A Reference-Set Approach to Information Extraction from Unstructured, Ungrammatical Data Sources

Craig Knoblock
University of Southern California

This is joint work with Matthew Michelson
Fetch Technologies
Motivation: Data Integration

**Query:** Average price for a 3-star crash-rated Honda, and reviews.

Structured Sources
- NHTSA Ratings

Semi-Structured Sources
- Car Review

Integrate?
Introduction • Unsupervised IE • Building Reference Sets • Supervised IE • Conclusion

Unstructured, Ungrammatical Data:

[ Fri, 14 Mar 11:45:39 ]

[ ALERT - offers to ship cars/trucks are fraudulent ] [ partial list of prohibited items ]
[ avoid recalled items ] [ success story? ] [ AVOIDING SCAMS & FRAUD ]
[ PERSONAL SAFETY TIPS ]

Fri Mar 14

91 Civic SI RHD SHELL - $2900 - (West Comma) pic

2001 Automatic Mazda Millenia Clear Title - $3800 - pic

1984 Ford Tow Truck - $10000 - (Bell)

2004 Audi A4 1.8T - $6800 - pic

1998 International 4700 Tow Truck - $12000 - (Bell)

1994 >>>>>> LEXUS ES 300 >> LEATHER INTERIOR <<< - $3000 - (RESEDA) pic

1987 Chevrolet Tahoe 4x4 just smogged - $1400 - (Palmdale) pic
Structured Queries? … Information Extraction/Annotation!

Introduction • Unsupervised IE • Building Reference Sets • Supervised IE • Conclusion

MAKE: HONDA (implied!)
MODEL: CIVIC
TRIM: 2 Door SI
YEAR: 1991

Price: $2900

[ ALERT - offers to ship cars/trucks are avoid recalled items ] [ success... ]

Model: Civic
Year: 91
Trim: SI
Price: $2900

Fri Mar 14

91 Civic SI RHD SHELL - $2900 - (West Covina)

2001 Automatic Mazda Millenia Clear Title - $3800 - pic

1984 Ford Tow Truck - $10000 - (Bell)
Difficulties

● Unstructured
  ● No assumptions on structure
  ● “Rule/Pattern” based techniques unsuited

● Ungrammatical
  ● Does not conform to English grammar
  ● Natural-Language Processing techniques unsuited
Reference-Set Based Extraction/Annotation

91 Civic SI RHD SHELL - $2900 -

Reference Set(s)

Record Linkage

Information Extraction

Annotation

HONDACLICICT2 Door SI1991

Extracted Attributes

Civic SI 91 $2900

Query

Integrate
Reference Sets

- Collections of entities and their attributes
  - List cars $\rightarrow$ <make, model, trim, ...>

Extract make, model, trim, year for all cars from 1990-2005...
Talk Topics

- Automatic matching and extraction using reference sets
  - Michelson & Knoblock, IJDAR, 2007
  - Code @ mmichelson.com

- Automatically building reference sets from the posts
  - Michelson & Knoblock, IJCAI, 2009
  - Michelson & Knoblock, JAIR, 2010

- Supervised machine learning w/ reference sets
  - Michelson & Knoblock, IJCAI, 2005
  - Michelson & Knoblock, JAIR, 2008
  - Code @ mmichelson.com
Automatic method: Three steps

1) Select reference set(s)

2) Find best matches (automatic)

3) Extraction using matches (automatic)

ARX: Automatic Reference-set based eXtraction
Selecting the Reference Set(s)

Vector space model: set of posts are 1 doc, reference sets are 1 doc

Select reference set most similar to the set of posts…

- FORD Thunderbird - $4700
- 2001 White Toyota Corolla CE Excellent Condition - $8200

<table>
<thead>
<tr>
<th>Reference Set</th>
<th>SIM</th>
<th>Similarity Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cars</td>
<td>0.7</td>
<td>PD(C,H) = 0.75 &gt; T</td>
</tr>
<tr>
<td>Hotels</td>
<td>0.4</td>
<td>PD(H,R) = 0.33 &lt; T</td>
</tr>
<tr>
<td>Restaurants</td>
<td>0.3</td>
<td>Avg. 0.47</td>
</tr>
</tbody>
</table>

