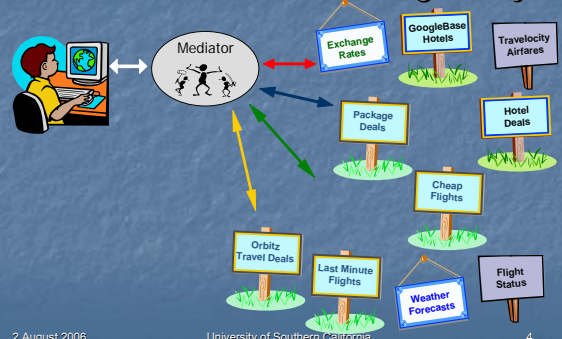


2 August 2006

University of Southern California

3

Mediators resolve Heterogeneity



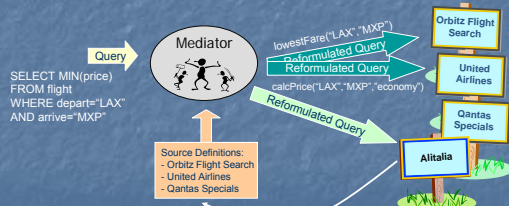
2 August 2006

University of Southern California

4

Mediators Require Source Definitions

- New service => no source definition!
- Can we discover a definition automatically?

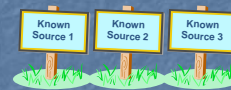


2 August 2006

University of Southern California

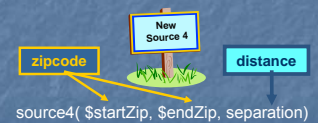
5

Inducing Source Definitions by Example



```
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




2 August 2006

University of Southern California

6

Motivation Approach Classify Search Scoring Related Work Conclusions

Inducing Source Definitions - Step 2



- 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).

```

2 August 2006 University of Southern California

Motivation Approach Classify Search Scoring Related Work Conclusions

Inducing Source Definitions – Step 3

- 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),
  convertKm2Mi(dist1, 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

2 August 2006 University of Southern California 8

Motivation Approach Classify Search Scoring Related Work Conclusions

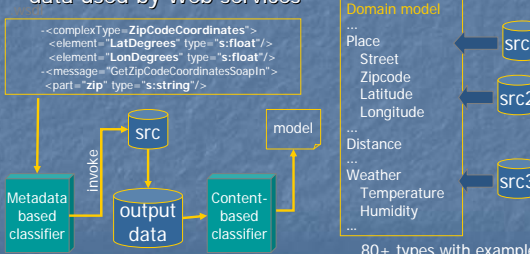
Our Approach to Semantic Labeling

Leverage existing knowledge to learn semantics of data used by Web services

```

<-complexType=ZipCodeCoordinates">
  <element=LatDegrees type="s:float"/>
  <element=LonDegrees type="s:float"/>
<-message=GetZipCodeCoordinatesSoapIn">
  <part=zip type="s:string"/>

```



80+ types with examples

2 August 2006 University of Southern California 9

Motivation Approach Classify Search Scoring Related Work Conclusions

Metadata-based Classification

- Observation 1: Similar data types tend to be named with similar words, and/or belong to operations that have similar name
 - Treat as (ungrammatical) text classification problem
 - Approach taken by previous works
- Observation 2: The classifier must be a soft classifier
 - Instance can belong to more than one class
 - Rank classification results

2 August 2006 University of Southern California 10

Motivation Approach Classify Search Scoring Related Work Conclusions

Independence Assumption

- Naïve Bayes classifier
 - Used to classify parameters used by Web services (Hess & Kushmerick, 2004)
 - Each input/output parameter represented by a term vector t
 - Based on independence assumption
 - Terms are independent from each others given the class label D (semantic type)
 - $P(D|t) \propto \prod_i P(t_i|D)$
 - Independence assumption unrealistic for Web services
 - e.g., "TempFahrenheit": "Temp" and "Fahrenheit" often co-occur in the Temperature semantic type
 - Logistic regression avoids the independence assumption
 - Estimates probabilities from the data
 - $P(D|t) = \text{logreg}(wt)$

