SI485i : NLP

Set 13
Information Extraction
“Yesterday GM released third quarter results showing a 10% in profit over the same period last year.”

“John Doe was convicted Tuesday on three counts of assault and battery.”

“Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.”
Why Information Extraction

1. You have a desired relation/fact you want to monitor.
   • Profits from corporations
   • Actions performed by persons of interest

2. You want to build a question answering machine
   • Users ask questions (about a relation/fact), you extract the answers.

3. You want to learn general knowledge
   • Build a hierarchy of word meanings, dictionaries on the fly (is-a relations, WordNet)

4. Summarize document information
   • Only extract the key events (arrest, suspect, crime, weapon, etc.)
Current Examples

- Fact extraction about people. Instant biographies.
  - Search “tom hanks” on google

- Never-ending Language Learning
  - http://rtw.ml.cmu.edu/rtw/

Tom Hanks

Thomas Jeffrey "Tom" Hanks is an American actor, producer, writer, and director. Hanks is known for his roles in Apollo 13, Big. That Thing You Do!, The Green Mile, You've Got Mail, Sleepless in Seattle, ... Wikipedia

- Born: July 9, 1956 (age 56), Concord
- Parents: Amos Mefford Hanks, Janet Marylyn Frager
- Children: Colin Hanks, Chet Hanks, Elizabeth Ann Hanks, Truman Theodore Hanks
- Spouse: Rita Wilson (m. 1988), Samantha Lewes (m. 1978–1987)
- Siblings: Jim Hanks, Larry Hanks, Sandra Hanks
Extracting structured knowledge

Each article can contain hundreds or thousands of items of knowledge...

“The Lawrence Livermore National Laboratory (LLNL) in Livermore, California is a scientific research laboratory founded by the University of California in 1952.”

LLNL EQ Lawrence Livermore National Laboratory
LLNL LOC-IN California
Livermore LOC-IN California
LLNL IS-A scientific research laboratory
LLNL FOUNDED-BY University of California
LLNL FOUNDED-IN 1952
Relation extraction: 5 easy methods

1. Hand-built patterns
2. Supervised methods
3. Bootstrapping (seed) methods
4. Unsupervised methods
5. Distant supervision
Adding hyponyms to WordNet

- Intuition from Hearst (1992)
  - “Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use”
- What does *Gelidium* mean?
- How do you know?`
Adding hyponyms to WordNet

- Intuition from Hearst (1992)
  - “Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use”
- What does Gelidium mean?
- How do you know?
Predicting the hyponym relation

“...works by such authors as Herrick, Goldsmith, and Shakespeare.”

“If you consider authors like Shakespeare...”

“Some authors (including Shakespeare)…”

“Shakespeare was the author of several…”

“Shakespeare, author of The Tempest…”

\[
\text{Shakespeare IS-A author (0.87)}
\]

How can we capture the variability of expression of a relation in natural text from a large, unannotated corpus?
Hearst’s lexico-syntactic patterns

“Y such as X ((, X)* (, and/or) X)”
“such Y as X…”
“X… or other Y”
“X… and other Y”
“Y including X…”
“Y, especially X…”

(Hearst, 1992): Automatic Acquisition of Hyponyms
## Examples of Hearst patterns

<table>
<thead>
<tr>
<th>Hearst pattern</th>
<th>Example occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>X and other Y</td>
<td>...temples, treasuries, and other important civic buildings.</td>
</tr>
<tr>
<td>X or other Y</td>
<td>bruises, wounds, broken bones or other injuries...</td>
</tr>
<tr>
<td>Y such as X</td>
<td>The bow lute, such as the Bambara ndang...</td>
</tr>
<tr>
<td>such Y as X</td>
<td>...such authors as Herrick, Goldsmith, and Shakespeare.</td>
</tr>
<tr>
<td>Y including X</td>
<td>...common-law countries, including Canada and England...</td>
</tr>
<tr>
<td>Y, especially X</td>
<td>European countries, especially France, England, and Spain...</td>
</tr>
</tbody>
</table>
Patterns for detecting part-whole relations (meronym-holonym)

Berland and Charniak (1999)

<table>
<thead>
<tr>
<th>Berland pattern</th>
<th>Example occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NP_Y$’s $NP_X$:</td>
<td>...building’s basement...</td>
</tr>
<tr>
<td>$NP_X$ of {the</td>
<td>a} $NP_Y$:</td>
</tr>
<tr>
<td>$NP_X$ in {the</td>
<td>a} $NP_X$:</td>
</tr>
<tr>
<td>$NP_X$ of $NP_Y$:</td>
<td>...basements of buildings...</td>
</tr>
<tr>
<td>$NP_X$ in $NP_Y$:</td>
<td>...basements in buildings...</td>
</tr>
</tbody>
</table>
Results with hand-built patterns

- Hearst: hypernyms
  - 66% precision with “X and other Y” patterns

- Berland & Charniak: meronyms
  - 55% precision
Problem with hand-built patterns

• Requires that we hand-build patterns for each relation!
• Don’t want to have to do this for all possible relations!
• Plus, we’d like better accuracy
Relation extraction: 5 easy methods

