Final Term Review

Semantics

Information Extraction
Semantics

• Concepts
• Word Similarities based on thesauri
• Word Vectors (Sparse, Dense)
• Semantic Role Labeling
Concepts

• Word Meanings
  • Homonymy (Bank: financial institution, river bank)
  • Polysemy (Bank: financial institution, bank building)
  • Metonymy (Bank or School: organization, building)

• Word Relations
  • Synonyms (big / large)
  • Antonyms (big / small)
  • Hyponym (car is a hyponym of vehical)
  • Hypernym (vehical is a hyponym of car)
  • Instance (College Station is a town)
Word Similarity based on Thesauri

• Path based, 1/pathlen

• Information Content, IC (LCS (c1, c2)), -\log P(LCS (c1, c2)) (Resnik)

• Improved Information Content, considering both commonality and differences, \frac{2\log P(LCS (c1, c2))}{\log P(c1) + \log P(c2)} (Dekang Lin)
Word Vectors

• Distributional vectors (sparse)
  • Term-document matrix -> term-term matrix
  • Frequency -> PPMI, $\log(p(w_1,w_2)/p(w_1)*p(w_2))$
  • Similarity: Cosine of two word vectors

• Dense vectors
  • Singular Value Decomposition
  • Prediction-based
  • Brown clustering
Semantic Role Labeling

• Semantic roles (thematic roles): the abstract role that arguments of a predicate can take wrt the event represented by the predicate.
• Agent, theme, source, target ...
• Propbank, framenet
A simple modern algorithm

\begin{verbatim}
function SEMANTICROLELABEL(words) returns labeled tree

    parse ← PARSE(words)
    for each predicate in parse do
        for each node in parse do
            featurevector ← EXTRACTFEATURES(node, predicate, parse)
            CLASSIFYNODE(node, featurevector, parse)

\end{verbatim}
Information Extraction

• Semantic Lexicon Induction
• Relation Extraction
• Coreference resolution
• Event Extraction
Semantic Lexicon Induction

• Syntactic Heuristics
• Co-occurrence based Bootstrapping
• Mutual bootstrapping
# Syntactic Heuristics for Learning Semantic Labels

<table>
<thead>
<tr>
<th>Conjunctions</th>
<th>lions and tigers and bears</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lists</td>
<td>lions, tigers, bears</td>
</tr>
<tr>
<td>Appositives</td>
<td>the horse, a stallion</td>
</tr>
<tr>
<td>Predicate Nominals</td>
<td>the wolf is a mammal</td>
</tr>
<tr>
<td>Compound nouns</td>
<td>tuna fish</td>
</tr>
<tr>
<td></td>
<td>Honda Sedan</td>
</tr>
</tbody>
</table>

[Riloff & Shepherd 97; Roark & Charniak 98; Phillips & Riloff 02; etc.]

<table>
<thead>
<tr>
<th>Hyponym patterns</th>
<th>dogs such as beagles and boxers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dogs, including beagles and boxers</td>
</tr>
</tbody>
</table>

[Hearst 92; KnowItAll (U.Washington), Kozareva et al. 2008; etc.]
Bootstrapping Semantic Lexicons

Unannotated Texts

Co-occurrence Statistics

prospective category words

Ex: dog, cat, lion, lizard, snake

N best words

Ex: terrier, poodle, tiger, frog, iguana
Mutual Bootstrapping [Riloff & Jones 99]

Unannotated Texts

Ex: dog, cat, lion, lizard, snake

Best Extraction Pattern

Ex: <NP> growled

Ex: Rottweiler, terrier, cougar

Extractions (Nouns)
How to build relation extractors

1. Hand-written patterns
2. Supervised machine learning
3. Semi-supervised and unsupervised
   - Bootstrapping (using seeds)
   - Distant supervision
   - Unsupervised learning from the web
Two different things...

• Anaphora
  – Text
  – World

• (Co)Reference
  – Text
  – World
Kinds of Models

• Mention Pair models
  – Treat coreference chains as a collection of pairwise links
  – Make independent pairwise decisions and reconcile them in some way (e.g. clustering or greedy partitioning)

• Mention ranking models
  – Explicitly rank all candidate antecedents for a mention

• Entity-Mention models
  – A cleaner, but less studied, approach
  – Posit single underlying entities
  – Each mention links to a discourse entity [Pasula et al. 03], [Luo et al. 04]
Patterns/Rules vs. Sequence Tagging

Two general approaches to IE:

*Pattern-based systems* use patterns or rules that are applied to text.

*Sequence tagging models* classify individual tokens as to whether or not they should be extracted.
[The World Trade Center], [an icon] of [New York City], was horrifically attacked on [an otherwise beautiful day] in [September 2001] by [Al Qaeda].

Shallow Parser

Syntactic Templates

Extraction Patterns:

<subj> was attacked
icon of <np>
was attacked on <np>
was attacked in <np>
was attacked by <np>
AutoSlog-TS (Step 2)

Relevant

Irrelevant

Extraction Patterns:  
<subj> was attacked
icon of <np>
was attacked on <np>
was attacked in <np>
was attacked by <np>

<table>
<thead>
<tr>
<th>Extraction Patterns</th>
<th>Freq</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subj&gt; was attacked</td>
<td>100</td>
<td>.90</td>
</tr>
<tr>
<td>icon of &lt;np&gt;</td>
<td>5</td>
<td>.20</td>
</tr>
<tr>
<td>was attacked on &lt;np&gt;</td>
<td>80</td>
<td>.79</td>
</tr>
<tr>
<td>was attacked in &lt;np&gt;</td>
<td>85</td>
<td>.87</td>
</tr>
<tr>
<td>was attacked by &lt;np&gt;</td>
<td>95</td>
<td>.95</td>
</tr>
</tbody>
</table>