2 August 2006 University of Southern California 11

Motivation Approach Classify Search Scoring Related Work Conclusions

Metadata-based Classification Evaluation

- Data collection
 - Data extracted from 313 WSDL files from Web service portals (bindingpoint and webserviceX)
- Data processing
 - Names were extracted from operation, message, datatype and facet (predefined option)
 - Names tokenized into individual terms
- 10,000+ data types extracted
 - Each one assigned to one of 80 classes in geospatial and weather domains (e.g. latitude, city, humidity).
 - Other classes treated as "Unknown" class

2 August 2006 University of Southern California 12

Motivation Approach Classify Search Scoring Related Work Conclusions

Evaluation Results

- Both Naïve Bayes and Logistic regression were tested using 10-fold cross validation

Classifier	Top 1	Top 2	Top 3	Top 4
Naïve Bayes	0.65	0.84	0.88	0.90
Logistic Regression	0.93	0.98	0.99	0.99

2 August 2006 University of Southern California 13

Motivation Approach Classify Search Scoring Related Work Conclusions

Content-based Classification

- Idea: Learn a model of the content of data and use it to recognize new examples

Developed a domain-independent language to represent the structure of data

- Token-level
 - Specific tokens
 - General token types
 - based on syntactic categories of token's characters
- Hierarchy of types
 - allows for multi-level generalization

2 August 2006 University of Southern California 14

Motivation Approach Classify Search Scoring Related Work Conclusions

Patterns for Describing Data

- Pattern is a sequence of tokens and general types
 - Phone numbers
 - Examples: 310 448-8714, 310 448-8775, 212 555-1212
 - Patterns: [(310) 448 - 4DIGIT], [(3DIGIT) 3DIGIT - 4DIGIT]
- Algorithm to learn patterns from examples
- Patterns for all semantic types in the domain model

2 August 2006 University of Southern California 15

Motivation Approach Classify Search Scoring Related Work Conclusions

Patterns for Semantic Labeling

- 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
 - Heuristics include:
 - Number of matching patterns
 - How specific the matching patterns are
 - How many tokens of the example are left unmatched
 - Output four top-scoring types