1. Hand-built patterns
2. **Supervised methods**
3. Bootstrapping (seed) methods
4. Unsupervised methods
5. Distant supervision
Supervised relation extraction

- Sometimes done in 3 steps:
  1. Find all pairs of named entities
  2. Decide if the two entities are related
  3. If yes, then classify the relation

- Why the extra step?
  - Cuts down on training time for classification by eliminating most pairs
  - Producing separate feature-sets that are appropriate for each task
Relation extraction

- Task definition: to label the semantic relation between a pair of entities in a sentence (fragment)

...[leader arg-1] of a minority [government arg-2]...

located near  
Personal relationship  
employed by  
NIL
Supervised learning

- Extract features, learn a model ([Zhou et al. 2005], [Bunescu & Mooney 2005], [Zhang et al. 2006], [Surdeanu & Ciaramita 2007])

- Training data is needed for each relation type
We have competitions with labeled data

ACE 2008: six relation types

<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 (General 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-WHITE (part-to-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>
**Features: words**

*American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.*

**Bag-of-words features**
WM1 = \{American, Airlines\}, WM2 = \{Tim, Wagner\}

**Head-word features**
HM1 = Airlines, HM2 = Wagner, HM12 = Airlines+Wagner

**Words in between**
WBNULL = false, WBFL = NULL, WBF = a, WBL = spokesman,
WBO = \{unit, of, AMR, immediately, matched, the, move\}

**Words before and after**
BM1F = NULL, BM1L = NULL, AM2F = said, AM2L = NULL

Word features yield good precision, but poor recall
American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.

**Named entity types** (ORG, LOC, PER, etc.)
ET1 = ORG, ET2 = PER, ET12 = ORG-PER

**Mention levels** (NAME, NOMINAL, or PRONOUN)
ML1 = NAME, ML2 = NAME, ML12 = NAME+NAME

Named entity type features help recall a lot
Mention level features have little impact
American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.

Number of mentions and words in between
#MB = 1, #WB = 9

Does one mention include in the other?
M1>M2 = false, M1<M2 = false

Conjunctive features
ET12+M1>M2 = ORG-PER+false
ET12+M1<M2 = ORG-PER+false
HM12+M1>M2 = Airlines+Wagner+false
HM12+M1<M2 = Airlines+Wagner+false

These features hurt precision a lot, but also help recall a lot
American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.

Parse using the Stanford Parser, then apply Sabine Buchholz’s chunklink.pl:

[NP American Airlines], [NP a unit] [PP of] [NP AMR], [ADVP immediately] [VP matched] [NP the move], [NP spokesman Tim Wagner] [VP said].
Features: base phrase chunking

[\text{NP American Airlines}], [\text{NP a unit}] [\text{PP of}] [\text{NP AMR}], [\text{ADV immediately}] [\text{VP matched}] [\text{NP the move}], [\text{NP spokesman Tim Wagner}] [\text{VP said}].

Phrase heads before and after

CPHBM1F = NULL, CPHBM1L = NULL, CPHAM2F = said, CPHAM2L = NULL

Phrase heads in between

CPHBNULL = false, CPHBFL = NULL, CPHBF = unit, CPHBL = move
CPHBO = \{of, AMR, immediately, matched\}

Phrase label paths

CPP = [\text{NP, PP, NP, ADVP, VP, NP}]
CPPPH = NULL

These features increased both precision & recall by 4-6%
Features: syntactic features

Features of mention dependencies
ET1DW1 = ORG:Airlines
H1DW1 = matched:Airlines
ET2DW2 = PER:Wagner
H2DW2 = said:Wagner

Features describing entity types and dependency tree
ET12SameNP = ORG-PER-false
ET12SamePP = ORG-PER-false
ET12SameVP = ORG-PER-false

These features had disappointingly little impact!
Features: syntactic features

Phrase label paths

PTP = [NP, S, NP]
PTPH = [NP:Airlines, S:matched, NP:Wagner]

These features had disappointingly little impact!
Feature examples

American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said.

<table>
<thead>
<tr>
<th>Entity-based features</th>
<th>ORG</th>
<th>airlines</th>
<th>PERS</th>
<th>Wagner</th>
<th>ORGPERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity\textsubscript{1} type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity\textsubscript{1} head</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity\textsubscript{2} type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity\textsubscript{2} head</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concatenated types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Word-based features           |          |          |          |          |         |
| Between-entity bag of words   | \{ a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman \} |
| Word(s) before Entity\textsubscript{1} | NONE     |          |          |          |         |
| Word(s) after Entity\textsubscript{2} | said     |          |          |          |         |

| Syntactic features            |          |          |          |          |         |
| Constituent path              | NP \uparrow NP \uparrow S \uparrow S \downarrow NP |
| Base syntactic chunk path     | NP \rightarrow NP \rightarrow PP \rightarrow NP \rightarrow VP \rightarrow NP \rightarrow NP |
| Typed-dependency path         | Airlines \leftarrow_{subj} matched \leftarrow_{comp} said \rightarrow_{subj} Wagner |
Classifiers for supervised methods

Use any classifier you like:

- Naïve Bayes
- MaxEnt
- SVM
- etc.

