Relation Extraction II

Luke Zettlemoyer
CSE 517
Winter 2013

[with slides adapted from many people, including Bill MacCartney, Raphael Hoffmann, Dan Jurafsky, Rion Snow, Jim Martin, Chris Manning, William Cohen, and others]
Supervised RE: summary

- Supervised approach can achieve high accuracy
  - At least, for *some* relations
  - If we have lots of hand-labeled training data
- But has significant limitations!
  - Labeling 5,000 relations (+ named entities) is expensive
  - Doesn’t generalize to different relations
- Next: beyond supervised relation extraction
  - Distantly supervised relation extraction
  - Unsupervised relation extraction
Relation extraction: 5 easy methods

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

Snow, Jurafsky, Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. NIPS 17


• **Hypothesis:** If two entities belong to a certain relation, any sentence containing those two entities is likely to express that relation
• **Key idea:** use a *database* of relations to get lots of *noisy* training examples
  o instead of hand-creating seed tuples (bootstrapping)
  o instead of using hand-labeled corpus (supervised)
Benefits of distant supervision

• Has advantages of supervised approach
  o leverage rich, reliable hand-created knowledge
  o relations have canonical names
  o can use rich features (e.g. syntactic features)

• Has advantages of unsupervised approach
  o leverage unlimited amounts of text data
  o allows for very large number of weak features
  o not sensitive to training corpus: genre-independent
Hypernyms via distant supervision

We construct a noisy training set consisting of occurrences from our corpus that contain a hyponym-hypernym pair from WordNet.

This yields high-signal examples like:

“...consider authors like Shakespeare...”
“Some authors (including Shakespeare)...”
“Shakespeare was the author of several...”
“Shakespeare, author of The Tempest...”
Hypernyms via distant supervision

We construct a noisy training set consisting of occurrences from our corpus that contain a hyponym-hypernym pair from WordNet.

This yields high-signal examples like:
“...consider authors like Shakespeare...”
“Some authors (including Shakespeare)...”
“Shakespeare was the author of several...”
“Shakespeare, author of The Tempest...”

But also noisy examples like:
“The author of Shakespeare in Love...”
“...authors at the Shakespeare Festival...”

slide adapted from Rion Snow
Learning hypernym patterns

Key idea: work at corpus level (entity pairs), instead of sentence level!

1. Take corpus sentences

   ... doubly heavy hydrogen atom called deuterium ...

2. Collect noun pairs

   e.g. (atom, deuterium)
   752,311 pairs from 6M sentences of newswire

3. Is pair an IS-A in WordNet?

   14,387 yes; 737,924 no

4. Parse the sentences

5. Extract patterns

6. Train classifier on patterns

   logistic regression with 70K features
   (converted to 974,288 bucketed binary features)

slide adapted from Rion Snow
One of 70,000 patterns

Pattern: <superordinate> called <subordinate>

Learned from cases such as:
(sarcoma, cancer)  ...an uncommon bone cancer called osteogenic sarcoma and to...
(deuterium, atom)  ...heavy water rich in the doubly heavy hydrogen atom called deuterium.

New pairs discovered:
(efflorescence, condition)  ...and a condition called efflorescence are other reasons for...
(O’neal_inc, company)  ...The company, now called O’Neal Inc., was sole distributor of...
(hat_creek_outfit, ranch)  ...run a small ranch called the Hat Creek Outfit.
(hiv-1, aids_virus)  ...infected by the AIDS virus, called HIV-1.
(bateau_mouche, attraction)  ...local sightseeing attraction called the Bateau Mouche...
Syntactic dependency paths

Patterns are based on paths through dependency parses generated by MINIPAR (Lin, 1998)

Example word pair: (Shakespeare, author)
Example sentence: “Shakespeare was the author of several plays...”

Minipar parse:

Extract shortest path:
-N:s:VBE, be, VBE:pred:N
Hearst patterns to dependency paths

**Hearst Pattern**

1. Y such as X ...
2. Such Y as X ...
3. X ... and other Y

**MINIPAR Representation**

1. \(-N:pcomp-n:Prep,such\_as,such\_as,-Prep:mod:N\)
2. \(-N:pcomp-n:Prep,as,as,-Prep:mod:N,(such,PreDet:pre:N)}\)

slide adapted from Rion Snow
P/R of hypernym extraction patterns

![Graph showing Precision vs Recall](image-url)

slide adapted from Rion Snow
P/R of hypernym extraction patterns

Individual Feature Analysis

- Y including X
- Y such as X
P/R of hypernym extraction patterns

Individual Feature Analysis

- □ Y including X
- + Y such as X
- × X and/or other Y

Precision

Recall (log)
P/R of hypernym extraction patterns
P/R of hypernym classifier

\[
P(R|E) = \frac{1}{1 + e^{-\sum w_i x_i}}
\]

