Semantic Information Extraction: Overview and Basic Techniques

Shuming Shi
Microsoft Research Asia
Mar. 2012
Outline

• Overview
• Semantic class mining
• Semantic hierarchy construction
• Mining attribute names and values
• General relation extraction
• Demo
• Summary
Information Extraction (IE) Definitions

• In Wikipedia
  - A process to “extract structured information from unstructured and/or semi-structured machine-readable documents”
  - Sample
    • "Yesterday, New-York based Foo Inc. announced their acquisition of Bar Corp." ➔ MergeBetween(company1,company2,date)

• In “Speech and Language Processing” (D. Jurafsky & J.H. Martin)
  - “turns the unstructured information embedded in texts into structured data”
  - “an effective way to populate the contents of a relational database”
  - “extract limited kinds of semantic content from text”
IE and Semantic IE

Unstructured data
(plain text articles, sentences, query strings…)

Semi-structured data
(HTML documents, query logs, web search results, dictionaries, user interaction data…)

Semantic IE

IE

Structured data

Semantic knowledge
Semantic Information Extraction

• Motivations
  – Build “smarter” computer systems with the semantic knowledge-base
  – Better fulfill the information needs of programs & end users
    • Better web search
    • Better QA
    • Better machine translation
    • …
Semantic Information Extraction

- Major tasks
  - Named entity extraction
    - Named entity recognition (NER)
    - Co-reference resolution
  - Attribute extraction
  - Relation mining
    - Related terms and entities
    - Categorization
    - Relation detection and classification
  - Event mining
    - Event detection and classification
  - ...
Britney Jean Spears (born December 2, 1981) is an American recording artist and entertainer. Born in McComb, Mississippi, and raised in Kentwood, Louisiana, Spears began performing as a child, landing acting roles in stage productions and television shows. She signed with Jive Records in 1997 and...”

(from Wikipedia)
IE Task: Attribute Extraction

**Beijing:**
- **Country:** China
- **Time zone:** China Standard Time
- **Area:** 16,801.25 km²
- **Population:** 19,612,000
- **Elevation:** 43.5 m

**Kinect:**
- **Product family:** Xbox
- **Company:** Microsoft
- **Resolution:** 680*480
- **Release date:** Nov. 4, 2010
- **Games:** Kinect Sports, Kinect Adventures, Kinect Joy Ride, Kinectimals…

[Picture]:

---

Microsoft Research
IE Task: Relation & Event Extraction

- Related terms and entities
- Categorization
- Relation detection and classification
- Event detection and classification
- ...

- Similarity(significantly, substantially, 0.9)
- Synonym(China, People’s Republic of China)
- IsA(pear, fruit)
- Peer(Beijing, Shanghai, Guangzhou…)
- InClass(Beijing, C1)
- BornIn(Barack Obama, 1961)
- BornIn(PERSON, YEAR)
- LocatedIn(ORGANIZATION, LOCATION)
- DefeatedIn(Dallas Mavericks, Miami Heat, 2011 NBA Finals)
Outline

• Overview

➢ Semantic class mining
  • Semantic hierarchy construction
  • Mining attribute names and values
  • General relation extraction
  • Demo
  • Summary
Semantic Class Mining

• Goal
  - Discover peer terms (or coordinate terms)
  - Sample: \{C++, C#, Java, PHP, Perl, \ldots\}

• Main techniques
  - First-order co-occurrences
    • Standard co-occurrences
    • Patterns: Special first-order co-occurrences
  - Second-order co-occurrences
    • Distributional similarity
Pattern-Based (PB)

Hours may vary on holidays, such as Easter, Thanksgiving and Christmas.

Pattern: (such as | including) T {,T}* (and|.|.)