Cars 0.7, PD(C,H) = 0.75 > T
Hotels 0.4, PD(H, R) = 0.33 < T
Restaurants 0.3, Avg. 0.47
Automatic matching between the posts and reference set

new 2007 altima

02 M3 Convertible .. Absolute beauty!!!
Awesome car for sale! Cheap too!

{NISSAN, ALTIMA, 4 Dr 3.5 SE Sedan, 2007}
{NISSAN, ALTIMA, 4 Dr 2.5 S Sedan, 2007}
{BMW, M3, 2 Dr STD Convertible, 2002}
{LINCOLN, TOWN CAR, 4 Dr, 2001}
{RENAULT, LE CAR, 2 Dr, 1987}

Vector-space matching

Prune false positives!

Introduction • Unsupervised IE • Building Reference Sets • Supervised IE • Conclusion
Automatic Extraction

91 Civic SI RHD SHELL - $2900 -

Honda

Civic

2 Dr SI

make model trim year

Civic SI 91

Clean Whole Attribute
Results: Information Extraction

- State-of-the-art comparison
  1. Conditional Random Field (structure)
     1. CRF-Orth
        - Orthographic features: cap, start-num, etc.
     2. CRF-Win
        - CRF-Orth + 2-word sliding window
          - more structure!
  2. Amilcare
     - NLP
     - “Gazetteers” (list of hotels, etc.)

- ARX = automatic, others = supervised
- Field-level extractions
  - All tokens required, no extras (strict!)
Results: Information Extraction

### Craigs Cars Posts (Craigslist)

<table>
<thead>
<tr>
<th></th>
<th>ARX</th>
<th>CRF-Orth</th>
<th>CRF-Win</th>
<th>Amilcare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make</td>
<td><strong>97.95</strong></td>
<td>83.66</td>
<td>78.67</td>
<td>94.57</td>
</tr>
<tr>
<td>Model</td>
<td><strong>88.61</strong></td>
<td>74.25</td>
<td>68.72</td>
<td>81.24</td>
</tr>
<tr>
<td>Trim</td>
<td><strong>49.70</strong></td>
<td>47.88</td>
<td>38.75</td>
<td>35.94</td>
</tr>
<tr>
<td>Year</td>
<td>86.47</td>
<td>88.04</td>
<td>84.52</td>
<td><strong>88.97</strong></td>
</tr>
</tbody>
</table>

~27,000 cars: Edmunds/ Super Lamb Auto

- **ARX**
  - Automatic & better than supervised on 5/7 attributes
  - Cases where ARX underperforms
    - w/in 5%
    - Strong numeric component
  - Recall issue

- **CRF-Win**
  - Worst on 6/7
  - Can’t rely on structure!

### BFT Posts (biddingfortravel.com)

<table>
<thead>
<tr>
<th></th>
<th>ARX</th>
<th>CRF-Orth</th>
<th>CRF-Win</th>
<th>Amilcare</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Rating</td>
<td>91.03</td>
<td>94.77</td>
<td>94.21</td>
<td><strong>96.46</strong></td>
</tr>
<tr>
<td>Hotel Name</td>
<td><strong>73.46</strong></td>
<td>67.47</td>
<td>41.33</td>
<td>62.91</td>
</tr>
<tr>
<td>Local Area</td>
<td><strong>71.98</strong></td>
<td>70.19</td>
<td>33.07</td>
<td>68.01</td>
</tr>
</tbody>
</table>

~130 hotels: BiddingForTravel.com

Automatic, state-of-the-art extraction on posts
Construction of Reference Sets

- What if there isn’t already a reference set?

<table>
<thead>
<tr>
<th>HP Pavilion DV2000 laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway ML6230, Intel Cel …</td>
</tr>
</tbody>
</table>

- What about coverage?