10-fold Cross Validation on 14,000 WordNet-Labeled Pairs
P/R of hypernym classifier

\[
P(R|E) = \frac{1}{1 + e^{-\sum w_i x_i}}
\]

10-fold Cross Validation on 14,000 WordNet-Labeled Pairs

Hypernym Classifiers on WordNet-labeled dev set

- Best Logistic Regression (Buckets): 0.3480
- Best Logistic Regression (Binary): 0.3200
- Best Multinomial Naive Bayes: 0.3175
- Best Complement Naive Bayes: 0.3024
- Hearst Patterns: 0.1500
- “And/Or Other” Pattern: 0.1170
What about other relations?

Mintz, Bills, Snow, Jurafsky (2009).
Distant supervision for relation extraction without labeled data.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase</td>
<td>Wikipedia</td>
</tr>
<tr>
<td>102 relations</td>
<td>1.8 million articles</td>
</tr>
<tr>
<td>940,000 entities</td>
<td>25.7 million sentences</td>
</tr>
<tr>
<td>1.8 million instances</td>
<td></td>
</tr>
</tbody>
</table>

slide adapted from Rion Snow
<table>
<thead>
<tr>
<th>Relation name</th>
<th>Size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>/people/person/nationality</td>
<td>281,107</td>
<td>John Dugard, South Africa</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>253,223</td>
<td>Belgium, Nijlen</td>
</tr>
<tr>
<td>/people/person/profession</td>
<td>208,888</td>
<td>Dusa McDuff, Mathematician</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>105,799</td>
<td>Edwin Hubble, Marshfield</td>
</tr>
<tr>
<td>/dining/restaurant/cuisine</td>
<td>86,213</td>
<td>MacAyo’s Mexican Kitchen, Mexican</td>
</tr>
<tr>
<td>/business/business_chain/location</td>
<td>66,529</td>
<td>Apple Inc., Apple Inc., South Park, NC</td>
</tr>
<tr>
<td>/biology/organism_classification_rank</td>
<td>42,806</td>
<td>Scorpaeniformes, Order</td>
</tr>
<tr>
<td>/film/film/rank</td>
<td>40,658</td>
<td>Where the Sidewalk Ends, Film noir</td>
</tr>
<tr>
<td>/film/film/language</td>
<td>31,103</td>
<td>Enter the Phoenix, Cantonese</td>
</tr>
<tr>
<td>/biology/organism_higher_classification</td>
<td>30,052</td>
<td>Calopteryx, Calopterygida</td>
</tr>
<tr>
<td>/film/film/country</td>
<td>27,217</td>
<td>Turtle Diary, United States</td>
</tr>
<tr>
<td>/film/writer/film</td>
<td>23,856</td>
<td>Irving Shulman, Rebel Without a Cause</td>
</tr>
<tr>
<td>/film/director/film</td>
<td>23,539</td>
<td>Michael Mann, Collateral</td>
</tr>
<tr>
<td>/film/producer/film</td>
<td>22,079</td>
<td>Diane Eskenazi, Aladdin</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>18,814</td>
<td>John W. Kern, Asheville</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>18,619</td>
<td>The Octopus Project, Austin</td>
</tr>
<tr>
<td>/people/person/religion</td>
<td>17,582</td>
<td>Joseph Chartrand, Catholicism</td>
</tr>
<tr>
<td>/book/author/works_written</td>
<td>17,278</td>
<td>Paul Auster, Travels in the Scriptorium</td>
</tr>
<tr>
<td>/soccer/football_position/players</td>
<td>17,244</td>
<td>Midfielder, Chen Tao</td>
</tr>
<tr>
<td>/people/deceased_person/cause_of_death</td>
<td>16,709</td>
<td>Richard Daintree, Tuberculosis</td>
</tr>
<tr>
<td>/film/film/music</td>
<td>14,070</td>
<td>Stavisky, Stephen Sondheim</td>
</tr>
<tr>
<td>/business/company/industry</td>
<td>13,805</td>
<td>ATS Medical, Health care</td>
</tr>
</tbody>
</table>
Collecting training data

Corpus text
- Bill Gates founded Microsoft in 1975.
- Bill Gates, founder of Microsoft, ...
- Bill Gates attended Harvard from ...
- Google was founded by Larry Page ...