{Easter, Thanksgiving, Christmas}

Exploit first-order Co-occurrences

Pattern:

<select>
<option> T … <option> T
</select>

{Alabama, Alaska, Arizona…}
PB Implementation

- **RASC mining**
  - Employ predefined patterns to extract Raw Semantic Classes (RASCs)

<table>
<thead>
<tr>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical</td>
<td>T {, T}*{,} (and</td>
</tr>
<tr>
<td></td>
<td>(such as</td>
</tr>
<tr>
<td></td>
<td>T, T, T {,T}*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;ul&gt; &lt;li&gt; T &lt;/li&gt; ... &lt;li&gt; T &lt;/li&gt; &lt;/ul&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;ol&gt; &lt;li&gt; T &lt;/li&gt; ... &lt;li&gt; T &lt;/li&gt; &lt;/ol&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;select&gt; &lt;option&gt; T ...&lt;option&gt; T &lt;/select&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;table&gt; &lt;tr&gt; &lt;td&gt; T &lt;/td&gt; ... &lt;td&gt; T &lt;/td&gt; &lt;/tr&gt; ... &lt;/table&gt;</td>
</tr>
<tr>
<td></td>
<td>Other Html-tag repeat patterns</td>
</tr>
</tbody>
</table>
PB Implementation

• Compute Term Similarity
  - Based on the RASCs containing both terms

\[
Sim(a, b) = \sum_{i=1}^{m} \log(1 + \sum_{j=1}^{k_i} w(P(C_{i,j})))
\]

\[
Sim^*(a, b) = Sim(a, b) \cdot \sqrt{IDF(a) \cdot IDF(b)}
\]

\[
IDF(a) = \log(1 + N/N(a))
\]

(Zhang et al., ACL’09)
Distributional Similarity (DS)

- Distributional hypothesis (Harris, 1985): Terms occurring in analogous (lexical or syntactic) contexts tend to be similar

- Contexts shared by *Easter* and *Christmas*
  - the date _ is celebrated
  - _ is a religious festival
  - history of the _ festival
  - ...  

- Contexts shared by *significantly* and *dramatically*
  - is _ improved by
  - unlikely to _ alter the
  - can _ increase health risks
  - ...  

Exploit second-order Co-occurrences
DS Implementation

• Step-1: Define context
  - Syntactic context, lexical context…

• Step-2: Represent each term by a feature vector
  - Feature: A context in which the term appears
  - Feature value: “Weight” of the context w.r.t. the term

• Step 3: Compute term similarity
  - Term similarity = similarity between corresponding feature vectors
### DS Implementation

<table>
<thead>
<tr>
<th>Contexts</th>
<th>Text window (window size: 2, 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Syntactic</td>
</tr>
<tr>
<td>Feature value</td>
<td>PMI</td>
</tr>
<tr>
<td>Similarity measure</td>
<td>Cosine, Jaccard</td>
</tr>
</tbody>
</table>

DS approaches implemented in the study:

Pointwise mutual information:

\[
f_{w,c} = \text{PMI}_{w,c} = \log \frac{F(w,c) \cdot F(*,*)}{F(w,*) \cdot F(*,c)}
\]

\[
\text{Cosine}(\tilde{x}, \tilde{y}) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2 \cdot \sum_i y_i^2}}
\]

\[
\text{Jaccard}(\tilde{x}, \tilde{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i x_i + \sum_i y_i - \sum_i \min(x_i, y_i)}
\]
Compare DS and PB with Set Expansion

- Set Expansion: Problem statement
  - Given a list of seed terms in a semantic class
    \[ Q = \{ s_1, s_2, \ldots, s_k \} \] (e.g. \[ Q = \{ \text{Lent, Epiphany} \} \])
  - To find other members of the class
    - E.g., \{ \text{Advent, Easter, Christmas} \} ...

- Set Expansion with a similarity graph \( G \)
  - Select the terms most similar to the seeds
    \[
    f(t, Q) = \sum_{i=1}^{k} w_i \cdot \text{Sim}(t, s_i)
    \]
    \[
    \text{Sim}(t, s_i) = \frac{1}{\log(\lambda + R(t, s_i))}
    \]
Compare and Combine PB & DS (cont.)

