Relation Extraction

Bill MacCartney

CS224U

18-20 April 2016

[with slides adapted from many people, including Dan Jurafsky,
Rion Snow, Jim Martin, Chris Manning, William Cohen,
Michele Banko, Mike Mintz, Steven Bills, and others]

Goal: "machine reading"

Reading the Web: A Breakthrough Goal for AI

I believe AI has an opportunity to achieve a true breakthrough over the coming decade by at last solving the problem of reading natural language text to extract its factual content. In fact, I hereby offer to bet anyone a lobster dinner that by 2015 we will have a computer program capable of automatically reading at least 80% of the factual content [on the] web, and placing those facts in a structured knowledge base. The significance of this AI achievement would be tremendous: it would immediately increase by many orders of magnitude the volume, breadth, and depth of ground facts and general knowledge accessible to knowledge based AI programs. In essence, computers would be harvesting in structured form the huge volume of knowledge that millions of humans are entering daily on the web in the form of unstructured text.

— Tom Mitchell, 2004

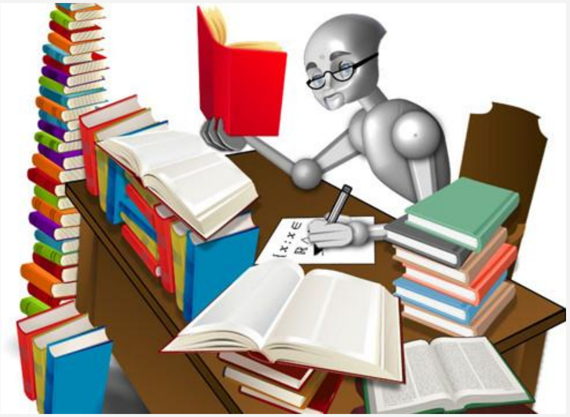


illustration from DARPA

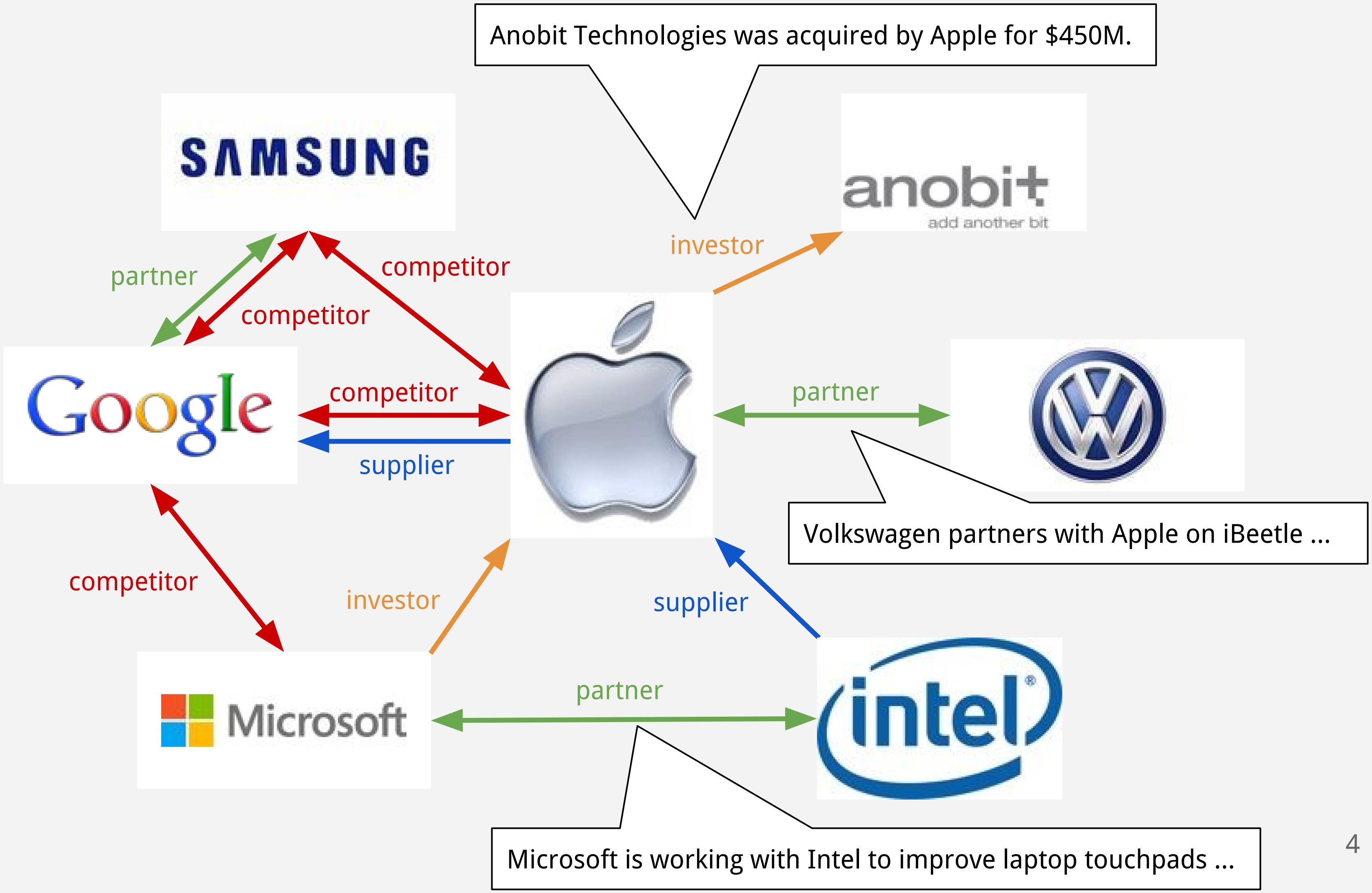
Relation extraction example

CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York.

Subject	Relation	Object
American Airlines	subsidiary	AMR
Tim Wagner	employee	American Airlines
United Airlines	subsidiary	UAL

example from Jim Martin

Example: company relationships



Example: gene regulation

Involvement of Tumor Necrosis Factor Receptor-associated Protein 1 (TRAP1) in Apoptosis Induced by β-Hydroxyisovalerylshikonin*

Subject	Relation	Object
p53	**is_a**	protein
Bax	**is_a**	protein
p53	has_function	apoptosis
Bax	has_function	induction
apoptosis	involved_in	cell_death
Bax	is_in	mitochondrial outer membrane
Bax	is_in	cytoplasm
apoptosis	related_to	caspase activation
...	...	...

textual abstract:
summary for human

structured knowledge extraction:
summary for machine

Lexical semantic relations

Many NLP applications require understanding relations between word senses: synonymy, antonymy, hyponymy, meronymy.

WordNet is a machine-readable database of relations between word senses, and an indispensable resource in many NLP tasks.

http://wordnetweb.princeton.edu/perl/webwn

```
vehicle
    craft
        aircraft
            airplane
            dirigible
            helicopter
        spacecraft
        watercraft
            boat
            ship
            yacht
    rocket
        missile
        multistage rocket
    wheeled vehicle
        automobile
        bicycle
        locomotive
        wagon
```

WordNet is incomplete

But WordNet is manually constructed, and has many gaps!