Training data

Freebase
- Founder: (Bill Gates, Microsoft)
- Founder: (Larry Page, Google)
- CollegeAttended: (Bill Gates, Harvard)
Collecting training data

Corpus text

*Bill Gates* founded *Microsoft* in 1975.
*Bill Gates*, founder of *Microsoft*, …
*Bill Gates* attended Harvard from…
*Google* was founded by *Larry Page* …

Training data

(Bill Gates, Microsoft)
Label: Founder
Feature: X founded Y

Freebase

Founder: *(Bill Gates, Microsoft)*
Founder: (Larry Page, Google)
CollegeAttended: (Bill Gates, Harvard)
Bill Gates founded Microsoft in 1975. 

Bill Gates, founder of Microsoft, …

Bill Gates attended Harvard from …

Google was founded by Larry Page …

(Bill Gates, Microsoft)
Label: Founder
Feature: X founded Y
Feature: X, founder of Y

 Founder: (Bill Gates, Microsoft)
Founder: (Larry Page, Google)
CollegeAttended: (Bill Gates, Harvard)
Collecting training data

Corpus text

Bill Gates founded Microsoft in 1975.
Bill Gates, founder of Microsoft, …
**Bill Gates attended Harvard from**…
Google was founded by Larry Page …

Training data

(Bill Gates, Microsoft)
Label: Founder
Feature: X founded Y
Feature: X, founder of Y

(Bill Gates, Harvard)
Label: CollegeAttended
Feature: X attended Y

Freebase

Founder: (Bill Gates, Microsoft)
Founder: (Larry Page, Google)
CollegeAttended: ([Bill Gates], [Harvard])
Collecting training data

Corpus text

Bill Gates founded Microsoft in 1975.
Bill Gates, founder of Microsoft, …
Bill Gates attended Harvard from…
Google was founded by Larry Page …

Freebase

Founder: (Bill Gates, Microsoft)
Founder: (Larry Page, Google)
CollegeAttended: (Bill Gates, Harvard)

Training data

(Bill Gates, Microsoft)
Label: Founder
Feature: X founded Y
Feature: X, founder of Y

(Bill Gates, Harvard)
Label: CollegeAttended
Feature: X attended Y

(Larry Page, Google)
Label: Founder
Feature: Y was founded by X
Negative training data

Can’t train a classifier with only positive data! Need negative training data too!

Solution?
Sample 1% of unrelated pairs of entities.

Larry Page took a swipe at Microsoft...
...after Harvard invited Larry Page to...
Google is Bill Gates' worst fear ...

Training data
- (Larry Page, Microsoft)
  - Label: NO_RELATION
  - Feature: X took a swipe at Y

- (Larry Page, Harvard)
  - Label: NO_RELATION
  - Feature: Y invited X

- (Bill Gates, Google)
  - Label: NO_RELATION
  - Feature: Y is X's worst fear
Preparing test data

Corpus text

- Henry Ford founded Ford Motor Co. in...
- Ford Motor Co. was founded by Henry Ford...
- Steve Jobs attended Reed College from...

Test data
Preparing test data

Corpus text

Henry Ford founded Ford Motor Co. in...
Ford Motor Co. was founded by Henry Ford...
Steve Jobs attended Reed College from...

Test data

(Henry Ford, Ford Motor Co.)
Label: ???
Feature: X founded Y
Preparing test data

Corpus text

Henry Ford founded Ford Motor Co. in…
Ford Motor Co. was founded by Henry Ford…
Steve Jobs attended Reed College from…

Test data

(Henry Ford, Ford Motor Co.)
Label: ???
Feature: X founded Y
Feature: Y was founded by X
Preparing test data

Corpus text

Henry Ford founded Ford Motor Co. in…
Ford Motor Co. was founded by Henry Ford…
Steve Jobs attended Reed College from…

Test data

(Henry Ford, Ford Motor Co.)
Label: ???
Feature: X founded Y
Feature: Y was founded by X

(Steve Jobs, Reed College)
Label: ???
Feature: X attended Y
The experiment

Learning: multiclass logistic regression

Positive training data
(Bill Gates, Microsoft)
Label: Founder
Feature: X
Feature: X, founder of Y

(Bill Gates, Harvard)
Label: CollegeAttended
Feature: X

(Larry Page, Google)
Label: Founder
Feature: Y was founded by X

Negative training data
(Larry Page, Microsoft)
Label: NO_RELATION
Feature: X took a swipe at Y