- Corpus: ClueWeb (500 million English pages)
- Five term categories: proper nouns, common nouns, verbs, adjectives, adverbs
- Key observations: PB performs better for proper nouns; DS has better performance for other term categories
Samples (Query: significantly)

<table>
<thead>
<tr>
<th>Sample</th>
<th>PB Result</th>
<th>DS Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>significantly</td>
<td>187.419</td>
<td>0.121576</td>
</tr>
<tr>
<td>and</td>
<td>54.8759</td>
<td>0.0162357</td>
</tr>
<tr>
<td>slightly</td>
<td>23.4412</td>
<td>0.0154982</td>
</tr>
<tr>
<td>but</td>
<td>21.83044</td>
<td>0.0138213</td>
</tr>
<tr>
<td>moderately</td>
<td>21.7083</td>
<td>0.013429</td>
</tr>
<tr>
<td>English</td>
<td>20.4911</td>
<td>0.010923</td>
</tr>
<tr>
<td>seriously</td>
<td>20.4479</td>
<td>0.0089119</td>
</tr>
<tr>
<td>Yiddish</td>
<td>20.2321</td>
<td>0.00800886</td>
</tr>
<tr>
<td>Hebrew</td>
<td>19.7871</td>
<td>0.0074269</td>
</tr>
<tr>
<td>Too</td>
<td>19.6313</td>
<td>0.00731532</td>
</tr>
<tr>
<td>Kigezi</td>
<td>18.4164</td>
<td>0.00688791</td>
</tr>
<tr>
<td>Bunyoro</td>
<td>17.8679</td>
<td>0.00640118</td>
</tr>
<tr>
<td>Specifically</td>
<td>17.8268</td>
<td>0.0061907</td>
</tr>
<tr>
<td>Also</td>
<td>17.4519</td>
<td>0.00606039</td>
</tr>
<tr>
<td>Mbaye</td>
<td>17.3605</td>
<td>0.00604854</td>
</tr>
<tr>
<td>Especially</td>
<td>17.2895</td>
<td>0.0060448</td>
</tr>
<tr>
<td>Rich americans</td>
<td>17.1207</td>
<td>0.00603508</td>
</tr>
<tr>
<td>Surely</td>
<td>16.7604</td>
<td>0.00602135</td>
</tr>
<tr>
<td>Sharply</td>
<td>16.5638</td>
<td>0.00601235</td>
</tr>
<tr>
<td>It</td>
<td>15.6475</td>
<td>0.00590148</td>
</tr>
</tbody>
</table>

PB results

DS results
Samples (Query: Apple)

PB results

|     | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|     | apple | 1741.13 | microsoft | 639.909 | ibm | 617.503 | sony | 613.111 | dell | 601.909 | hp | 597.473 | toshiba | 546.464 | orange | 537.578 | samsung | 528.885 | compaq | 490.275 | canon | 476.098 | cherry | 472.247 | pear | 470.911 |
|     | panasonic | 467.727 | peach | 460.441 | pineapple | 444.158 | intel | 434.583 | acer | 433.825 | lemon | 424.788 | strawberry | 423.942 |     |     |     |     |     |     |

DS results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>apple</td>
<td>0.0808821</td>
<td>microsoft</td>
<td>0.00336825</td>
<td>the government</td>
<td>0.00237455</td>
<td>the company</td>
<td>0.00223547</td>
<td>google</td>
<td>0.00212872</td>
<td>sony</td>
<td>0.00193015</td>
<td>ibm</td>
<td>0.00185744</td>
<td>obama</td>
<td>0.00163117</td>
<td>dell</td>
<td>0.00161188</td>
<td>nintendo</td>
<td>0.00135578</td>
</tr>
</tbody>
</table>
Explain by Frequency

- Normalized frequency ($F_{norm}$) of term $t$
  - Frequency in the RASCs
  - Frequency in the sentences of the original documents
- Mean normalized frequency (MNF) of a query set $S$

$$MNF(S) = \frac{\sum_{t \in S} F_{norm}(t)}{|S|}$$

<table>
<thead>
<tr>
<th>Seed Categories</th>
<th>Terms</th>
<th>MNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proper nouns</td>
<td>40</td>
<td>0.2333</td>
</tr>
<tr>
<td>Common nouns</td>
<td>40</td>
<td>0.0716</td>
</tr>
<tr>
<td>Verbs</td>
<td>40</td>
<td>0.0099</td>
</tr>
<tr>
<td>Adjectives</td>
<td>40</td>
<td>0.0126</td>
</tr>
<tr>
<td>Adverbs</td>
<td>40</td>
<td>0.0053</td>
</tr>
</tbody>
</table>
Related Papers