2 August 2006 University of Southern California 16

Motivation Approach Classify Search Scoring Related Work Conclusions

Semantic Labeling Evaluation

Information domains and semantic types

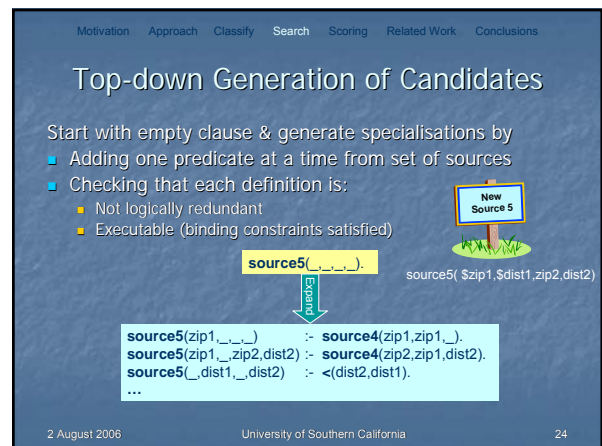
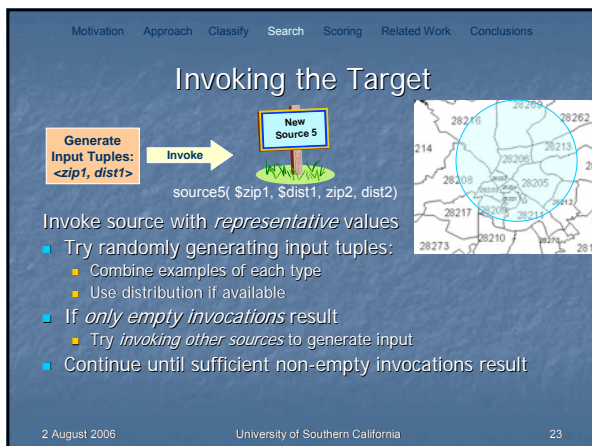
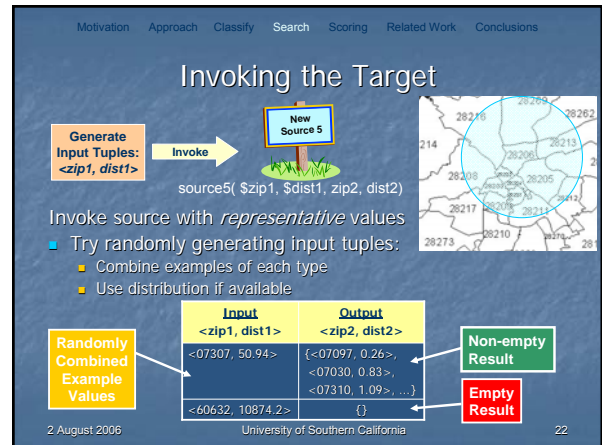
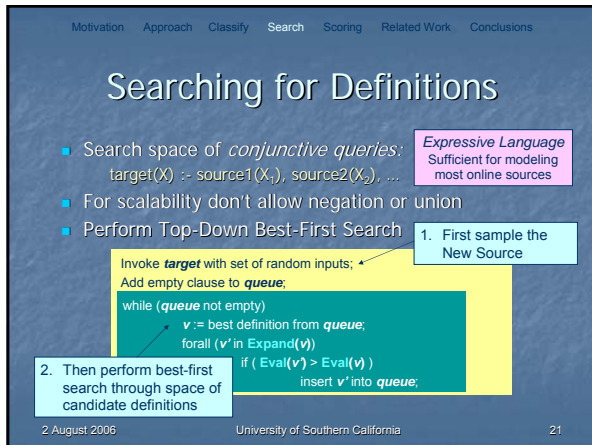
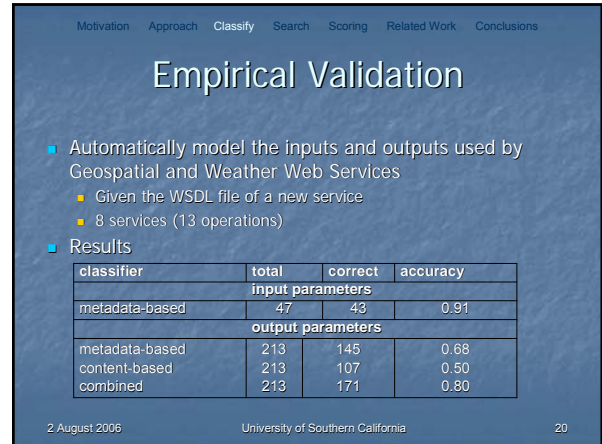
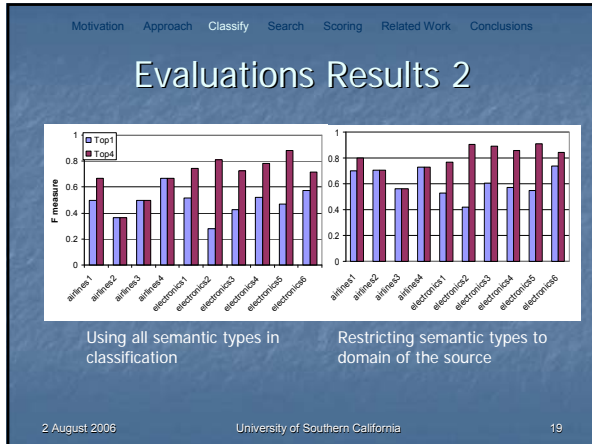
- Weather Services
 - Temperature, SkyConditions, WindSpeed, WindDir, Visibility
- Directory Services
 - Name, Phone, Address
- Electronics equipment purchasing
 - ModelName, Manufacturer, DisplaySize, ImageBrightness, ...
- UsedCars
 - Model, Make, Year, BodyStyle, Engine, ...
- Geospatial Services
 - Address, City, State, Zipcode, Latitude, Longitude
- Airline Flights
 - Airline, flight number, flight status, gate, date, time