(Larry Page, Harvard)
Label: NO_RELATION
Feature: X

(Bill Gates, Google)
Label: NO_RELATION
Feature: Y is X's worst fear

Test data
(Henry Ford, Ford Motor Co.)
Label: ???
Feature: X

(Bill Gates, Microsoft)
Label: Founder
Feature: X

(Henry Ford, Ford Motor Co.)
Label: Founder
Feature: X was founded by X

(Steve Jobs, Reed College)
Label: CollegeAttended

Trained relation classifier

Predictions!
Advantages of the approach

• ACE paradigm: labeling sentences
• This paradigm: labeling entity pairs
• We make use of multiple appearances of entities
• If a pair of entities appears in 10 sentences, and each sentence has 5 features extracted from it, the entity pair will have 50 associated features
Lexical and syntactic features

Astronomer Edwin Hubble was born in Marshfield, Missouri.
## High-weight features

<table>
<thead>
<tr>
<th>Relation</th>
<th>Feature type</th>
<th>Left window</th>
<th>NE1</th>
<th>Middle</th>
<th>NE2</th>
<th>Right window</th>
</tr>
</thead>
<tbody>
<tr>
<td>/architecture/structure/architect</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>the designer of the</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/book/author/writes/author</td>
<td>SYN</td>
<td>designed</td>
<td>ORG</td>
<td>by _mod story _pred is _s</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>co-founder</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/business/company/place_founded</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/company/company/place_founded</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/film/film/country</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/geo/geo/river/mouth</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/government/political_party/country</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/government/political_party/country</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/influence/influence_node/influenced</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/language/human_language/region</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/music/music/artist/band</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/people/deceased/person/place_of_death</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/people/person/parents</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
<tr>
<td>/people/person/religion</td>
<td>LEX</td>
<td></td>
<td>ORG</td>
<td>_nn owner _person</td>
<td>PER</td>
<td></td>
</tr>
</tbody>
</table>
Implementation

• Classifier: multi-class logistic regression optimized using L-BFGS with Gaussian regularization (Manning & Klein 2003)

• Parser: MINIPAR (Lin 1998)

• POS tagger: MaxEnt tagger trained on the Penn Treebank (Toutanova et al. 2003)

• NER tagger: Stanford four-class tagger \{PER, LOC, ORG, MISC, NONE\} (Finkel et al. 2005)

• 3 configurations: lexical features, syntax features, both
Experimental set-up

• 1.8 million relation instances used for training
  ○ Compared to 17,000 relation instances in ACE

• 800,000 Wikipedia articles used for training, 400,000 different articles used for testing

• Only extract relation instances not already in Freebase
Newly discovered instances

Ten relation instances extracted by the system that weren’t in Freebase

<table>
<thead>
<tr>
<th>Relation name</th>
<th>New instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>/location/location/contains</td>
<td>Paris, Montmartre</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>Ontario, Fort Erie</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>Mighty Wagon, Cincinnati</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>Fyodor Kamensky, Clearwater</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>Marianne Yvonne Heemskerk, Netherlands</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>Wavell Wayne Hinds, Kingston</td>
</tr>
<tr>
<td>/book/author/works_written</td>
<td>Upton Sinclair, Lanny Budd</td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>WWE, Vince McMahon</td>
</tr>
<tr>
<td>/people/person/profession</td>
<td>Thomas Mellon, judge</td>
</tr>
</tbody>
</table>
Human evaluation

Precision, using Mechanical Turk labelers:

<table>
<thead>
<tr>
<th>Relation name</th>
<th>100 instances</th>
<th></th>
<th>1000 instances</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Syn</td>
<td>Lex</td>
<td>Both</td>
<td>Syn</td>
</tr>
<tr>
<td>/film/director/film</td>
<td>0.49</td>
<td>0.43</td>
<td>0.44</td>
<td>0.49</td>
</tr>
<tr>
<td>/film/writer/film</td>
<td>0.70</td>
<td>0.60</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>/geography/river/basin_countries</td>
<td>0.65</td>
<td>0.64</td>
<td>0.67</td>
<td>0.73</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>0.68</td>
<td>0.59</td>
<td>0.70</td>
<td>0.72</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>0.81</td>
<td>0.89</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td>/location/us_county/county_seat</td>
<td>0.51</td>
<td>0.51</td>
<td>0.53</td>
<td>0.47</td>
</tr>
<tr>
<td>/music/artist/origin</td>
<td>0.64</td>
<td>0.66</td>
<td>0.71</td>
<td>0.61</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>0.80</td>
<td>0.79</td>
<td>0.81</td>
<td>0.80</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>0.61</td>
<td>0.70</td>
<td>0.72</td>
<td>0.56</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>0.78</td>
<td>0.77</td>
<td>0.78</td>
<td>0.88</td>
</tr>
<tr>
<td>Average</td>
<td>0.67</td>
<td>0.66</td>
<td>0.69</td>
<td>0.68</td>
</tr>
</tbody>
</table>