- Harris, 1985 (in The Philosophy of Linguistics)  
  *Distributional Structure*
- Pantel & Lin, SIGKDD’2002  
  *Discovering Word Senses from Text*
- Etzioni et al., WWW’2004  
  *Web-Scale Information Extraction in KnowItAll*
- Wang & Cohen, ICDM’2008  
  *Iterative Set Expansion of Named Entities Using the Web*
- Pantel, EMNLP’2009  
  *Web-Scale Distributional Similarity and Entity Set Expansion*
- Agirre et al., NAACL’2009  
  *A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches*
- Shi et al., COLING’2010  
  *Corpus-based Semantic Class Mining: Distributional vs. Pattern-Based Approaches*
Outline

- Overview
- Semantic class mining
  - Semantic hierarchy construction
  - Mining attribute names and values
  - General relation extraction
- Demo
- Summary
Semantic Hierarchy Construction
Semantic hierarchy construction

• Major subtasks
  - Assign category labels (hyponyms) to terms
    • Beijing → city, capital…
    • Apple → company, fruit…
    • Red → color, symptom, hue…
    • Canon EOS 400D → digital camera, product…
  - Assign category labels to semantic classes
    • {Beijing, Shanghai, Dalian…} → cities, Chinese cities…
    • {Microsoft, IBM, Apple…} → companies, manufacturers…
  - Build the hierarchy
Subtask: Term → Label

- Approach: Pattern matching + counting

**Tuple:** `<term, label, pattern, source, weight>`

- `<pear, fruit, P1, S1, 1.0>`
- `<pear, shape, P2, S2, 1.0>`
- `<pear, fruit, P3, S3, 1.0>`
- `<New York, city, P1, S4, 1.0>`
- `<New York, office, P2, S6, 1.0>`
- `<New York, state, P4, S7, 1.0>`
  ...
  ...

Corpus  →  Pattern matching  →  Merge tuples  →  Term-Label graph
Subtask: Term → Label (cont.)

- Pattern matching
  - Manually designed or automatically generated patterns
  - Text patterns or HTML tables

<table>
<thead>
<tr>
<th>Label</th>
<th>Label</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>Term</td>
<td>Term</td>
</tr>
<tr>
<td>Term</td>
<td>Term</td>
<td>Term</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

- Output: <term, label, pattern, source, weight> tuples
- Challenges
  - Boundary detection: term boundary, label boundary
  - Label selection

<table>
<thead>
<tr>
<th>Type</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hearst-I</td>
<td>NP_{L} { np } (such as) { np }^* { and</td>
</tr>
<tr>
<td>Hearst-II</td>
<td>NP_{L} { np } (include(s)</td>
</tr>
<tr>
<td>Hearst-III</td>
<td>NP_{L} { np } (e.g.</td>
</tr>
<tr>
<td>IsA-I</td>
<td>NP (is</td>
</tr>
<tr>
<td>IsA-II</td>
<td>NP (is</td>
</tr>
<tr>
<td>IsA-III</td>
<td>NP (is</td>
</tr>
</tbody>
</table>
Subtask: Term $\rightarrow$ Label (cont.)

- **Merge tuples**
  - For each term $T$ and label $L$, compute $w(T, L)$

- **Methods**
  - **Simple counting**
    - Count the number of $<T, L, P, S, W>$ tuples for each $(T, L)$ pair
    - Or TF-IDF
  - **Nonlinear evidence fusion** (Zhang et al., ACL’2011)
    
    $\text{Score}(T, L) = \left( \sum_{i=1}^{K} \sqrt[p]{m_i} \right) \cdot \text{IDF}(L)$

$m_i$: #tuples for pattern $i$
$x_{i,j}$: Gain value given the $j$’th tuple for pattern $i$
Subtask: Class $\rightarrow$ Label