2 August 2006 University of Southern California 17

Motivation Approach Classify Search Scoring Related Work Conclusions

Evaluations Results

2 August 2006 University of Southern California 18



Motivation Approach Classify Search Scoring Related Work Conclusions

Best-first Enumeration of Candidates

- Evaluate each clause produced
- Then expand best one found so far
- Expand high-arity predicates incrementally

2 August 2006 University of Southern California 25

Motivation Approach Classify Search Scoring Related Work Conclusions

Limiting the Search

- Extremely Large Search space
- Constrained by use of Semantic Types
- Limit search by:
 - Maximum Clause length
 - Maximum Predicate Repetition
 - Maximum Number of Existential Variables
 - Definition must be Executable
 - Maximum Variable Repetition within Literal

Standard ILP techniques

Non-standard technique

2 August 2006 University of Southern California 26

Motivation Approach Classify Search Scoring Related Work Conclusions

Evaluating Candidates

- Compare output of clause with that of target.
- Average the results across different input tuples.

2 August 2006 University of Southern California 27

Motivation Approach Classify Search Scoring Related Work Conclusions

Evaluating Candidates II

Candidates may return multiple tuples per input

- Need measure that compares sets of tuples!

Input	Target Output	Clause Output
<\$zip1, \$dist1>	<zip2, dist2>	<zip2, dist2>
<60632, 874.2>	{}	{<60629, 2.15>, <60632, 2.27>, <60623, 2.64>, ...}
<07307, 50.94>	{<07097, 0.26>, <07030, 0.83>, <07310, 1.09>, ...}	{}
<28041, 240.46>	{<28072, 1.74>, <28146, 3.41>, <28138, 3.97>, ...}	{<28072, 1.74>, <28146, 3.41>, ...}

No Overlap

No Overlap

Overlap!

2 August 2006 University of Southern California 28

Motivation Approach Classify Search Scoring Related Work Conclusions

Evaluating Candidates III

PROBLEM: All sources assumed incomplete

- Even *optimal definition* may only produce overlap
- Want definition that *best predicts* the target's output
- Use Jaccard similarity to score candidates

At least half of input tuples are non-empty invocations of target

```

forall (tuple in InputTuples)
  T_target = invoke(target, tuple)
  T_clause = execute(clause, tuple)
  if not (|T_target|=0 and |T_clause|=0)
    fitness = |T_target ∩ T_clause| / |T_target ∪ T_clause|
return average(fitness)

```

Average results only when output is returned

Similarity metric is Jaccard similarity between the sets

2 August 2006 University of Southern California 29

Motivation Approach Classify Search Scoring Related Work Conclusions

Missing Output Attributes

- Some candidates produce less output attributes:
 - Makes comparing them difficult

- source5(zip1,_) :- source4(zip1,zip1,_)
- source5(zip1_.,zip2,dist2) :- source4(zip2,zip1,dist2)

Penalize candidate by number of "negative examples"

source5(\$zipcode, \$distance, zipcode, distance)

- First candidate doesn't produce either outputs, thus:
 - Penalty = |{zipcode}| x |{distance}|
 - For numeric types use accuracy to approximate cardinality

2 August 2006 University of Southern California 30

Motivation Approach Classify Search Scoring Related Work Conclusions

Different Input Attributes

- Some clauses take different inputs from target:


```
source5($zip1,$dist1,zip2,...) :- source4($zip1,$zip2,dist1).
```

Target Input

Clause Input
- zip2** is an input parameter for clause but not target
- Should invoke operation with *every possible zip code!*
 - > 40,000 zip codes in US
- Problem: algorithm should return & not get banned!
- Solution: sample to estimate score for clause;
 - record the scaling factor = $|{\text{zipcode}}| / \# \text{invocations}$
 - bias search: choose at least half of tuples to be positive

2 August 2006 University of Southern California 31

Motivation Approach Classify Search Scoring Related Work Conclusions

Approximating Equality

Allow flexibility in values from different sources

- Numeric Types like *distance*
 - 10.6 km \approx 10.54 km
 - Error Bounds (eg. +/- 1%)
- Nominal Types like *company*
 - Google Inc. \approx Google Incorporated
 - String Distance Metrics (e.g. Jaro/Winkler Score > 0.9)
- Complex Types like *date*
 - Mon, 31. July 2006 \approx 7/31/06
 - Hand-written equality checking procedures.