- At recall of 100 instances, using both feature sets (lexical and syntax) offers the best performance for a majority of the relations.
- At recall of 1000 instances, using syntax features improves performance for a majority of the relations.
Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, Daniel S. Weld

University of Washington

06/20/11
Previous Work: Aggregate Extraction

1. Steve Jobs presents 2. Apple’s HQ.
2. Apple CEO 1. Steve Jobs ...
2. Steve Jobs, CEO of Apple, ...
1. Google’s takeover of Youtube ...
2. Youtube, now part of Google, ...
2. Apple and 1. IBM are public.
   1. Microsoft’s purchase of 2. Skype.

e.g. [Mintz et al. 2010]
This Talk: Sentence-level Reasoning

1. Steve Jobs presents Apple’s HQ.
2. Apple CEO Steve Jobs ...
1. Steve Jobs holds Apple stock.
2. Steve Jobs, CEO of Apple, ...
1. Google’s takeover of Youtube ...
2. Youtube, now part of Google, ...
2. Apple and IBM are public.
… 1. Microsoft’s purchase of Skype.

Train so that extracted facts match facts in DB

CEO-of(Rob Iger, Disney)
CEO-of(Steve Jobs, Apple)
Acquired(Google, Youtube)
Acquired(Msft, Skype)
Acquired(Citigroup, EMI)
Model

Steve Jobs, Apple:

\[ p(Y = y, Z = z | x; \theta) \overset{\text{def}}{=} \frac{1}{Z_x} \prod_r \Phi^{\text{join}}(y^r, z) \prod_i \Phi^{\text{extract}}(z_i, x_i) \]

\[ \Phi^{\text{join}}(y^r, z) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } y^r = \text{true} \land \exists i : z_i = r \\ 0 & \text{otherwise} \end{cases} \]

All features at sentence-level
(join factors are deterministic ORs)
Inference

Need:

• Most likely sentence labels:

\[
\arg \max_{y,z} p(y, z|x; \theta)
\]

Easy

• Most likely sentence labels *given* facts:

\[
\arg \max_z p(z|x, y; \theta)
\]

Challenging
Learning: Hidden-Variable Perceptron

\[
\text{initialize} \ \text{parameter vector } \Theta \leftarrow 0
\]
\[
\text{for } t = 1 \ldots T \ \text{do}
\]
\[
\text{for } i = 1 \ldots n \ \text{do}
\]
\[
(y', z') \leftarrow \arg \max_{y,z} p(y, z|x_i; \theta)
\]
\[
\text{if } y' \neq y_i \ \text{then}
\]
\[
z^* \leftarrow \arg \max_z p(z|x_i, y_i; \theta)
\]
\[
\Theta \leftarrow \Theta + \phi(x_i, z^*) - \phi(x_i, z')
\]
\[
\text{end if}
\]
\[
\text{end for}
\]
\[
\text{end for}
\]

Return \( \Theta \)
Experimental Setup

• Data as in Riedel et al. 10:
  – LDC NYT corpus, 2005-06 (training), 2007 (testing)
  – Data first tagged with Stanford NER system
  – Entities matched to Freebase, ~ top 50 relations
  – Mention-level features as in Mintz et al. 09

• Systems:
  – MultiR: proposed approach
  – SoloR: re-implementation of Riedel et al. 2010
Aggregate Extraction

How does set of predicted facts match to facts in Freebase?

Metric

• For each entity pair compare inferred facts to facts in Freebase
• Automated, but underestimates precision
Dip: manual check finds that 23 out of the top 25 extractions were true facts, missing from Freebase
How accurate is extraction from a given sentence?

Metric

• Sample 1000 sentences from test set
• Manual evaluation of precision and recall
7.2 Sentential Extraction

Although their model includes variables to model sentential extraction, Riedel et al. did not report sentence level performance. To generate the precision vs recall curve we used the joint model as assignment score for each of the sentences that contributed to the aggregate extraction decision. Figure 1 shows approximate precision vs recall curves for MULTI and SOLOR computed against manually generated sentence labels, as defined in Section 5. MULTI achieves significantly higher recall with a consistently high level of precision. At the highest recall point, MULTI reaches (.6, x) precision and (.06%) recall, for an F0 score of (.96/).
Relation-specific Performance

What is the quality of the matches for different relations?