- **Input**
  - Class $C$: \{orange, apple, pear, banana…\}
- **Output**
  - Label list for $C$: fruit, tree, flavor…
- **Method: Voting**
  - orange: color, flavor, client, network, fruit, county, tree…
  - apple: company, brand, fruit, manufacturer, client, tree…
  - pear: fruit, tree, shape, flavor, juice, cut, wood…
  - banana: fruit, crop, flavor, tree, food, plant, vegetable…
Related Papers

• Hearst, COLING’1992
  *Automatic Acquisition of Hyponyms from Large Text Corpora*

• Pantel & Ravichandran, HLT-NAACL’2004
  *Automatically Labeling Semantic Classes*

• Snow et al., COLING-ACL’2006
  *Semantic Taxonomy Induction from Heterogenous Evidence*

• Banko et al., IJCAI’2007
  *Open Information Extraction from the Web*

• Cafarella et al., VLDB’2008
  *WebTables: Exploring the Power of Tables on the Web*

• Durme & Pasca, AAAI’2008
  *Finding cars, Goddesses and Enzymes: Parametrizable Acquisition of Labeled Instances for Open-Domain Information Extraction*

• Zhang et al., ACL’2011
  *Nonlinear Evidence Fusion and Propagation for Hyponymy Relation Mining*
Outline

• Overview
• Semantic class mining
• Semantic hierarchy construction
  ➢ Mining attribute names and values
• General relation extraction
• Demo
• Summary
Semantic Attributes

(city, population)
(country, flag)
(country, capital)
(company, CEO)
(China, capital, Beijing)
(Microsoft, CEO, Steve Ballmer)
(Barack Obama, Birth year, 1961)
**Attribute Name Extraction from Unstructured Text**

- **Corpus (sentences, query logs...)**
- **Semantic class**
- **(term, attr) seeds**
- **Patterns**
- **Pattern generation**
- **Pattern matching**
- **Tuples**
- **Class-Attr graph**
- **Merge tuples**

Sample patterns:
- A of I
- I’s A
- I A

**Tuple:** `<term, attr, pattern, source, weight>`
- `<China, capital, P1, S1, 1.0>`
- `<China, republic, P2, S2, 1.0>`
- `<United States, land area, P3, S3, 1.0>`
Attribute Name Extraction from Unstructured Text

• Major papers:
  - Pasca, WWW’2007
    *Organizing and Searching the World Wide Web of Facts Step Two: Harnessing the Wisdom of the Crowds*
  - Durme et al., COLING’2008
    *Class-Driven Attribute Extraction*
  - Pasca et al., CIKM’2007
    *The Role of Documents vs. Queries in Extracting Class Attributes from Text*
  - Bellare et al., NIPS’2007
    *Lightly-Supervised Attribute Extraction*
  - Reisinger & Pasca, 2009
    *Low-Cost Supervision for Multiple-Source Attribute Extraction*
  - Tokunaga et al., IJCNLP’2005 (Japanese data)
    *Automatic Discovery of Attribute Words from Web Documents*
  - …
Attribute Name & Value Extraction

- From Unstructured Text
  - Similar with extracting attribute names from unstructured text.
  - Changed patterns: A of I → A of I is V; V is A of I
  - Changed seeds: (China, capital) → (China, capital, Beijing)

- From HTML tables

<table>
<thead>
<tr>
<th>Mountain Peak</th>
<th>Continent</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount Everest</td>
<td>Asia</td>
<td>8,850 m</td>
</tr>
<tr>
<td>Aconcagua</td>
<td>South America</td>
<td>6,959 m</td>
</tr>
<tr>
<td>Mount McKinley (Denali)</td>
<td>North America</td>
<td>6,194 m</td>
</tr>
<tr>
<td>Kilimanjaro</td>
<td>Africa</td>
<td>5,895 m</td>
</tr>
<tr>
<td>Mount Elbrus</td>
<td>Europe</td>
<td>5,642 m</td>
</tr>
<tr>
<td>Vinson Massif</td>
<td>Antarctica</td>
<td>4,897 m</td>
</tr>
<tr>
<td>Carstensz Pyramid</td>
<td>Australia - Oceania</td>
<td>4,884 m</td>
</tr>
<tr>
<td>Mount Kosciuszko (The highest point on the Australian landmass)</td>
<td></td>
<td>2,228 m</td>
</tr>
</tbody>
</table>

