Open Domain Relation Extraction

David Israel
SRI (Emeritus)
Sapienza (Visiting)
Distant/Indirect/Weak Supervision

• Simple empirical hypothesis: If two named entities are in a certain relation, then any sentence containing names for both of them is likely to express that relation

• SO, use a database or set of db’s (or something like that) to get lots of examples of (named) entities standing in various relations
  – Instead of using hand-labeled corpus
  – Instead of using a small sample of hand-created or hand-chosen seeds
In a Bit More Detail

• For each pair of entities in a large database,
  – Find sentences containing names of both in a large corpus
  – Extract lots of (typically noisy) features from those sentences
  – Combine those features in a classifier
Benefits of the Approach

• Over Supervised Approaches:
  – Can leverage a rich, reliable hand-crafted knowledge (in db’s)
  – Relations are named via db schemata
  – Can do expensive analysis over small-ish resulting corpus

• Over Unsupervised Approaches:
  – Not sensitive to training corpus
  – Can do expensive analysis over small-ish resulting corpus
First Example: Hypernyms

- Use WordNet as the DB for hyponym-hypernym relation
- Note: this is a special case, because most of the actual words are not names!
  - E.g. “Shakespeare” vs. “author”, “play-write”
- Collect noun pairs from a large corpus of sentences
  - 752,311 pairs from 6Million sentences of newswire
- Is the pair in a Hypo-Hyper relation in WordNet?
  - 14,387 Yes; 737,924 No
- Parse the sentences
- Extract patterns/features
- Train classifier on patterns/features (many, many features: 70K)
One of 70K Patterns

• Pattern: <superordinate> called <subordinate>
• Learned from, e.g.,
  – an uncommon bone cancer called osteogenic sarcoma
  – heavy water rich in the doubly heavy hydrogen atom called deuterium.
• New pairs discovered
  – and a condition called efflorescence are other ...
  – The company, now called O’Neal Inc., was sole distributor
  – run a small ranch called the Hat Creek Outfit.
  – infected by the AIDS virus, called HIV-1
  – local sightseeing attraction called the Bateau Mouche...
What About Other Relations?

• Use Freebase!
  – 102 relations
  – 940,000 entities
  – 1.8 Million instances
• For corpus: Wikipedia
  – 1.8 Million articles
  – 25.7 Million sentences
Example

• Freebase:
  – Founder: (Bill Gates, Microsoft)
  – Founder: (Larry Page, Google)

• Wikipedia Text:
  – Bill Gates founded Microsoft in 1975 ...
  – Bill Gates, founder of Microsoft,...
  – Google was founded by Larry Page ...
Example (cont.)

- Text: Bill Gates founded Microsoft in 1975
- FB: Founder(Bill Gates, Microsoft)
- Training data:
  - (Bill Gates, Microsoft)
  - Label: Founder
  - Feature: X founded Y
- Text: Bill Gates, founder of Microsoft
- New Training Data:
  - (Bill Gates, Microsoft)
  - Label: Founder
  - Feature: X founded Y
  - Feature: X, founder of Y
Problem of Negative Data

• You can’t train a classifier with only positive data!
• You need negative training data, too!
• Solution: Sample 1% of (FB-) unrelated pairs of entities
• Text from corpus
  – Larry Page took a swipe at Microsoft
  – Google is Bill Gates’ worst fear

• Training data:
  – (Larry Page, Microsoft)
  – Label: NO_RELATION
  – Feature: X tok a swipe at Y
  – (Bill Gates, Google)
  – Label: NO_RELATION
  – Feature: Y is X’s worst fear
Advantage of This Approach

• Label entities, often named entities, not sentences
• Makes use of multiple occurrence of entity-mentions
• E.g. if a pair of entity mentions, $N_1$ and $N_2$ appear together in 10 sentences and each sentence has 5 features extracted from it, the pair $<N_1, N_2>$ will have 50 associated features
Experimental SetUp

• 1.8 Million relation instances used for training
• 800,000 Wikipedia articles used for training
• 400,000 articles for testing
• Only extract relation instances not already in FREEBASE
• Results? Very Good!
• See papers for details
Summary on Distant Supervision for Relation Extraction