How does our approach perform for different relations?

Metric:

• Select 10 relations with highest #matches
• Sample 100 sentences for each relation
• Manually evaluate precision and recall
## Quality of the Matching

<table>
<thead>
<tr>
<th>Relation</th>
<th>Freebase Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>/business/person/company</td>
<td>302</td>
</tr>
<tr>
<td>/people/person/place_lived</td>
<td>450</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>2793</td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>95</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>723</td>
</tr>
<tr>
<td>/location/neighborhood/neighborhood_of</td>
<td>68</td>
</tr>
<tr>
<td>/people/person/children</td>
<td>30</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>68</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>162</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>424</td>
</tr>
</tbody>
</table>
Quality of the Matching

<table>
<thead>
<tr>
<th>Relation</th>
<th>Freebase Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sents</td>
</tr>
<tr>
<td>/business/person/company</td>
<td>302</td>
</tr>
<tr>
<td>/people/person/place_lived</td>
<td>450</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>2793</td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>95</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>723</td>
</tr>
<tr>
<td>/location/neighborhood/neighborhood_of</td>
<td>68</td>
</tr>
<tr>
<td>/people/person/children</td>
<td>30</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>68</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>162</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>424</td>
</tr>
</tbody>
</table>
Performance of MultiR

<table>
<thead>
<tr>
<th>Relation</th>
<th>Freebase Matches</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sents</td>
<td>% true</td>
<td>precision</td>
<td>recall</td>
</tr>
<tr>
<td>/business/person/company</td>
<td>302</td>
<td>89.0</td>
<td>100.0</td>
<td>25.8</td>
</tr>
<tr>
<td>/people/person/place_lived</td>
<td>450</td>
<td>60.0</td>
<td>80.0</td>
<td>6.7</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>2793</td>
<td>51.0</td>
<td>100.0</td>
<td>56.0</td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>95</td>
<td>48.4</td>
<td>71.4</td>
<td>10.9</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>723</td>
<td>41.0</td>
<td>85.7</td>
<td>15.0</td>
</tr>
<tr>
<td>/location/neighborhood/neighborhood_of</td>
<td>68</td>
<td>39.7</td>
<td>100.0</td>
<td>11.1</td>
</tr>
<tr>
<td>/people/person/children</td>
<td>30</td>
<td>80.0</td>
<td>100.0</td>
<td>8.3</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>68</td>
<td>22.1</td>
<td>100.0</td>
<td>20.0</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>162</td>
<td>12.0</td>
<td>100.0</td>
<td>33.0</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>424</td>
<td>0.2</td>
<td>N/A</td>
<td>0.0</td>
</tr>
</tbody>
</table>
## Overlapping Relations

<table>
<thead>
<tr>
<th>Relation</th>
<th>Freebase Matches</th>
<th>MultiR</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sents</td>
<td>% true</td>
<td>precision</td>
<td>recall</td>
</tr>
<tr>
<td>/business/person/company</td>
<td>302</td>
<td>89.0</td>
<td>100.0</td>
<td>25.8</td>
</tr>
<tr>
<td>/people/person/place_lived</td>
<td>450</td>
<td>60.0</td>
<td>80.0</td>
<td>6.7</td>
</tr>
<tr>
<td>/location/location/contains</td>
<td>2793</td>
<td>51.0</td>
<td>100.0</td>
<td>56.0</td>
</tr>
<tr>
<td>/business/company/founders</td>
<td>95</td>
<td>48.4</td>
<td>71.4</td>
<td>10.9</td>
</tr>
<tr>
<td>/people/person/nationality</td>
<td>723</td>
<td>41.0</td>
<td>85.7</td>
<td>15.0</td>
</tr>
<tr>
<td>/location/neighborhood/neighborhood_of</td>
<td>68</td>
<td>39.7</td>
<td>100.0</td>
<td>11.1</td>
</tr>
<tr>
<td>/people/person/children</td>
<td>30</td>
<td>80.0</td>
<td>100.0</td>
<td>8.3</td>
</tr>
<tr>
<td>/people/deceased_person/place_of_death</td>
<td>68</td>
<td>22.1</td>
<td>100.0</td>
<td>20.0</td>
</tr>
<tr>
<td>/people/person/place_of_birth</td>
<td>162</td>
<td>12.0</td>
<td>100.0</td>
<td>33.0</td>
</tr>
<tr>
<td>/location/country/administrative_divisions</td>
<td>424</td>
<td>0.2</td>
<td>N/A</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Running Time