- From Wikipedia Infobox

http://woodlands-junior.kent.sch.uk/Homework/mountains/tallest.htm
Table Mining References

- G. Limaye, S. Sarawagi, and S. Chakrabarti. PVLDB'2010
  *Annotating and searching web tables using entities, types and relationships*

  *WebTables: Exploring the power of tables on the web*

  WWW'2007
  *Towards domain-independent information extraction from web tables*

- Y. Wang and J. Hu. WWW'2002
  *A machine learning based approach for table detection on the web*

  *Mining tables from large scale HTML texts*

- …
Outline

• Overview
• Semantic class mining
• Semantic hierarchy construction
• Mining attribute names and values
  ➢ **General relation extraction**
• Demo
• Summary
General Relations

• Relations: Facts involving entities
  • [PER Susan Dumais] works for [ORG Microsoft Research], which is headquartered in [LOC Redmond, WA]
  • DefeatedIn(Dallas Mavericks, Miami Heat, 2011 NBA Finals)

• Relations vs. Events
  - Vague boundary

• History
  - Introduced in MUC-7 (1997), extended by ACE, continued by KBP
  - Gain popularity in molecular biology, recent works including extracting protein-protein interaction

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART (artifact)</td>
<td>User-Owner-Inventor-Manufacturer</td>
</tr>
<tr>
<td>GEN-AFF (Gen-affiliation)</td>
<td>Citizen-Resident-Religion-Ethnicity, Org-Location</td>
</tr>
<tr>
<td>METONYMY*</td>
<td>none</td>
</tr>
<tr>
<td>ORG-AFF (Org-affiliation)</td>
<td>Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership</td>
</tr>
<tr>
<td>PART-WHOLE (part-whole)</td>
<td>Artifact, Geographical, Subsidiary</td>
</tr>
<tr>
<td>PER-SOC* (person-social)</td>
<td>Business, Family, Lasting-Personal</td>
</tr>
<tr>
<td>PHYS* (physical)</td>
<td>Located, Near</td>
</tr>
</tbody>
</table>

ACE’05 relation types
General Relations (cont.)

- **Relation triples**
  - <Euro, be the currency of, Germany>
  - <authorship, be the currency of, science>
  - <Euro, be the currency used in, Germany>
  - <Dinar, be legal tender in, Iraq>

- **Concept-level relations**

\[
< \{ \text{marijuana}, \text{caffeine}, \text{nicotine}, \ldots \} >, \quad \{ \text{result in}, \text{be risk factor for}, \text{be major cause of}, \ldots \} >, \quad \{ \text{insomnia}, \text{emphysema}, \text{breast cancer}, \ldots \} >
\]
Supervised Learning

• Treat relation mining as a classification problem
  - Use relational and non-relational mentions as positive and negative data, respectively

• Solve it with supervised Machine learning algorithms
  - Popular choices include SVM, MaxEnt, KNN

• Key: data representation
  - Feature based methods
  - Kernel based methods

• Evaluate metrics: Precision, Recall, F1 on relation mention level
Features

• List of common features (Kambhatla 2004)
  • **Words**: Words of both the entity mentions and all the words in between.
  • **Entity Type**: Entity type of both the mentions.
  • **Mention Level**: Mention level of both the mentions.
  • **Overlap**: Number of words separating the two mentions, number of other mentions in between, flags indicating whether the two mentions are in the same noun phrase, verb phrase or prepositional phrase.
  • **Dependency**: Words and PoS and chunk labels of the words on which the mentions are dependent in the dependency tree
  • **Parse Tree**: Path of non-terminals (removing duplicates) connecting the two mentions in the parse tree, and the path annotated with head words.