<table>
<thead>
<tr>
<th>Ford</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dodge</td>
<td>Caravan</td>
</tr>
</tbody>
</table>

- ACURA TL 3.2 VTEC - 1999

Diagram:
1. Mine Reference Set
2. Reference Set (s)
3. Find Best Match from Reference Set
4. Information Extraction
Seed-Based Reference Set Construction

- Use posts themselves
  - Overcome difficulty in finding full reference sets
    - Enumeration
    - Dynamic data
  - Overcome coverage issues
    - Using posts guarantees coverage
Seed-Based Reference Set Construction

- Seeds
  - Smallest (most obvious) domain knowledge
    - Computer Makers: Apple, Dell, Lenovo
    - Easy to enumerate
  - Constrains tuples constructed (roots)
    - Cleaner reference set
  - Relatively static
    - Less change to worry about
- Posts themselves to fill in details
  - Computer Models, Model Nums…
Entity Trees

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Honda</td>
<td>Accord</td>
</tr>
<tr>
<td>Honda</td>
<td>Civic</td>
</tr>
<tr>
<td>Ford</td>
<td>Focus</td>
</tr>
</tbody>
</table>

Reference Set

Forest of “Entity Trees”

Reference Set Construction = Constructing this forest
Entity Trees from Posts

**Introduction**

**Unsupervised IE**

**Building Reference Sets**

**Supervised IE**

**Conclusion**

**Step 1**

Construct Bi-Grams

**Step 2**

Create entity trees

Form reference set

**Poses**

---

91 Civic SI RHD ...

{91 Civic}

{Civic SI}

{SI RHD}

...

Seeds = roots

Fill in rest using posts
Constructing Entity Trees

- Sanderson & Croft heuristic
  - $x$ SUBSUMES $y$ IF $P(x|y) \geq 0.75$ & $P(y|x) \leq P(x|y)$

- Merge heuristic
  - MERGE($x, y$) IF $x$ SUBSUMES $y$ & $P(y|x) \geq 0.75$

Honda civic is cool
Honda civic is nice
Honda accord rules
Honda accord 4 u!

- Construct hierarchies, then flatten

<table>
<thead>
<tr>
<th>Honda</th>
<th>CIVIC</th>
<th>ACCORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>HONDA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
General Tokens

- \{a, y\}, \{b, y\}, \{c, y\} → y is “general token”
  - Occurs across entity trees…

- Instead use \( P( \{a U b U c\} \mid y) \)
- e.g. car trims: Pathfinder LE, Corolla LE, …
- Build entity trees
  - Do 1 Scan
    - Build initial trees
  - Iterate
    - Find “general tokens”
No seeds?

- “Iterative Locking Algorithm”
  - Instead of seeds, “lock” levels of the tree
  - Entropy of finding current leaves
    - Uncertainty labeling attributes
    - Compare % diff across # posts
  - Locks out noise
- **How many posts are enough?**
  - When you lock all levels

Key: redundancy:
At some point you’ve gotten all you can from the posts
Experiments & Results

- **Goal**
  - How to compare reference sets?
    - Ontology comparison is rather open…
    - Might not take into account utility of reference set…
  - Extraction = proxy task to compare reference sets
    - Poor coverage $\rightarrow$ poor recall
    - Noise $\rightarrow$ bad extractions $\rightarrow$ worse results

- **Compare extraction** (use ARX)
  - Constructed using seeds ("Seed-based")
  - Constructed without seeds ("Auto")
  - Manually constructed reference sets ("Manual")
# Experiments & Results

## Experimental Domains:

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Attributes</th>
<th>Num. Posts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cars</td>
<td>Craigslist</td>
<td>make, model, trim</td>
<td>2,568</td>
</tr>
<tr>
<td>Laptops</td>
<td>Craigslist</td>
<td>maker, model, model num.</td>
<td>2,921</td>
</tr>
<tr>
<td>Skis</td>
<td>eBay</td>
<td>brand, model, model spec.</td>
<td>4,981</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Num. Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cars</td>
<td>Edmunds</td>
<td>~27,000</td>
</tr>
<tr>
<td>Laptops</td>
<td>Overstock</td>
<td>279</td>
</tr>
<tr>
<td>Skis</td>
<td>Skis.com</td>
<td>213</td>
</tr>
</tbody>
</table>

“Manual” reference sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Source</th>
<th>Num. Seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cars</td>
<td>Edmunds</td>
<td>102 makes</td>
</tr>
<tr>
<td>Laptops</td>
<td>Wikipedia</td>
<td>40 makers</td>
</tr>
<tr>
<td>Skis</td>
<td>Skis.com</td>
<td>18 brands</td>
</tr>
</tbody>
</table>
Experiments & Results (seed based)

<table>
<thead>
<tr>
<th></th>
<th>vs. Auto</th>
<th>vs. Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outperforms</td>
<td>9/9</td>
<td>5/9</td>
</tr>
<tr>
<td>Within 5%</td>
<td>9/9</td>
<td>7/9</td>
</tr>
</tbody>
</table>