[Zhou et al. used a one-vs-many SVM]
## Sample results

Surdeanu & Ciaramita 2007

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART</td>
<td>74</td>
<td>34</td>
<td>46</td>
</tr>
<tr>
<td>GEN-AFF</td>
<td>76</td>
<td>44</td>
<td>55</td>
</tr>
<tr>
<td>ORG-AFF</td>
<td>79</td>
<td>51</td>
<td>62</td>
</tr>
<tr>
<td>PART-WHOLE</td>
<td>77</td>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>PER-SOC</td>
<td>88</td>
<td>59</td>
<td>71</td>
</tr>
<tr>
<td>PHYS</td>
<td>62</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>TOTAL</td>
<td>76</td>
<td>43</td>
<td>55</td>
</tr>
</tbody>
</table>
Relation extraction: summary

• Supervised approach can achieve high accuracy
  • At least, for *some* relations
  • If we have lots of hand-labeled training data

• Significant limitations!
  • Labeling 5,000 relations (+ named entities) is expensive
  • Doesn’t generalize to different relations
Relation extraction: 5 easy methods

1. Hand-built patterns
2. Supervised methods
3. **Bootstrapping (seed) methods**
4. Unsupervised methods
5. Distant supervision
Bootstrapping approaches

• If you don’t have enough annotated text to train on...
• But you do have:
  • some *seed instances* of the relation
  • (or some patterns that work pretty well)
  • and lots & lots of *unannotated text* (e.g., the web)
• ... can you use those seeds to do something useful?

• Bootstrapping can be considered *semi-supervised*
Bootstrapping example

- Target relation: *product-of*
- Seed tuple: <Apple, iphone>
- Grep (Google) for “Apple” and “iphone”
  - “Apple released the iphone 3G....”
    - X released the Y
  - “Find specs for Apple’s iphone”
    - X’s Y
  - “iphone update rejected by Apple”
    - Y update rejected by X
- Use those patterns to grep for new tuples
Bootstrapping à la Hearst

- Choose a lexical relation, e.g., hypernymy
- Gather a set of pairs that have this relation
- Find places in the corpus where these expressions occur near each other and record the environment
- Find the commonalities among these environments and hypothesize that common ones yield patterns that indicate the relation of interest

Shakespeare and other authors metals such as tin and lead such diseases as malaria regulators including the SEC

X and other Ys Ys such as X such Ys as X Ys including X
Bootstrapping relations

There are weights at every step!!
DIPRE (Brin 1998)

- Extract <author, book> pairs
- Start with these 5 seeds

<table>
<thead>
<tr>
<th>Author</th>
<th>Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isaac Asimov</td>
<td>The Robots of Dawn</td>
</tr>
<tr>
<td>David Brin</td>
<td>Startide Rising</td>
</tr>
<tr>
<td>James Gleick</td>
<td>Chaos: Making a New Science</td>
</tr>
<tr>
<td>Charles Dickens</td>
<td>Great Expectations</td>
</tr>
<tr>
<td>William Shakespeare</td>
<td>The Comedy of Errors</td>
</tr>
</tbody>
</table>

- Learn these patterns:

<table>
<thead>
<tr>
<th>URL Prefix</th>
<th>Text Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.sff.net/locus/c">www.sff.net/locus/c</a>.*</td>
<td>&lt;LI&gt;&lt;B&gt;title&lt;/B&gt; by author (</td>
</tr>
<tr>
<td>dns.city-net.com/~lmann/awards/hugos/1984.html</td>
<td>&lt;i&gt;title&lt;/i&gt; by author (</td>
</tr>
<tr>
<td>dolphin.upenn.edu/~dcummins/texts/sf-award.htm</td>
<td>author</td>
</tr>
</tbody>
</table>

- Now iterate, using these patterns to get more instances and patterns...
**Snowball (Agichtein & Gravano 2000)**

New idea: require that X and Y be named entities of particular types

<table>
<thead>
<tr>
<th>Organization</th>
<th>Location of Headquarters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>Redmond</td>
</tr>
<tr>
<td>Exxon</td>
<td>Irving</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>

```plaintext
ORGANIZATION  's 0.4 headquarters 0.4 in 0.1 LOCATION
LOCATION -0.75 based 0.75 ORGANIZATION
```
Bootstrapping problems

- Requires seeds for each relation
  - Sensitive to original set of seeds
- Semantic drift at each iteration
- Precision tends to be not that high
- Generally, lots of parameters to be tuned
- Don’t have a probabilistic interpretation
  - Hard to know how confident to be in each result
Relation extraction: 5 easy methods

1. Hand-built patterns
2. Supervised methods
3. Bootstrapping (seed) methods
4. Unsupervised methods
5. Distant supervision

No time to cover these. These assume we don’t have seed examples, nor labeled data. How do we extract what we don’t know is there? Lots of interesting work! Including Dr. Chambers’ research!