• Other features (Zhou et al. 2005)
  • **Based phrase chunking** chunk labels and chunk heads in between
  • **Semantic resources** (country list, etc)
Kernel based Methods

• Kernel \((X, Y)\) defines similarity between \(X\) and \(Y\)
• \(X\) and \(Y\) can be
  - Vectors of features (as in previous slides)
  - Objects (string sequence, Parse trees)

• Kernel-based methods
  - Don’t require extensive feature engineering
  - Maybe computational expensive

• Multiple Kernels can also be used in combination with a composite kernel (Zhao and Grishman, 2005)
Subsequence Kernel (Bunescu and Mooney, 2005)

- Implicit features are sequences of words anchored at the two entity names
  - \( s \) = a word sequence

\[
\begin{align*}
<e_1> & \quad \ldots \quad \text{bought} \quad \ldots \quad <e_2> \quad \ldots \quad \text{billion} \quad \ldots \quad \text{deal}.
\end{align*}
\]

- \( x \) = an example sentence, containing \( s \) as a subsequence

Google has bought video-sharing website YouTube in a controversial $1.6 billion deal.

\[ g_1 = 1 \quad g_2 = 3 \quad g_3 = 4 \quad g_4 = 0 \]

- \( \varphi_s(x) \) = the value of feature \( s \) in example \( x \)

\[
\varphi_s(x) = \lambda \sum g_i = \lambda^{\text{gap}(s,x)} = \lambda^{1+3+4+0}
\]

- \( K(x_1, x_2) = \varphi(x_1) \varphi(x_2) \) = the number of common “anchored” subsequences between \( x_1 \) and \( x_2 \), weighted by their total gap
Tree Kernel for RDC

- Convolution kernels for NLP (Collins and Duffy. 2001)
  - $K(T_1, T_2)$ defined over trees $T_1$ and $T_2$
  - Measured as number of overlapping fragments.

An example parse tree(a) and its sub-trees(b)

- Parse tree needs to be augmented before used for RDC
- Tree kernel for RDC differs in ways to augment/prune trees
Tree kernels for RDC

- An example of pruned parse tree augmented with entity types (Zhang et al. 2006)
Semi-Supervised Learning

- Supervised learning requires sufficient amount of annotated data
  - Expensive to obtain
  - Annotation error still occurs even dual annotated and adjudicated (ACE 2005)
- Semi-supervised learning (SSL) use a handful of seed tuples or patterns
- Bootstrapping alternates between finding pairs of arguments and contexts(pattern) of them
Initial Seed Tuples:

<table>
<thead>
<tr>
<th>ORGANIZATION</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICROSOFT</td>
<td>REDMOND</td>
</tr>
<tr>
<td>IBM</td>
<td>ARMONK</td>
</tr>
<tr>
<td>BOEING</td>
<td>SEATTLE</td>
</tr>
<tr>
<td>INTEL</td>
<td>SANTA CLARA</td>
</tr>
</tbody>
</table>

DIRPRE (Brin 1998) patterns:

<STRING1>‘s headquarters in <STRING2>

Snowball patterns:

<left, NE tag1, middle, NE tag2, right>, left, middle, right are weighted terms

Evaluating Patterns and tuples (Snowball)

\[
Conf(Pat) = \frac{Positive}{Positive + Negative}
\]

\[
Conf(Tuple) = 1 - \prod(1 - Conf(P_i))
\]
Weakly Supervision

- Handful of seeds for supervision

Table 1: Corporate Acquisition Pairs.

<table>
<thead>
<tr>
<th>+/-</th>
<th>Arg a₁</th>
<th>Arg a₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Google</td>
<td>YouTube</td>
</tr>
<tr>
<td>+</td>
<td>Adobe Systems</td>
<td>Macromedia</td>
</tr>
<tr>
<td>+</td>
<td>Viacom</td>
<td></td>
</tr>
<tr>
<td>+</td>
<td>Novartis</td>
<td>Eon Labs</td>
</tr>
<tr>
<td>-</td>
<td>Yahoo</td>
<td>Microsoft</td>
</tr>
<tr>
<td>-</td>
<td>Pfizer</td>
<td>Teva</td>
</tr>
</tbody>
</table>

Bunescu and Mooney, 2007

Figure 1: Sentence examples.