- Seed-based vs. Manual
  - Outperforms on majority of attributes / Competitive on most
    - # seeds << # records in manual reference set
  - Does best on hard to cover attributes
    - Ski model & model spec., Laptop model & model num.
      - Only 53.15% of values for these exist in manual sets!
      - Overstock = New computers, Craigslist = old computers
  - Poor performance vs. manual
    - Car trim: missing tokens (didn’t mine)
      - E.g. Manual = 4 Dr DX 4WD, Seed = DX
      - Miss “4 Dr” part of extraction → wrong in field-level results
Experiments & Results (locking based)

- Converges in all domains
  - E.g., locks before seen all posts
- Outperforms “Auto” on all Laptop attributes
  - Stat sig. 95%
- Cars/Skis
  - Only 1 significant difference vs. “Auto”
- Should try to lock
  - Can’t hurt you (only 1 significant drop), and in best case can help a lot (laptop)
Supervised Machine Learning for Extraction from Posts

- Require **highest-accuracy** extraction
  - Ambiguity: 626, Mazda or car price?
Supervised Machine Learning for Extraction

Record Level Similarity + Field Level Similarities

1. Record Linkage

\[ V_{RL} = \langle RL_{scores}(post, attribute_1, attribute_2 \ldots attribute_n), \]
\[ \quad RL_{scores}(post, attribute_1), \]
\[ \quad \ldots, \]
\[ \quad RL_{scores}(post, attribute_n) \rangle \]

Binary Rescoring

SVM

2. Supervised Extraction

Compare to match’s attributes

Multiclass-SVM / CRF
Results: Information Extraction

<table>
<thead>
<tr>
<th>Domain</th>
<th>Num. of Attributes with Max F-Mes.</th>
<th>Total Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Phoebus</td>
<td>PhoebusCRF</td>
</tr>
<tr>
<td>BFT</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>eBay Comics</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Craig’s Cars</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>All</td>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

- Phoebus/PhoebusCRF
  - Best 12/16 attributes (> ARX > other methods)
  - Different extraction methods → reference set makes difference

- CRF-Win max: Comics price attribute
  - Not statistically significant…
  - CRFs outperformed
    - No structure to rely on!

- Amilcare/ARX use reference sets
  - Every max F-mes. used reference set
Related Work

- **Semantic Annotation**
  - Require grammar/structure (Cimiano, Handschuh & Staab, 2004; Dingli, Ciravegna, & Wilks, 2003; Handschuh, Staab & Ciravegna, 2002; Vargas-Vera, et. al., 2002)

- **Record Linkage**
  - Decomposed attributes (Fellegi & Sunter, 1969; Bilenko & Mooney, 2003)
  - WHIRL (Cohen, 2000): simple matching

- **Data Cleaning**
  - Tuple-to-Tuple (Lee, et. al., 1999; Chaudhuri, et. al., 2003)

- **Blocking**
  - Other work focuses on methods, not choosing attributes (Baxter, Christen, & Churches, 2003; McCallum, Nigam, & Ungar, 2000; Winkler, 2005)
  - Bilenko, Kamath, & Mooney, 2006: graphical set covering
Related Work (2)

- **Unstructured information extraction**
  - DataMold (Borkar, Deshmukh, & Sarawagi, 2001), CRAM (Agichtein & Ganti, 2004): no junk tokens
  - Semi-CRF methods (Cohen & Sarawagi, 2004): dictionary component, but look-up

- **Ontology based IE**
  - requires ontology management (Embley, et. al., 1999; Ding, Embley & Liddle, 2006; Muller, et. al., 2004)

- **Ontology creation**
  - Use web pages to build single hierarchies (Sanderson & Croft, 1999; Schmitz, 2006; Comiano, Hotho & Staab, 2004; Dupret & Piwowarski, 2006; Makrehchi & Kamel, 2007)

- See papers for more comprehensive RW…
Conclusion: Topics Covered

- Automatic, state-of-the-art extraction on posts given reference set(s)
- Automatically build reference set for cases where difficult to do so manually
- Supervised extraction on posts with highest accuracy
Questions?

Code & Data:
mmichelson.com

Thanks

THANK YOU