• Extracts high-precision patterns for a wide variety of relations
• Can make use of 1000X more data than more directly supervised approaches
• Syntactic (dependency-parse) features almost always a help
• But of course, lexical+syntactic features does best
Unsupervised Methods

- KnowItAll (Etzioni, et al. 2005)
- Like Bootstrapping, but more fully automated
- Start with target class and relation labels
- Use patterns to find instances of classes:
  - X and other Y’s; such Y’s as X
  - A la Matti Hearst (“Automatic acquisition of hyponyms from large text corpora”, 1992)
- Now use pattern templates to find relations
  - X is the RELATION (to/of) Y
  - X, RELATION (to/of) Y
- Assess candidates using a variant (?) of PMI
TextRunner (ReVerb)
Banko et al. (2007)

• Self-supervised learner:
  – automatically labels +/- examples and learns a crude relation extractor

• Single-pass extractor:
  – makes one pass over the corpus, extracting candidate relations in each sentence

• Redundancy-based assessor:
  – assigns a probability to each extraction, based on frequency counts
Self-Supervised Learner

• Run a dependency parser over a small-ish sample (2000) of sentences
  – For each pair of base noun phrases NP\(_i\), NP\(_j\): extract all tuples: \(<\text{NP}_i \text{ relation}_{ij} \text{NP}_j>\)

• Label each tuple based on features of the parse
  – Positive (+) iff the dependency path between NPs is short; doesn’t cross a clause boundary and neither NP is a pronoun

• Train a (Naïve Bayes) classifier on the labeled tuples
  – Using \textit{lightweight} features like POS tags, STOP words, etc.
Single-Pass Extractor

• Over a huge (Web-size) corpus
  – Run a simple POS tagger
  – Run a simple Base NP chunker
  – Extract all text strings between base NPs
  – Run heuristic rules to simplify text strings
    • “Scientists from many universities are intently studying stars” → <scientists, are studying, stars>

• Pass candidate tuples to classifier
• Save only those predicted to be “good”
Redundancy-based Assessment

• Collect counts for each candidate tuple:
  – <scientists, are studying, stars> → 17

• Compute likelihood of each tuple
  – Given the counts for each relation
  – The number of sentences
  – And a combinatorial balls-and-urn model
TextRunner Results

• From corpus of 9M Web pages, with 133M sentences
• Extracted 60.5 M tuples
• Evaluation (of 11.3 Million)
  – Three categories
  – Not well-formed: <demands, of securing, border>: 3.5 M
  – Abstract: <executive, hired by, company>: 6.8 M, of which almost 80% were correct
  – Concrete: <Tesla, invented, coil transformer>: 1 M, of which 88% were correct
DIRT

• Discovery of Inference Rules from Text
• Starts from MINIPAR (Lin’s dependency-parser) paths between noun pairs:
  – X finds solution Y
  – N:subj:V←find→V:obj:N→solution→N:to:N
• Applies extended distributional hypothesis: if two paths tend to occur in similar contexts, the meanings of the two paths tend to be similar
• Defines path similarity in terms of co-occurrence counts with various slot fillers
• Thus, extends ideas of Lin from words to paths
Examples

• Top-10 paths most similar to $X$ solves $Y$:
  
  Y is solved by X
  X resolves Y
  X finds a solution to Y
  X tries to solve Y
  X deals with Y
  Y is resolved by X
  X addresses Y
  X seeks a solution to Y
  X do something about Y
  X solution to Y
Problem of Ambiguous Paths

• **X addresses Y**
  – I addressed my letter to him personally.
  – She addressed an audience of Shawnee chiefs.
  – Will Congress finally address the immigration issue?