+/S₁: Search engine giant Google has bought video-sharing website YouTube in a controversial $1.6 billion deal.

−/S₂: The companies will merge Google’s search expertise with YouTube’s video expertise, pushing what executives believe is a hot emerging market of video offered over the Internet.

+/S₃: Google has acquired social media company, YouTube for $1.65 billion in a stock-for-stock transaction as announced by Google Inc. on October 9, 2006.

+/S₄: Drug giant Pfizer Inc. has reached an agreement to buy the private biotechnology firm Rinat Neuroscience Corp., the companies announced Thursday.

−/S₅: He has also received consulting fees from Alpharma, Eli Lilly and Company, Pfizer, Wyeth Pharmaceuticals, Rinat Neuroscience, Elan Pharmaceuticals, and Forest Laboratories.
Weakly Supervision (cont.)

- A SVM solution to tolerate noisy positive instances

\[
J(w, b, \xi) = \frac{1}{2} \|w\|^2 + \frac{C}{L} \left( c_p \frac{L_p}{L} \Xi_p + c_n \frac{L_p}{L} \Xi_n \right)
\]

\[
\Xi_p = \sum_{X \in X_p} \sum_{x \in X} \xi_x
\]

\[
\Xi_n = \sum_{X \in X_n} \sum_{x \in X} \xi_x
\]

subject to:

\[
w \phi(x) + b \geq 1 - \xi_x, \quad \forall x \in X \in X_p
\]

\[
w \phi(x) + b \leq -1 + \xi_x, \quad \forall x \in X \in X_n
\]

\[
\xi_x \geq 0
\]

Use a lower penalize factor for positive errors to tolerate noises from positive instances.
Unsupervised Learning

- Automatically find major relations and respective arguments
- builds on the same duality of name pairs and contexts as relation bootstrapping methods

Hasegawa et al. 2004
- Uses Sekine’s Extended NE tagger
- A domain is defined as a pair of name classes
- Bag-of-words features to model relational context
- hierarchical clustering
References for General Relation Mining

• ACE, http://www.itl.nist.gov/iad/mig/tests/ace/
• Nanda Kambhatla. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction. ACL 2004
• GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring Various Knowledge in Relation Extraction. ACL 2005
• Shubin Zhao and Ralph Grishman. Extracting Relations withh Integrated Information Using Kernel Methods. ACL 2005
References for General Relation Mining (cont.)

- Min ZHANG, Jie ZHANG, Jian SU, Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel, In ACL 2006.
- Razvan Bunescu and Raymond J. Mooney. Learning to Extract Relations from the Web using Minimal Supervision. ACL 2007
- Takaaki Hasegawa, Satoshi Sekine, Ralph Grishman Discovering Relations among Named Entities from Large Corpora. ACL 2004.
Outline

• Overview
• Semantic class mining
• Semantic hierarchy construction
• Mining attribute names and values
• General relation extraction
  ➢ Demo
  ➢ Summary
“Find needles in a haystack”

- Mine open-domain semantic knowledge from web-scale data
- Empower upper-layer applications with semantic knowledge

URL: http://needleseek.msra.cn
Semantic IE: **Summary**

- **Semantic class mining**
  - Sample: \{C++, C#, Java, PHP, Perl, ...\}
  - Methods: Pattern matching (1st-order co-occurrences); distributional similarity (2nd-order co-occurrences)

- **Semantic hierarchy construction**
  - Key task: Hypernymy extraction (Beijing \(\rightarrow\) city; pear \(\rightarrow\) fruit; pear \(\rightarrow\) shape)
  - Pattern matching; tuple aggregation; Label voting

- **Mining attribute names and values**
  - Samples: (company, CEO); (China, capital, Beijing)
  - Pattern learning; pattern matching; Table extraction; Wikipedia Infobox

- **General relation & event extraction**
  - Sample: WorkFor(Susan Dumais, Microsoft Research)
  - Supervised, semi-supervised, & unsupervised learning
  - Process contexts (especially middle contexts)