In WordNet 3.1	Not in WordNet 3.1
insulin progesterone	leptin pregnenolone
combustibility navigability	affordability reusability
HTML	XML
Google, Yahoo	Microsoft, IBM

Esp. for specific domains: restaurants, auto parts, finance

Esp. neologisms: iPad, selfie, bitcoin, twerking, Hadoop, dubstep

Example: extending WordNet

Mirror ran a headline questioning whether the killer's actions were a result of playing Call of Duty, a first-person shooter game ...

Melee, in video game terms, is a style of elbow-drop hand-to-hand combat popular in first-person shooters and other shooters.

Tower defense is a kind of real-time strategy game in which the goal is to protect an area/place/locality and prevent enemies from reaching ...

```
video game
  action game
    ball and paddle game
      Breakout
    platform game
      Donkey Kong
    shooter
      arcade shooter
        Space Invaders
      first-person shooter
        Call of Duty
      third-person shooter
        Tomb Raider
  adventure game
    text adventure
    graphic adventure
  strategy game
    4X game
      Civilization
    tower defense
      Plants vs. Zombies
```

Example: extending Freebase

Freebase: 20K relations, 40M entities, 600M assertions

Curation is an ongoing challenge — things change!

Relies heavily on relation extraction from the web

/film/film/starring

Bad Words	Jason Bateman
Divergent	Shailene Woodley
Non-Stop	Liam Neeson

/organization/organization/parent

WhatsApp	Facebook
Nest Labs	Google
Nokia	Microsoft

/music/artist/track

Macklemore	White Privilege
Phantogram	Mouthful of Diamonds
Lorde	Royals

/people/person/date_of_death

Nelson Mandela	2013-12-05
Paul Walker	2013-11-30
Lou Reed	2013-10-27

Relation extraction: 5 easy methods

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

Relation extraction: 5 easy methods

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

A hand-built extraction pattern

```
;;; For <company> appoints <person> <position>

(defpattern appoint
   "np-sem(C-company)?  rn?  sa?  vg(C-appoint) np-sem(C-person) ´,´?
    to-be?  np(C-position) to-succeed?:
    company-at=1.attributes, sa=3.span, lv=4.span, person-at=5.attributes
    position-at=8.attributes |
...
(defun when-appoint (phrase-type)
    (let ((person-at (binding ´person-at))
        (company-entity (entity-bound ´company-at))
        (person-entity (essential-entity-bound ´person-at ´C-person))
        (position-entity (entity-bound ´position-at))
        (predecessor-entity (entity-bound ´predecessor-at))
        new-event)
     (not-an-antecedent position-entity)
     ;; if no company is specified for position, use agent
...
```

NYU Proteus system (1997)

Patterns for learning hyponyms

- 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?

Patterns for learning hyponyms

- 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?

Hearst's lexico-syntactic patterns

Ys such as X ((, X)* (, and/or) X)
such Ys as X…
X… or other Ys
X… and other Ys
Ys including X…
Ys, especially X…

Hearst, 1992. Automatic Acquisition of Hyponyms.

Examples: “Ys, especially X”

The best part of the night was seeing all of the tweets of the performers, especially Miley Cyrus and Drake. ✓

Those child stars, especially Miley Cyrus, I feel like you have to put the fault on the media. ✓

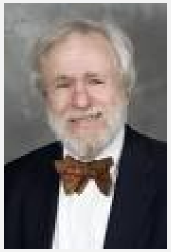
Kelly wasn’t shy about sharing her feelings about some of the musical acts, especially Miley Cyrus. ✓

Rihanna was bored with everything at the MTV VMAs, especially Miley Cyrus. ✗

The celebrities enjoyed themselves while sipping on delicious cocktails, especially Miley Cyrus who landed the coveted #1 spot. ✗

None of these girls are good idols or role models, especially Miley Cyrus. ✗

Patterns for learning meronyms

- Berland & Charniak (1999) tried it
- Selected initial patterns by finding all sentences in a corpus containing *basement* and *building*

whole NN[-PL] 's POS part NN[-PL]	... building's basement ...
part NN[-PL] of PREP {the|a} DET mods [JJ|NN]* whole NN	... basement of a building ...
part NN in PREP {the|a} DET mods [JJ|NN]* whole NN	... basement in a building ...
parts NN-PL of PREP wholes NN-PL	... basements of buildings ...
parts NN-PL in PREP wholes NN-PL	... basements in buildings ...

- Then, for each pattern:
    1. found occurrences of the pattern
    2. filtered those ending with *-ing*, *-ness*, *-ity*
    3. applied a likelihood metric — poorly explained
- Only the first two patterns gave decent (though not great!) results

Problems with hand-built patterns

- Requires hand-building patterns for each relation!
  - and every language!
  - hard to write; hard to maintain
  - there are zillions of them
  - domain-dependent
- Don't want to do this for all possible relations!
- Plus, we'd like better accuracy
  - Hearst: 66% accuracy on hyponym extraction
  - Berland & Charniak: 55% accuracy on meronyms

Relation extraction: 5 easy methods

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

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: *burial place*
- Seed tuple: [*Mark Twain, Elmira*]
- Grep/Google for “Mark Twain” and “Elmira”

  “Mark Twain is buried in Elmira, NY.”

  → X is buried in Y

  “The grave of Mark Twain is in Elmira”