• **X tackles Y**
  – Foley tackled the quarterback in the endzone.
  – Police are beginning to tackle rising crime.
Yao, Riedel, McCallum 2012

Unsupervised Relation Discovery with Sense Disambiguation

• Induce clusters of dependency paths that express the same semantic relation (like DIRT), but
• Improve upon DIRT by handling the ambiguity of individual paths

• Approach:
  – Extract tuples: \(<entity, path, entity>\)
  – Construct feature repns of every tuple
  – Group tuples for each path into sense clusters
  – Cluster the sense clusters into semantic relations
Extracting Tuples

• NYT corpus
• Apply lemmatization/stemming; NER tagging; dependency parsing
• For each pair of entity-mentions in a sentence
  – Extract dependency path between them
  – Form a tuple
• Filter rare tuples; tuples with 2 direct objects, etc
• Result: 1 M tuples, 500K entity mentions, 1300 patterns
Feature Representation

• Entity names, as bags of words, prefixed with :l or :r
  – ex: ("LA Lakers", "NY Knicks") => {l:LA, l:Lakers, r:NY, r:Knicks}
  – Using bag-of-words supports overlap and combats sparsity

• Words between and around the two entity mentions
  – Exclude stop words, words with capital letters
  – Include 2 words to the left and right

• Document theme/topic: (e.g., sports, politics, finance)
  – Assigned by LDA topic model that treats NYT topic descriptors as words in a synthetic document

• Sentence theme
  – Assigned by standard LDA topic model
Clustering Tuples into Senses

• Goal: to group tuples for each path into coherent sense clusters
• Apply yet another LDA topic model
  – Use Gibbs sampling for inference
• Results: each tuple is assigned one topic/sense
• Tuples with same topic/sense constitute a cluster
### Sense Cluster Examples

<table>
<thead>
<tr>
<th>Path</th>
<th>20: sports</th>
<th>30: entertainment</th>
<th>25: music/art</th>
</tr>
</thead>
</table>

| doc theme sen theme lexical words entity names | sports game yankees beat victory num-num won | music books television theater production book film show played plays directed artistic r: theater | music theater music reviews opera director conducted production r: theater r: hall r: york l: opera |

For path: *X plays Y*: with sample entity pairs, and top features
Now Cluster the Clusters!

• Cluster these sense clusters from different paths into semantic relation (very like Lin & Pantel 2003)
• Use Hierarchical Agglomerative Clustering
• Starts with minimal clusters, merges progressively
• Use cosine distance/similarity between sense-cluster feature vectors
Results

Like DIRT: each semantic relation has multiple paths; but: one path can now occur in multiple semantic relations
Evaluation Against Freebase

<table>
<thead>
<tr>
<th>System</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F-0.5</th>
<th>MCC</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F-0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rel-LDA/300</td>
<td>0.593</td>
<td>0.077</td>
<td>0.254</td>
<td>0.191</td>
<td>0.558</td>
<td>0.183</td>
<td>0.396</td>
</tr>
<tr>
<td>Rel-LDA/1000</td>
<td>0.638</td>
<td>0.061</td>
<td>0.220</td>
<td>0.177</td>
<td>0.626</td>
<td>0.160</td>
<td>0.396</td>
</tr>
<tr>
<td>HAC</td>
<td>0.567</td>
<td>0.152</td>
<td>0.367</td>
<td>0.261</td>
<td>0.523</td>
<td>0.248</td>
<td>0.428</td>
</tr>
<tr>
<td>Local</td>
<td>0.625</td>
<td>0.136</td>
<td>0.364</td>
<td>0.264</td>
<td>0.626</td>
<td>0.225</td>
<td>0.462</td>
</tr>
<tr>
<td>Local+Type</td>
<td>0.718</td>
<td>0.115</td>
<td>0.350</td>
<td>0.265</td>
<td>0.704</td>
<td>0.201</td>
<td>0.469</td>
</tr>
<tr>
<td><strong>Our Approach</strong></td>
<td><strong>0.736</strong></td>
<td><strong>0.156</strong></td>
<td><strong>0.422</strong></td>
<td><strong>0.314</strong></td>
<td>0.677</td>
<td>0.233</td>
<td><strong>0.490</strong></td>
</tr>
<tr>
<td><strong>Our Approach+Type</strong></td>
<td>0.682</td>
<td>0.110</td>
<td>0.334</td>
<td>0.250</td>
<td>0.687</td>
<td>0.199</td>
<td>0.460</td>
</tr>
</tbody>
</table>

HAC + Hierarchical Agglomerative Clustering alone – no sense disambiguation (similar to DIRT)

Sense clustering adds 17% to precision!