- **MultiR**
  - Training: 1 minute
  - Testing: 1 second

- **SoloR**
  - Training: 6 hours
  - Testing: 4 hours

Sentence-level extractions are efficient

Joint reasoning across sentences is computationally expensive
Distant supervision: conclusions

- Distant supervision extracts high-precision patterns for a variety of relations
- Can make use of 1000x more data than simple supervised algorithms
- Syntax features almost always help
- The combination of syntax and lexical features is sometimes even better
- Syntax features are probably most useful when entities are far apart, often when there are modifiers in between
Relation extraction: 5 easy methods

1. Hand-built patterns
2. Bootstrapping methods
3. Supervised methods
4. Distant supervision
5. Unsupervised methods
DIRT (Lin & Pantel 2003)

• DIRT = Discovery of Inference Rules from Text

• Looks at MINIPAR dependency paths between noun pairs
  - N:subj:V←find→V:obj:N→solution→N:to:N
  - i.e., X finds solution to Y

• Applies "extended distributional hypothesis"
  - If two paths tend to occur in similar contexts, the meanings of the paths tend to be similar.

• So, defines path similarity in terms of cooccurrence counts with various slot fillers

• Thus, extends ideas of (Lin 1998) from words to paths
DIRT examples

The top-20 most similar paths 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

Y is resolved in X
Y is solved through X
X rectifies Y
X copes with Y
X overcomes Y
X eases Y
X tackles Y
X alleviates Y
X corrects Y
X is a solution to Y
Ambiguous paths in DIRT

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

• X tackles Y
  o Foley tackled the quarterback in the endzone.
  o Police are beginning to tackle rising crime.

• X is a solution to Y
  o (5, 1) is a solution to the equation 2x – 3y = 7
  o Nuclear energy is a solution to the energy crisis.
TextRunner (Banko et al. 2007)

1. **Self-supervised learner**: automatically labels +/- examples & learns a crude relation extractor

2. **Single-pass extractor**: makes one pass over corpus, extracting candidate relations in each sentence

3. **Redundancy-based assessor**: assigns a probability to each extraction, based on frequency counts
Step 1: Self-supervised learner

• Run a parser over 2000 sentences
  o Parsing is relatively expensive, so can’t run on whole web
  o For each pair of base noun phrases $NP_i$ and $NP_j$
  o Extract all tuples $t = (NP_i, \text{relation}_{i,j}, NP_j)$

• Label each tuple based on features of parse:
  o Positive iff the dependency path between the NPs is short, and doesn’t cross a clause boundary, and neither NP is a pronoun

• Now train a Naïve Bayes classifier on the labeled tuples
  o Using lightweight features like POS tags nearby, stop words, etc.
Step 2: Single-pass extractor

- Over a huge (web-sized) corpus:
  - Run a dumb POS tagger
  - Run a dumb Base Noun Phrase chunker
  - Extract all text strings between base NPs
  - Run heuristic rules to simplify text strings
    
    Scientists from many universities are intently studying stars
    
    $\rightarrow$ \{scientists, are studying, stars\}

- Pass candidate tuples to Naïve Bayes classifier

- Save only those predicted to be “trustworthy”
Step 3: Redundancy-based assessor

- Collect counts for each simplified tuple
  \[ \langle \text{scientists, are studying, stars} \rangle \rightarrow 17 \]

- Compute likelihood of each tuple
  - given the counts for each relation
  - and the number of sentences
  - and a combinatoric balls-and-urns model [Downey et al. 05]

\[
P(x \in C | x \text{ appears } k \text{ times in } n \text{ draws}) \approx \frac{1}{1 + \frac{|E|}{|C|} \left( \frac{p_E}{p_C} \right)^k e^{n(p_C - p_E)}}
\]
TextRunner demo

http://www.cs.washington.edu/research/textrunner/

(Note that they’ve re-branded TextRunner as ReVerb, but it’s largely the same system.)
## TextRunner examples

