SI486m : 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.”

GM profit-increase 10%

John Doe convict-for assault

Gelidium is-a algae
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/
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.”

<table>
<thead>
<tr>
<th>LLNL</th>
<th>EQ Lawrence Livermore National Laboratory</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLNL</td>
<td>LOC-IN California</td>
</tr>
<tr>
<td>Livermore</td>
<td>LOC-IN California</td>
</tr>
<tr>
<td>LLNL</td>
<td>IS-A scientific research laboratory</td>
</tr>
<tr>
<td>LLNL</td>
<td>FOUNDED-BY University of California</td>
</tr>
<tr>
<td>LLNL</td>
<td>FOUNDED-IN 1952</td>
</tr>
</tbody>
</table>
Goal: machine-readable summaries

Textual abstract: Summary for human

Structured knowledge extraction: Summary for machine
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 ${\text{the</td>
<td>a}}$ $NP_Y$:</td>
</tr>
<tr>
<td>$NP_X$ in ${\text{the</td>
<td>a}}$ $NP_X$:</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
Exercise: coach-of relation

- What patterns will identify the coaches of teams?
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 pairs of named entities in text
  2. Decide if the two entities are related \textit{at all}
  3. If yes, then decide on the proper relation between them

- Why the extra step 2?
  - Cuts down on training time for classification by eliminating most pairs.
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]...
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-WHOLE (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>
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
FIRST = a, LAST = 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:

```
0 B-NP NNP American NOFUNC Airlines 1 B-S/B-S/B-NP/B-NP
1 I-NP NNPS Airlines NP matched 9 I-S/I-S/I-NP/I-NP
2 O COMMA COMMA NOFUNC Airlines 1 I-S/I-S/I-NP
3 B-NP DT a NOFUNC unit 4 I-S/I-S/I-NP/B-NP/B-NP
4 I-NP NN unit NP Airlines 1 I-S/I-S/I-NP/I-NP/I-NP
5 B-PP IN of PP unit 4 I-S/I-S/I-NP/I-NP/B-PP
6 B-NP NNP AMR NP of 5 I-S/I-S/I-NP/I-NP/I-PP/B-NP
7 O COMMA COMMA NOFUNC Airlines 1 I-S/I-S/I-NP
8 B-ADVP RB immediately ADVP matched 9 I-S/I-S/B-ADVP
9 B-VP VBD matched VP/S matched 9 I-S/I-S/B-VP
10 B-NP DT the NOFUNC move 11 I-S/I-S/I-VP/I-NP
11 I-NP NN move NP matched 9 I-S/I-S/I-VP/I-NP
12 O COMMA COMMA NOFUNC matched 9 I-S
13 B-NP NN spokesman NOFUNC Wagner 15 I-S/B-NP
14 I-NP NNP Tim NOFUNC Wagner 15 I-S/I-NP
15 I-NP NNP Wagner NP matched 9 I-S/I-NP
16 B-VP VBD said VP matched 9 I-S/B-VP
17 O . . NOFUNC matched 9 I-S
```

\[[np American Airlines], [np a unit] [pp of] [np AMR], [advp immediately] [vp matched] [np the move], [np spokesman Tim Wagner] [vp said].\]
Features: base phrase chunking

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

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 = [NP, PP, NP, ADVP, VP, NP]
- CPPH = 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!
American Airlines a unit of AMR immediately matched the move spokesman Tim Wagner said

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(_1) type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity(_1) head</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity(_2) type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity(_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>

<table>
<thead>
<tr>
<th>Word-based features</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Between-entity bag of words</td>
<td>{a, unit, of, AMR, Inc., immediately, matched, the, move, spokesman}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word(s) before Entity(_1)</td>
<td>NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word(s) after Entity(_2)</td>
<td>said</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntactic features</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constituent path</td>
<td>NP ↑ NP ↑ S ↑ S ↓ NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base syntactic chunk path</td>
<td>NP → NP → PP → NP → VP → NP → NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Typed-dependency path</td>
<td>Airlines ← subj matched ← comp said → subj Wagner</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Classifiers for supervised methods

Use any classifier you like:

- Naïve Bayes
- MaxEnt
- Support Vector Machine
- 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 (or Google search) 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/~1mann/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>

'\text{'s}_{0.4} \text{headquarters}_{0.4} \text{in}_{0.1} \text{LOCATION}'

'\text{LOCATION}_{-0.75} \text{based}_{0.75} \text{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!