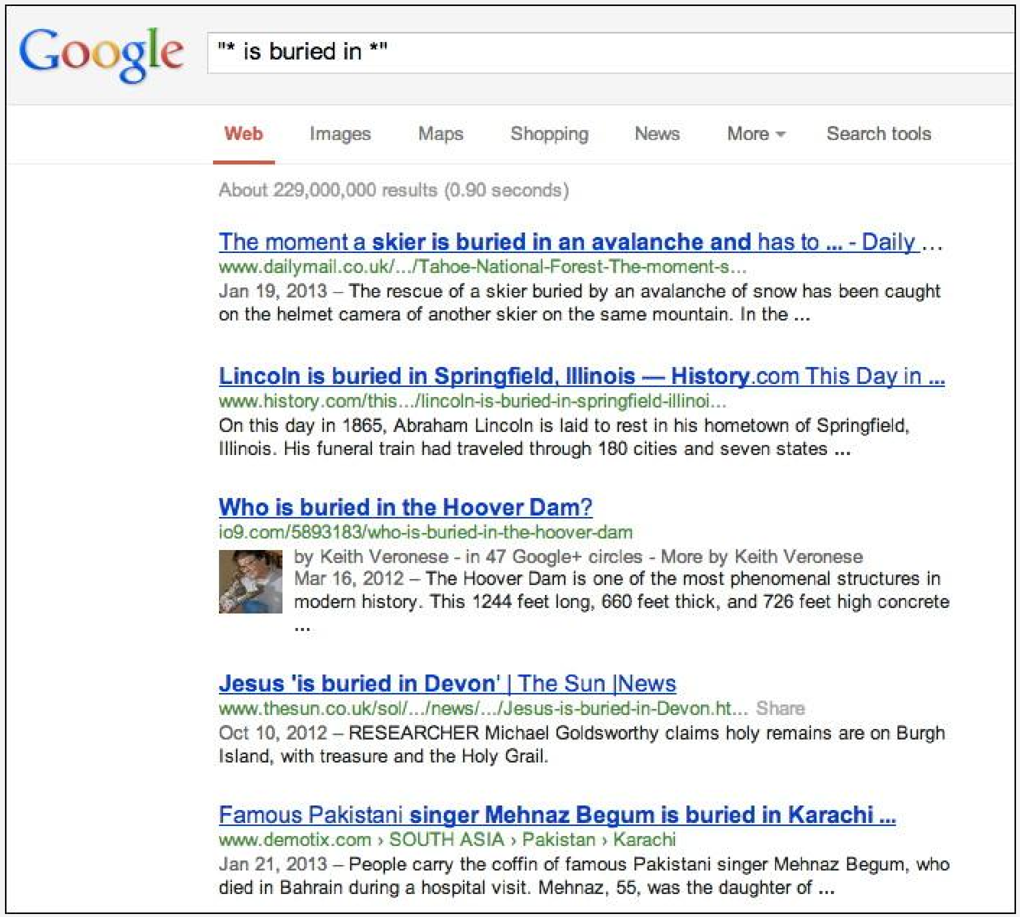
  → The grave of X is in Y

  “Elmira is Mark Twain’s final resting place”

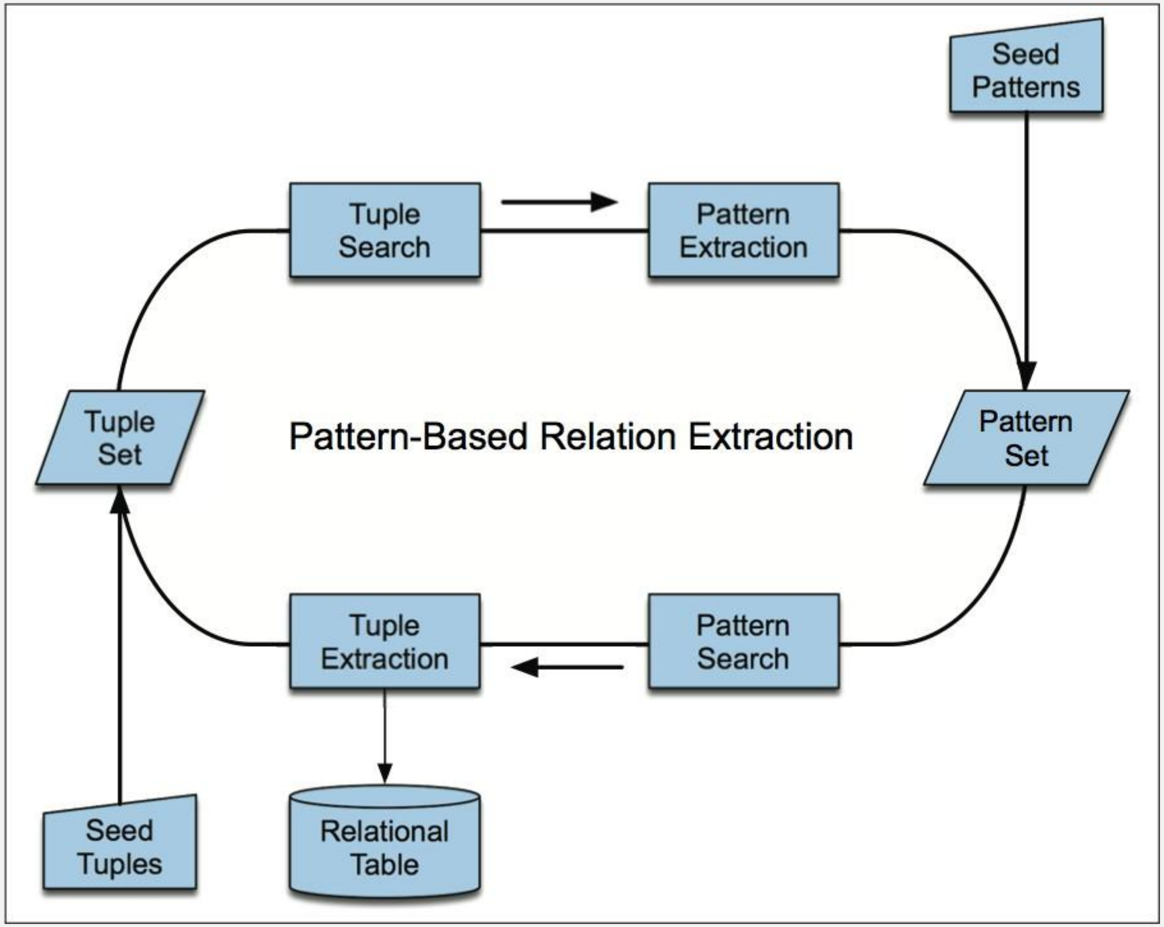
  → Y is X’s final resting place

- Use those patterns to search for new tuples

Bootstrapping example



Bootstrapping relations



DIPRE (Brin 1998)

Extract (author, book) pairs

Start with these 5 seeds:

Author	Book
Isaac Asimov	The Robots of Dawn
David Brin	Startide Rising
James Gleick	Chaos: Making a New Science
Charles Dickens	Great Expectations
William Shakespeare	The Comedy of Errors

Learn these patterns:

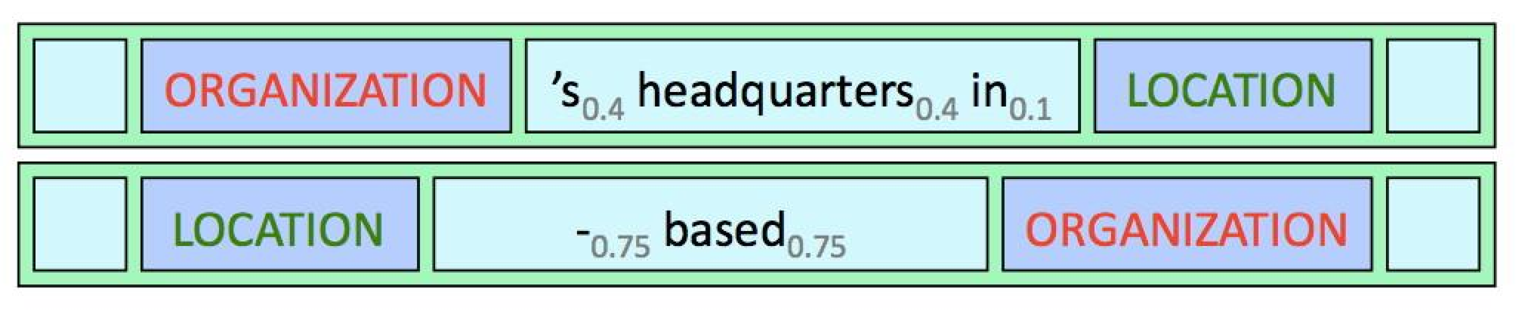
<table>
<tr><th>URL Prefix</th><th>Text Pattern</th></tr>
<tr><td>www.sff.net/locus/c.*</td><td><LI><B>title</B> by author (</td></tr>
<tr><td>dns.city-net.com/~lmann/awards/hugos/1984.html</td><td><i>title</i> by author (</td></tr>
<tr><td>dolphin.upenn.edu/~dcummins/texts/sf-award.htm</td><td>author || title || (</td></tr>
</table>