2 August 2006 University of Southern California 32

Motivation Approach Classify Search Scoring Related Work Conclusions

Experiments – Setup

Problems:

- 25 target predicates
- same domain model (70 Semantic Types at)
- 35 known sources

System Settings:

- Each target source invoked at least 20 times
- Time limit of 20 minutes imposed

Inductive search bias:

- Maximum clause length 7
- Predicate repetition limit 2
- Maximum variable level 5
- Candidate must be executable
- Only 1 variable occurrence per literal

Equality Approximations:

- 1% for *distance, speed, temperature & price*
- 0.002 degrees for *latitude & longitude*
- Jaro/Winkler > 0.85 for *company, hotel & airport*
- hand-written procedure for *date*.

2 August 2006 University of Southern California 33

Motivation Approach Classify Search Scoring Related Work Conclusions

Actual Learned Examples

- GetDistanceBetweenZipCodes(\$zip0, \$zip1, dis2):-
GetCentroid(zip0, lat1, lon2), GetCentroid(zip1, lat4, lon5),
GetDistance(lat1, lon2, lat4, lon5, dis10), ConvertKm2Mi(dis10, dis2).
- USGSElevation(\$lat0, \$lon1, dis2):-
ConvertFt2M(dis2, dis1), Altitude(lat0, lon1, dis1). Distinguished forecast from current conditions
- YahooWeather(\$zip0, cit1, sta2, , lat4, lon5, day6, dat7, tem8, tem9, sky10) :-
WeatherForecast(cit1, sta2, lat4, lon5, day6, dat7, tem9, tem8, , sky10, , ,),
GetCityState(zip0, cit1, sta2). current price = yesterday's close + change
- GetQuote(\$tic0, pri1, dat2, tim3, pri4, pri5, pri6, pri7, cou8, , pri10, , pri13, com15) :-
YahooFinance(tic0, pri1, dat2, tim3, pri4, pri5, pri6, pri7, cou8),
GetCompanyName(tic0, com15, ,), Add(pri5, pri13, pri10), Add(pri4, pri10, pri1).
- YahooAutos(\$zip0, \$mak1, dat2, yea3, mod4, , pri7, ,) :-
GoogleBaseCars(zip0, mak1, , mod4, pri7, , , yea3),
ConvertTime(dat2, , dat10, ,), GetCurrentTime(, , dat10, ,).

2 August 2006 University of Southern California 34

Motivation Approach Classify Search Scoring Related Work Conclusions

Experimental Results

- Results for different domains:

Problem Domain	# of Problems	Avg. # of Candidates	Avg. Time (sec)	Attributes Learnt
geospatial	9	136	303	84%
financial	2	1606	335	59%
weather	7	368	693	69%
hotels	4	43	374	60%
cars	2	68	940	50%

2 August 2006 University of Southern California 35

Motivation Approach Classify Search Scoring Related Work Conclusions

Related Work

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 (single search algorithm)
- We deal with problem of invoking sources

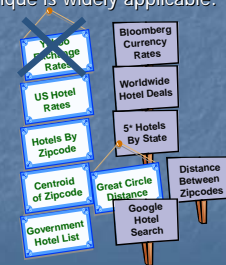
2 August 2006 University of Southern California 36

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

Discussion

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



Conclusions

- Integrated approach to learning:
 - *How to invoke a web service*
 - *The semantic types of the output*
 - *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 generate metadata automatically
- Also provides an approach to automatically discover new sources of data