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 probabilty 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: <NP_i $relation_{ij}$ 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 *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
  - Lin & Pantel (2003)
- 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 he 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 mnetions, 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

Path	20:sports	30:entertainment	25:music/art
A play B	Americans, Ireland Yankees, Angels Ecuador, England Redskins, Detroit Red Bulls, F.C. Barcelona	Jean-Pierre Bacri, Jacques Rita Benton, Gay Head Dance Jeanie, Scrabble Meryl Streep, Leilah Kevin Kline, Douglas Fairbanks	Daniel Barenboim, recital of Mozart Mr. Rose, Ballade Gil Shaham, Violin Romance Ms. Golabek, Steinways Bruce Springsteen, Saints
doc theme	sports	music books television	music theater
sen theme	game yankees	theater production book film show	music reviews opera
lexical words	beat victory num-num won	played plays directed artistic	director conducted production
entity names	-	r:theater	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

relation	paths
entertainment	A, who play B:30; A play B 30; star A as B:30
sports	lead A to victory over B:20; A play to B:20; A play B 20; A's loss to B:20; A beat B:20; A trail B:20; A face B:26; A hold B:26; A play B:26; A acquire (X) from B:26; A send (X) to B:26;
politics	A nominate B:39; A name B:39; A select B:39; A name B:42; A select B:42; A ask B:42; A choose B:42; A nominate B:42; A turn to B:42;
law	A charge B:39; A file against B:39; A accuse B:39; A sue B:39

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

Evaluation Against Freebase

System	Pairwise				B^3		
	Prec.	Rec.	F-0.5	MCC	Prec.	Rec.	F-0.5
Rel-LDA/300	0.593	0.077	0.254	0.191	0.558	0.183	0.396
Rel-LDA/1000	0.638	0.061	0.220	0.177	0.626	0.160	0.396
HAC	0.567	0.152	0.367	0.261	0.523	**0.248**	0.428
Local	0.625	0.136	0.364	0.264	0.626	0.225	0.462
Local+Type	0.718	0.115	0.350	0.265	**0.704**	0.201	0.469
Our Approach	**0.736**	**0.156**	**0.422**	**0.314**	0.677	0.233	**0.490**
Our Approach+Type	0.682	0.110	0.334	0.250	0.687	0.199	0.460

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

Sense clustering adds 17% to precision!