Iterate: use these patterns to get more instances & patterns…

Snowball (Agichtein & Gravano 2000)

New idea: require that X and Y be named entities of particular types

Organization	Location of Headquarters
Microsoft	Redmond
Exxon	Irving
IBM	Armonk
Boeing	Seattle
Intel	Santa Clara



Bootstrapping problems

- Requires that we have seeds for each relation
  - Sensitive to original set of seeds
- Big problem of semantic drift at each iteration
- Precision tends to be not that high
- Generally have lots of parameters to be tuned
- No probabilistic interpretation
  - Hard to know how confident to be in each result

Relation extraction: 5 easy methods

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

Supervised relation extraction

For each pair of entities in a sentence, predict the *relation type* (if any) that holds between them.

The supervised approach requires:

- Defining an inventory of relation types
- Collecting labeled training data (the hard part!)
- Designing a feature representation
- Choosing a classifier: Naïve Bayes, MaxEnt, SVM, ...
- Evaluating the results

An inventory of relation types

Type	Subtype
ART (artifact)	User-Owner-Inventor-Manufacturer
GEN-AFF (General affiliation)	Citizen-Resident-Religion-Ethnicity, Org-Location
METONYMY*	*None*
ORG-AFF (Org-affiliation)	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
PART-WHOLE (part-to-whole)	Artifact, Geographical, Subsidiary
PER-SOC* (person-social)	Business, Family, Lasting-Personal
PHYS* (physical)	Located, Near

Relation types used in the ACE 2008 evaluation

Labeled training data

Source	Training epoch	Approximate size
English Resources		
Broadcast News	3/03 – 6/03	55,000 words
Broadcast Conversations	3/03 – 6/03	40,000 words
Newswire	3/03 – 6/03	50,000 words
Weblog	11/04 – 2/05	40,000 words
Usenet	11/04 – 2/05	40,000 words
Conversational Telephone Speech	11/04-12/04 (differentiated by topic vs. eval)	40,000 words
Arabic Resources		
Broadcast News	10/00 – 12/00	30,000+ words
Newswire	10/00 – 12/00	55,000+ words
Weblog	11/04 – 2/05	20,000+ words

Datasets used in the ACE 2008 evaluation

Feature representations

- Lightweight features — require little pre-processing
  - Bags of words & bigrams between, before, and after the entities
  - Stemmed versions of the same
  - The types of the entities
  - The distance (number of words) between the entities
- Medium-weight features — require base phrase chunking
  - Base-phrase chunk paths
  - Bags of chunk heads
- Heavyweight features — require full syntactic parsing
  - Dependency-tree paths between the entities
  - Constituent-tree paths between the entities
  - Tree distance between the entities
  - Presence of particular constructions in a constituent structure

Classifiers

Now use any (multiclass) classifier you like:

- multiclass SVM
- MaxEnt (aka multiclass logistic regression)
- Naïve Bayes
- etc.

Zhou et al. 2005 results

Type	Subtype	#Testing Instances	#Correct	#Error	P	R	F
AT		**392**	**224**	**105**	**68.1**	**57.1**	**62.1**
	Based-In	85	39	10	79.6	45.9	58.2
	Located	241	132	120	52.4	54.8	53.5
	Residence	66	19	9	67.9	28.8	40.4
NEAR		**35**	**8**	**1**	**88.9**	**22.9**	**36.4**
	Relative-Location	35	8	1	88.9	22.9	36.4
PART		**164**	**106**	**39**	**73.1**	**64.6**	**68.6**
	Part-Of	136	76	32	70.4	55.9	62.3
	Subsidiary	27	14	23	37.8	51.9	43.8
ROLE		**699**	**443**	**82**	**84.4**	**63.4**	**72.4**
	Citizen-Of	36	25	8	75.8	69.4	72.6
	General-Staff	201	108	46	71.1	53.7	62.3
	Management	165	106	72	59.6	64.2	61.8
	Member	224	104	36	74.3	46.4	57.1
SOCIAL		**95**	**60**	**21**	**74.1**	**63.2**	**68.5**
	Other-Professional	29	16	32	33.3	55.2	41.6
	Parent	25	17	0	100	68.0	81.0

Table 4: Performance of different relation types and major subtypes in the test data

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, languages
- 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

Distant supervision paradigm

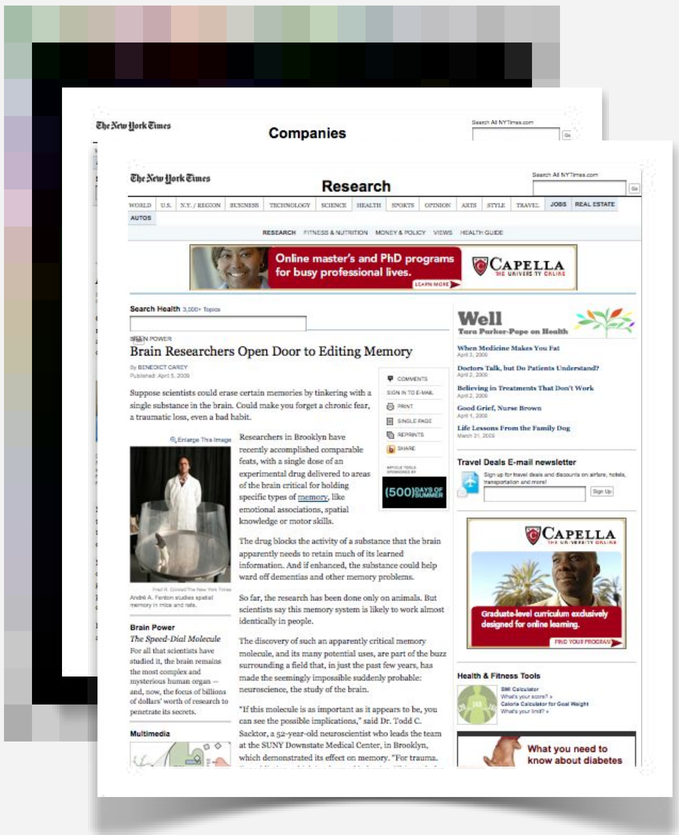
Snow, Jurafsky, Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. NIPS 17

Mintz, Bills, Snow, Jurafsky. 2009. Distant supervision for relation extraction without labeled data. ACL-2009.