<table>
<thead>
<tr>
<th>Probability</th>
<th>Count</th>
<th>Arg1</th>
<th>Predicate</th>
<th>Arg2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.98</td>
<td>59</td>
<td>Smith</td>
<td>invented</td>
<td>the margherita</td>
</tr>
<tr>
<td>0.97</td>
<td>49</td>
<td>Al Gore</td>
<td>invented</td>
<td>the Internet</td>
</tr>
<tr>
<td>0.97</td>
<td>44</td>
<td>manufacturing plant</td>
<td>first invented</td>
<td>the automatic revolver</td>
</tr>
<tr>
<td>0.97</td>
<td>41</td>
<td>Alexander Graham Bell</td>
<td>invented</td>
<td>the telephone</td>
</tr>
<tr>
<td>0.97</td>
<td>36</td>
<td>Thomas Edison</td>
<td>invented</td>
<td>light bulbs</td>
</tr>
<tr>
<td>0.97</td>
<td>29</td>
<td>Eli Whitney</td>
<td>invented</td>
<td>the cotton gin</td>
</tr>
<tr>
<td>0.96</td>
<td>23</td>
<td>C. Smith</td>
<td>invented</td>
<td>the margherita</td>
</tr>
<tr>
<td>0.96</td>
<td>19</td>
<td>the Digital Equipment</td>
<td>first invented</td>
<td>the automatic revolver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corporation manufacturing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>plant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.96</td>
<td>18</td>
<td>Edison</td>
<td>invented</td>
<td>the phonograph</td>
</tr>
</tbody>
</table>
TextRunner results

• From corpus of 9M web pages, containing 133M sentences
• Extracted 60.5 million tuples
  ▪ 〈FCI, specializes in, software development〉
• Evaluation
  o Not well formed:
    ▪ 〈demands, of securing, border〉 〈29, dropped, instruments〉
  o Abstract:
    ▪ 〈Einstein, derived, theory〉 〈executive, hired by, company〉
  o True, concrete:
    ▪ 〈Tesla, invented, coil transformer〉
Evaluating TextRunner

- Tuples: 11.3 million
- With Well-Formed Relation: 9.3 million
- With Well-Formed Entities: 7.8 million
- Abstract: 6.8 million, 79.2% correct
- Concrete: 1 million, 88.1% correct
Yao et al. 2012: motivation

• Goal: induce clusters of dependency paths which express the same semantic relation, like DIRT

• But, improve upon DIRT by properly handling semantic ambiguity of individual paths
Yao et al. 2012: approach

1. Extract tuples (entity, path, entity) from corpus
2. Construct feature representations of every tuple
3. Group the tuples for each path into sense clusters
4. Cluster the sense clusters into semantic relations
Extracting tuples

• Start with NYT corpus

• Apply lemmatization, NER tagging, dependency parsing

• For each pair of entities in a sentence:
  o Extract dependency path between them, as in Lin
  o Form a tuple consisting of the two entities and the path

• Filter rare tuples, tuples with two direct objects, etc.

• Result: 1M tuples, 500K entities, 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 encourages overlap, i.e., combats sparsity

- Words between and around the two entities
  - Exclude stop words, words with capital letters
  - Include two words to the left and right

- Document theme (e.g. sports, politics, finance)
  - Assigned by an LDA topic model which treats NYTimes topic descriptors as words in a synthetic document

- Sentence theme
  - Assigned by a standard LDA topic model
Clustering tuples into senses

• Goal: group tuples for each path into coherent sense clusters
• Currently exploring multiple different approaches:
  • LDA-like topic models
  • Matrix factorization approaches
• Result: each tuple is assigned one topic/sense
• Tuples with the 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>
<tbody>
<tr>
<td>doc theme</td>
<td>sports</td>
<td>music books television theater production book film show played plays directed artistic</td>
<td>music theater music reviews opera director conducted production r:theater r:hall r:york l:opera</td>
</tr>
<tr>
<td>sen theme</td>
<td>game yankees</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>lexical words</td>
<td>beat victory num-num won</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>entity names</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Clustering the clusters!

- Now cluster sense clusters from different paths into semantic relations — this is the part most similar to Lin & Pantel 2003
- Use Hierarchical Agglomerative Clustering (HAC)
- Start with minimal clustering, then merge progressively
- Uses cosine similarity between sense-cluster feature vectors
- Uses complete-linkage strategy
Just like DIRT, each semantic relation has multiple paths. But, one path can now appear in multiple semantic relations. DIRT can’t do that!
Evaluation against Freebase

Automatic evaluation against Freebase
HAC = hierarchical agglomerative clustering alone
(i.e. no sense disambiguation — most similar to DIRT)
Sense clustering adds 17% to precision!

<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>Our Approach</td>
<td>0.736</td>
<td>0.156</td>
<td>0.422</td>
<td>0.314</td>
<td>0.677</td>
<td>0.233</td>
<td>0.490</td>
</tr>
<tr>
<td>Our Approach+Type</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>