- 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 training examples
  - instead of hand-creating a few seed tuples (bootstrapping)
  - instead of using hand-labeled corpus (supervised)

Hypernyms via distant supervision

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

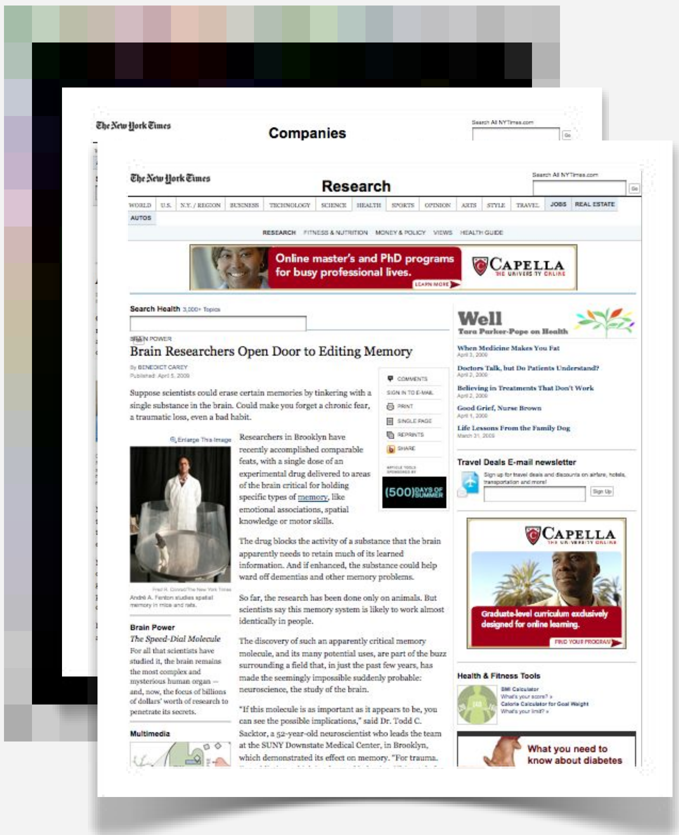


This yields high-signal examples like:

"...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…"

Learning hypernym patterns

1. Take 6M newswire sentences

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

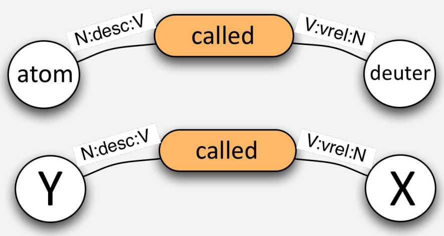
2. Collect noun pairs

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

3. Is pair a hypernym in WordNet?

   14,387 yes; 737,924 no

4. Parse the sentences



5. Extract patterns

   69,592 dependency paths with >5 pairs

6. Train classifier on patterns

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

One of 70,000 patterns

Pattern: <superordinate> called <subordinate>
or: <Y> called <X>

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…

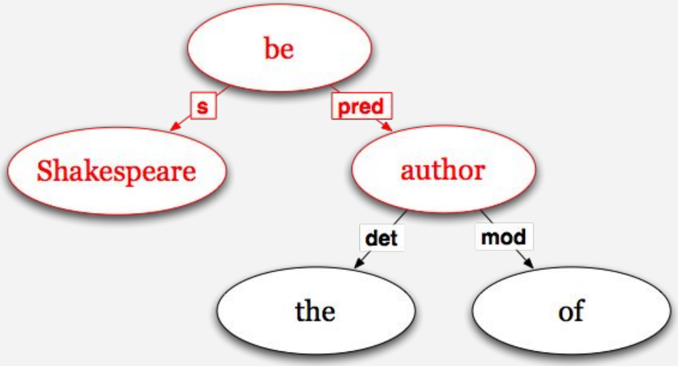
slide adapted from Rion Snow

Syntactic dependency paths

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

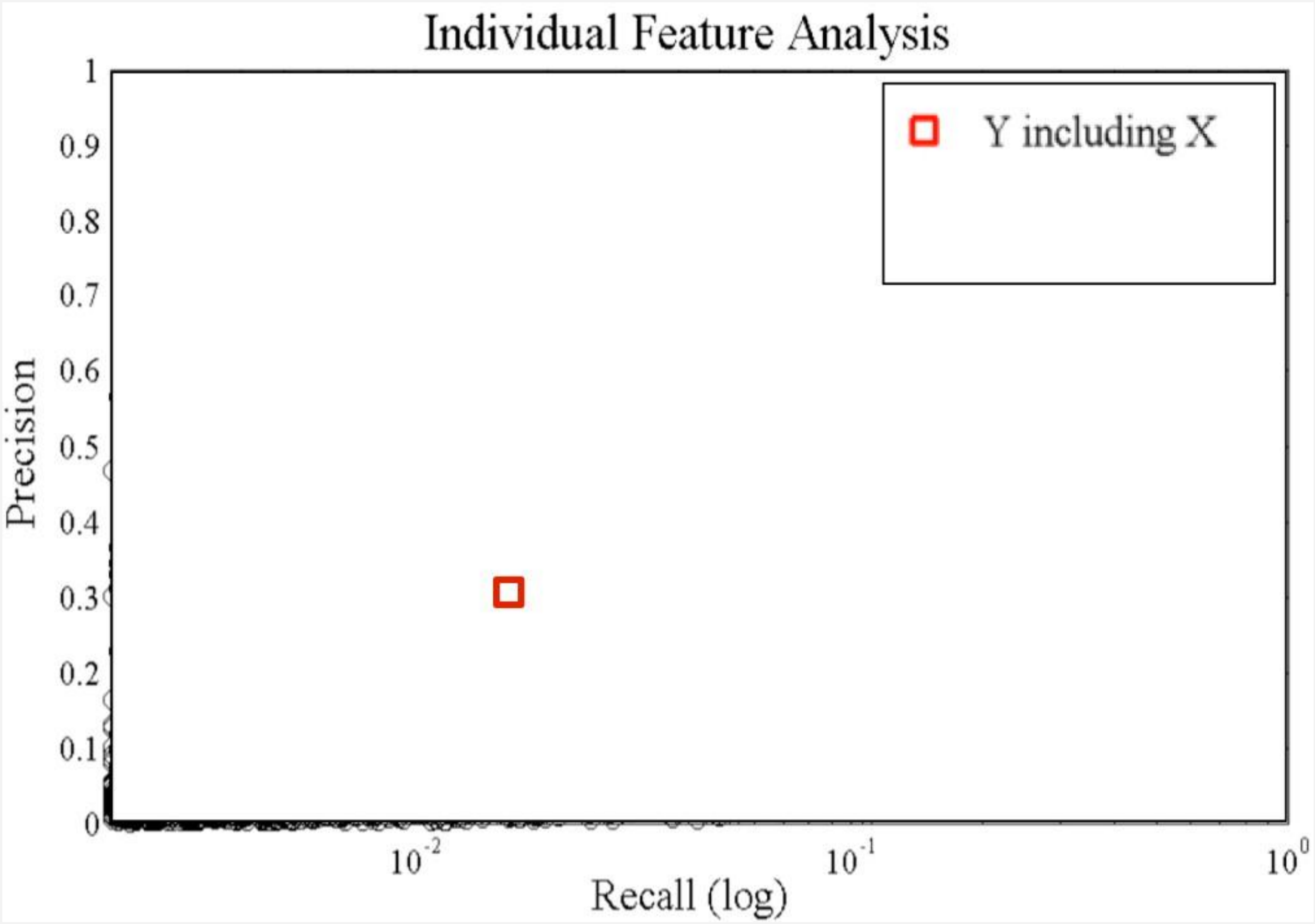
Example word pair: (Shakespeare, author)
Example sentence: "Shakespeare was the author of several plays..."

Minipar parse:

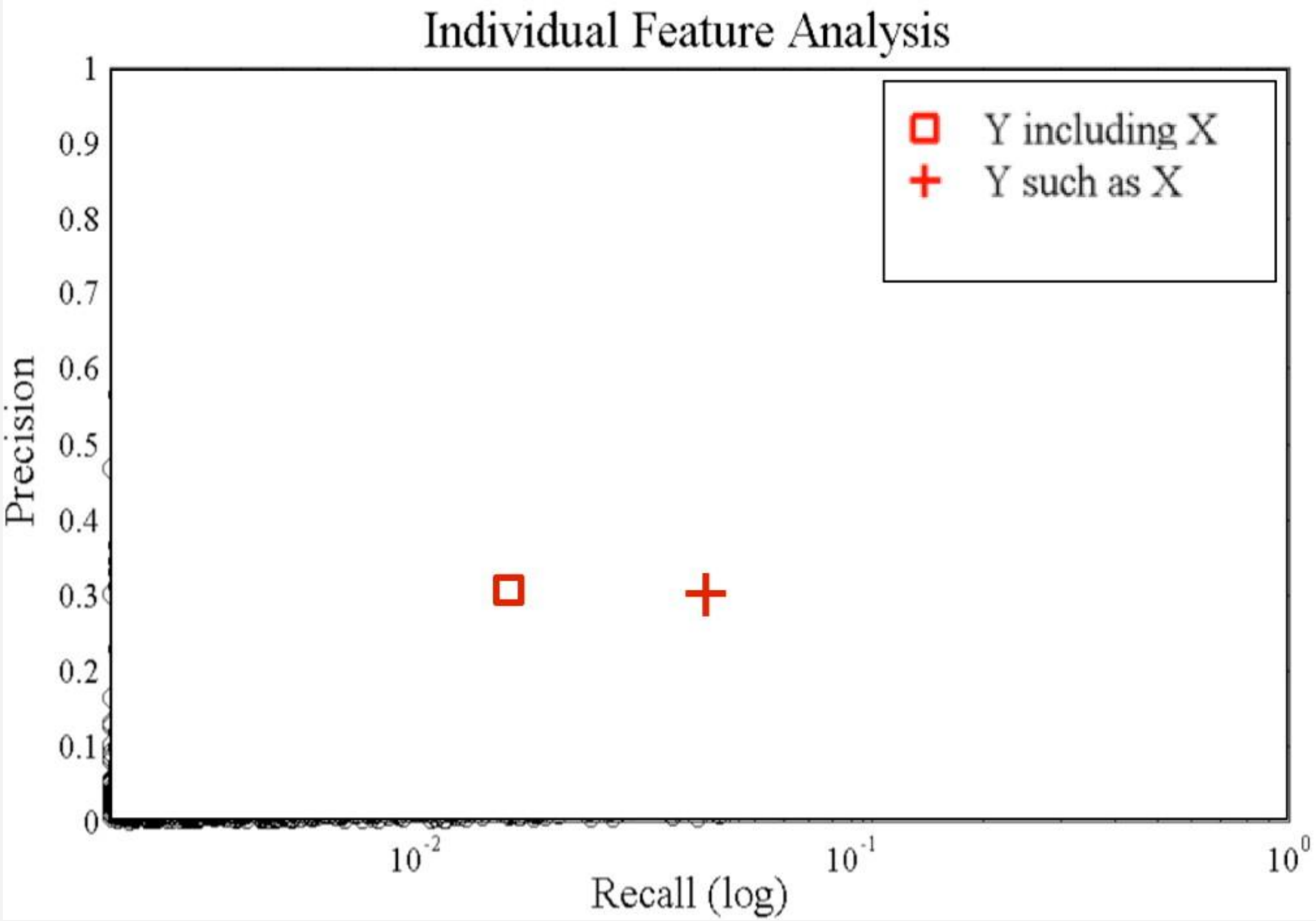


Extract shortest path:
-N:s:VBE, be, VBE:pred:N

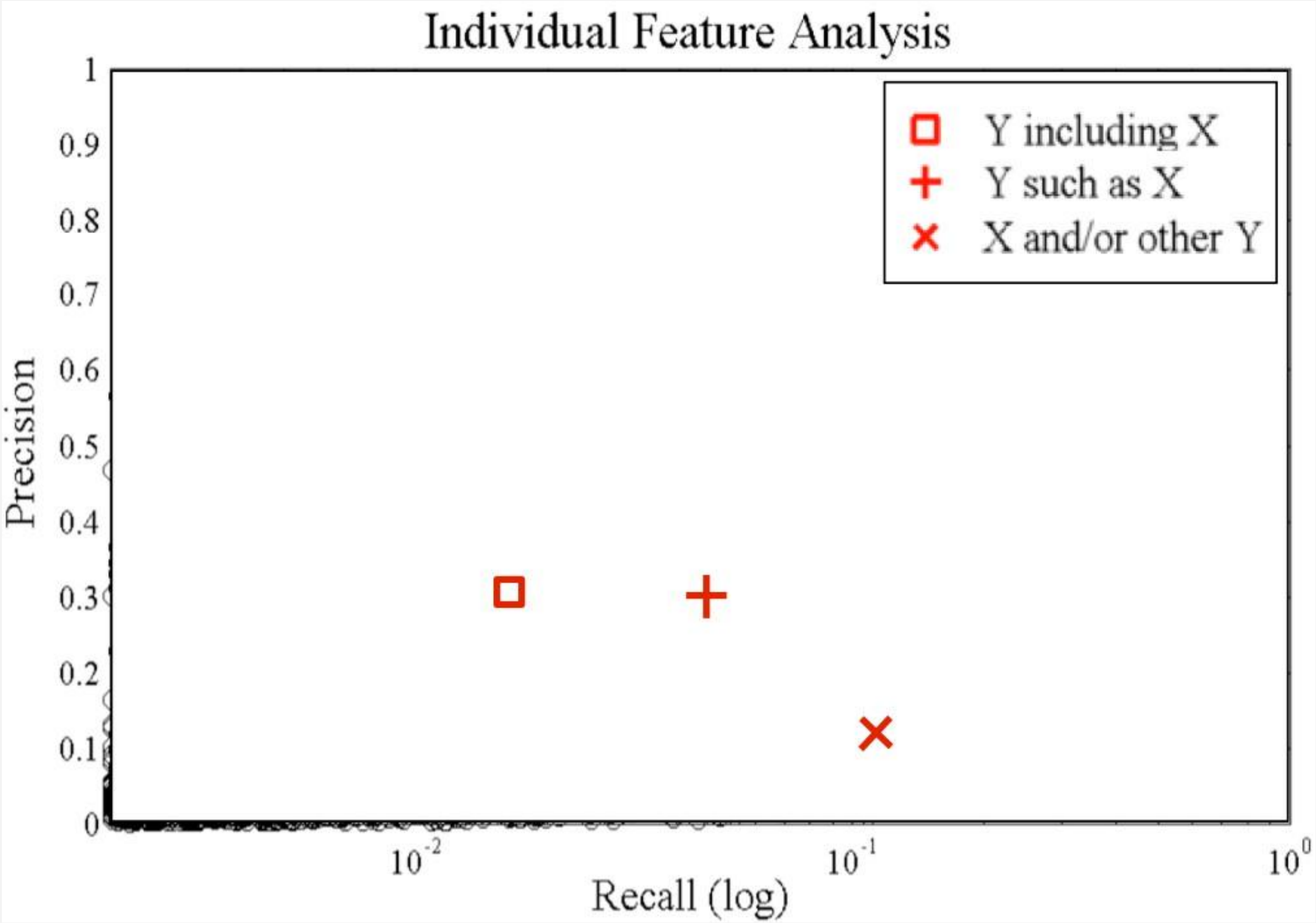
P/R of hypernym extraction patterns



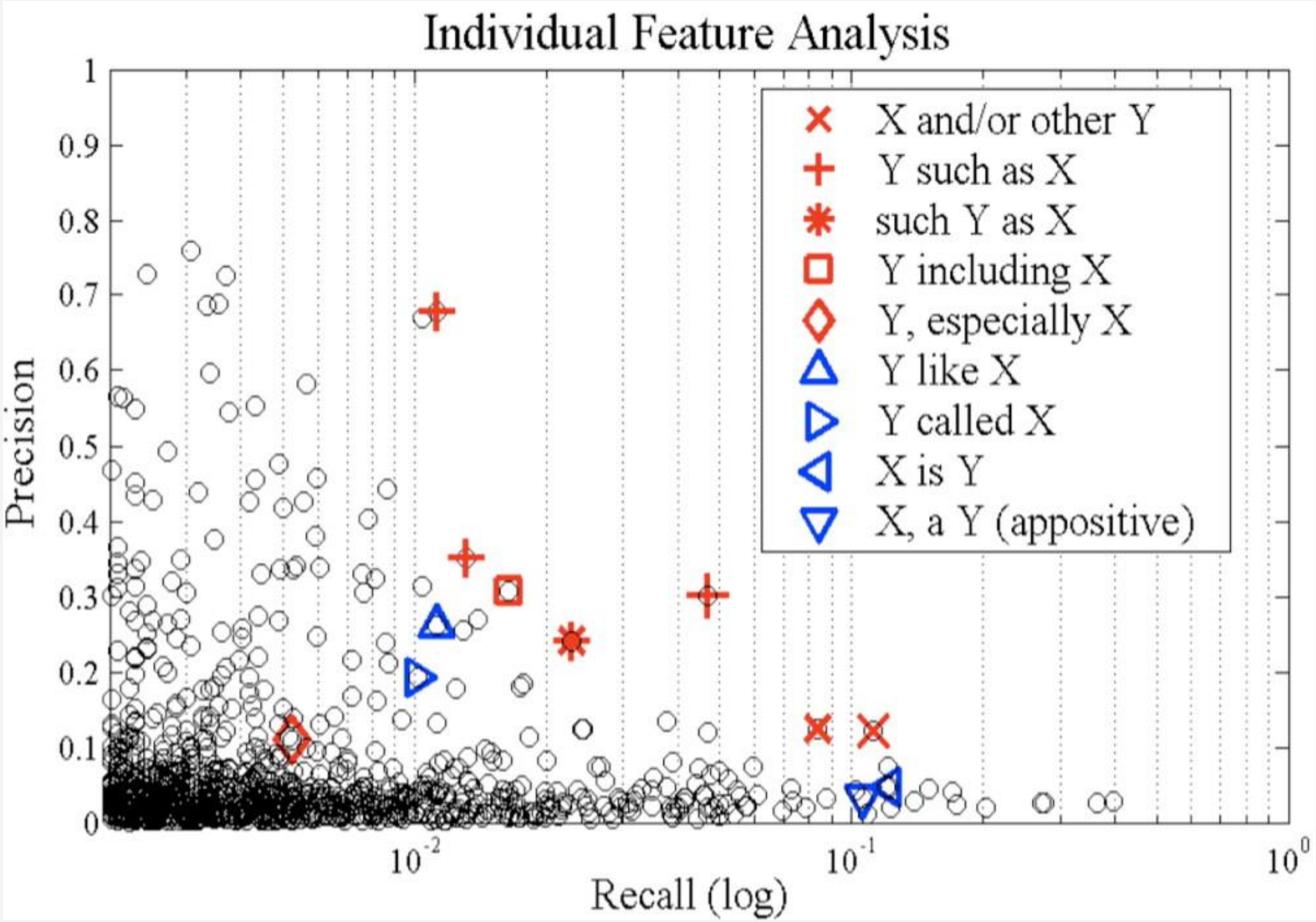
P/R of hypernym extraction patterns



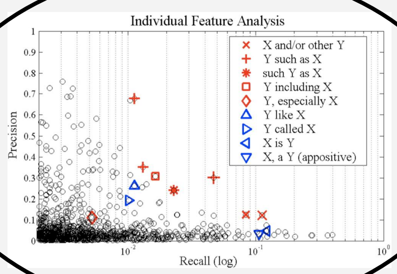
P/R of hypernym extraction patterns



P/R of hypernym extraction patterns

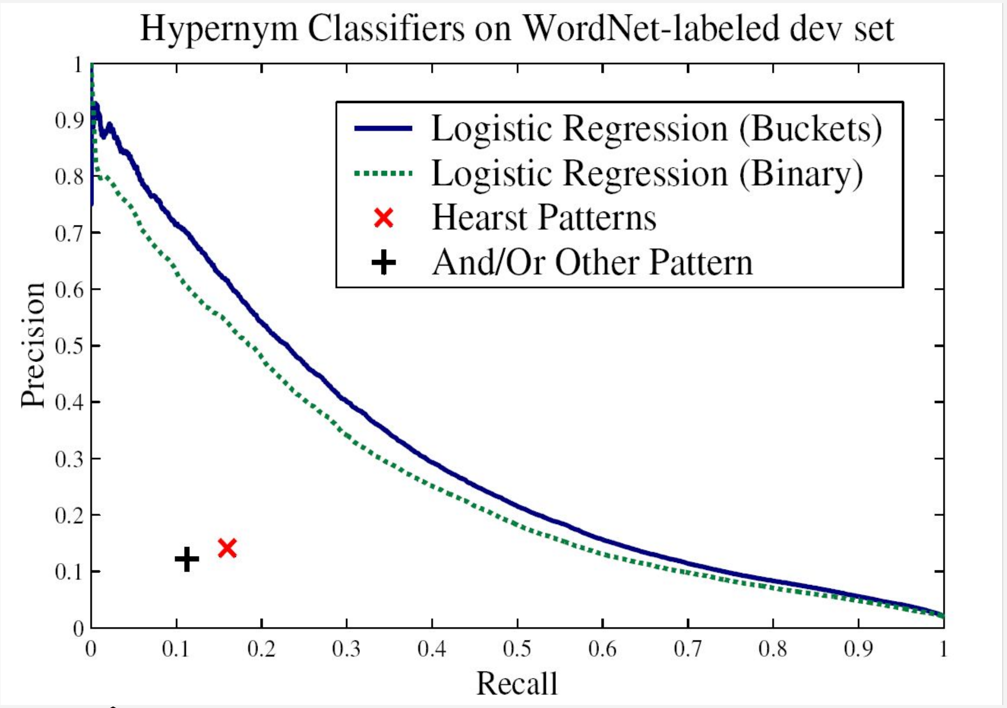


P/R of hypernym classifier



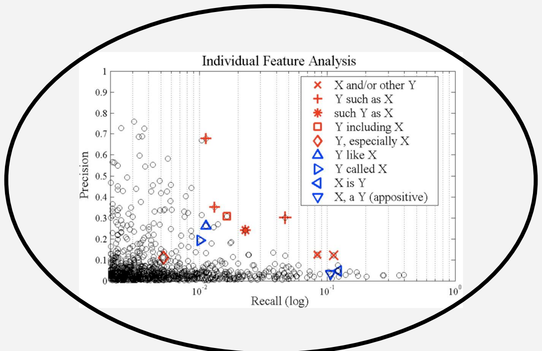
logistic regression

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



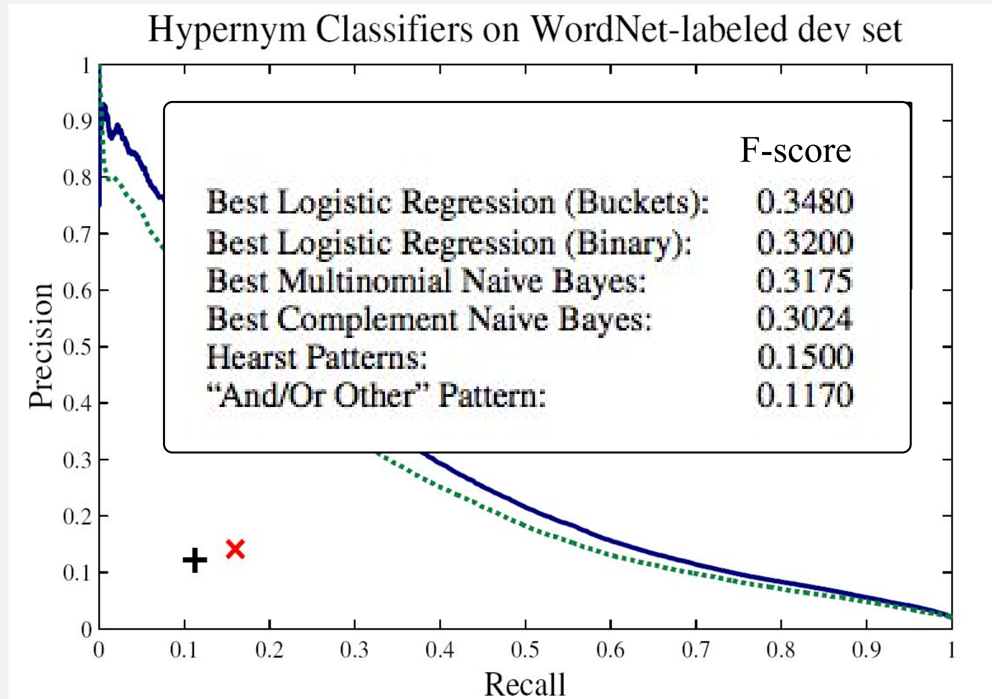
10-fold Cross Validation on
14,000 WordNet-Labeled Pairs

P/R of hypernym classifier



logistic regression

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

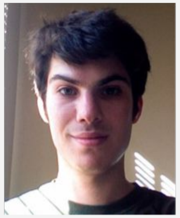


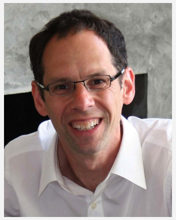
10-fold Cross Validation on
14,000 WordNet-Labeled Pairs

slide adapted from Rion Snow

What about other relations?

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



Training set

Freebase

102 relations
940,000 entities
1.8 million instances

Corpus

WIKIPEDIA

1.8 million articles
25.7 million sentences

Frequent Freebase relations

Relation name	Size	Example
/people/person/nationality	281,107	John Dugard, South Africa
/location/location/contains	253,223	Belgium, Nijlen
/people/person/profession	208,888	Dusa McDuff, Mathematician
/people/person/place_of_birth	105,799	Edwin Hubble, Marshfield
/dining/restaurant/cuisine	86,213	MacAyo's Mexican Kitchen, Mexican
/business/business_chain/location	66,529	Apple Inc., Apple Inc., South Park, NC
/biology/organism_classification_rank	42,806	Scorpaeniformes, Order
/film/film/genre	40,658	Where the Sidewalk Ends, Film noir
/film/film/language	31,103	Enter the Phoenix, Cantonese
/biology/organism_higher_classification	30,052	Calopteryx, Calopterygidae
/film/film/country	27,217	Turtle Diary, United States
/film/writer/film	23,856	Irving Shulman, Rebel Without a Cause
/film/director/film	23,539	Michael Mann, Collateral
/film/producer/film	22,079	Diane Eskenazi, Aladdin
/people/deceased_person/place_of_death	18,814	John W. Kern, Asheville
/music/artist/origin	18,619	The Octopus Project, Austin
/people/person/religion	17,582	Joseph Chartrand, Catholicism
/book/author/works_written	17,278	Paul Auster, Travels in the Scriptorium
/soccer/football_position/players	17,244	Midfielder, Chen Tao
/people/deceased_person/cause_of_death	16,709	Richard Daintree, Tuberculosis
/book/book/genre	16,431	Pony Soldiers, Science fiction
/film/film/music	14,070	Stavisky, Stephen Sondheim
/business/company/industry	13,805	ATS Medical, Health care

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)

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

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

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.

Result: roughly balanced data.

Corpus text

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

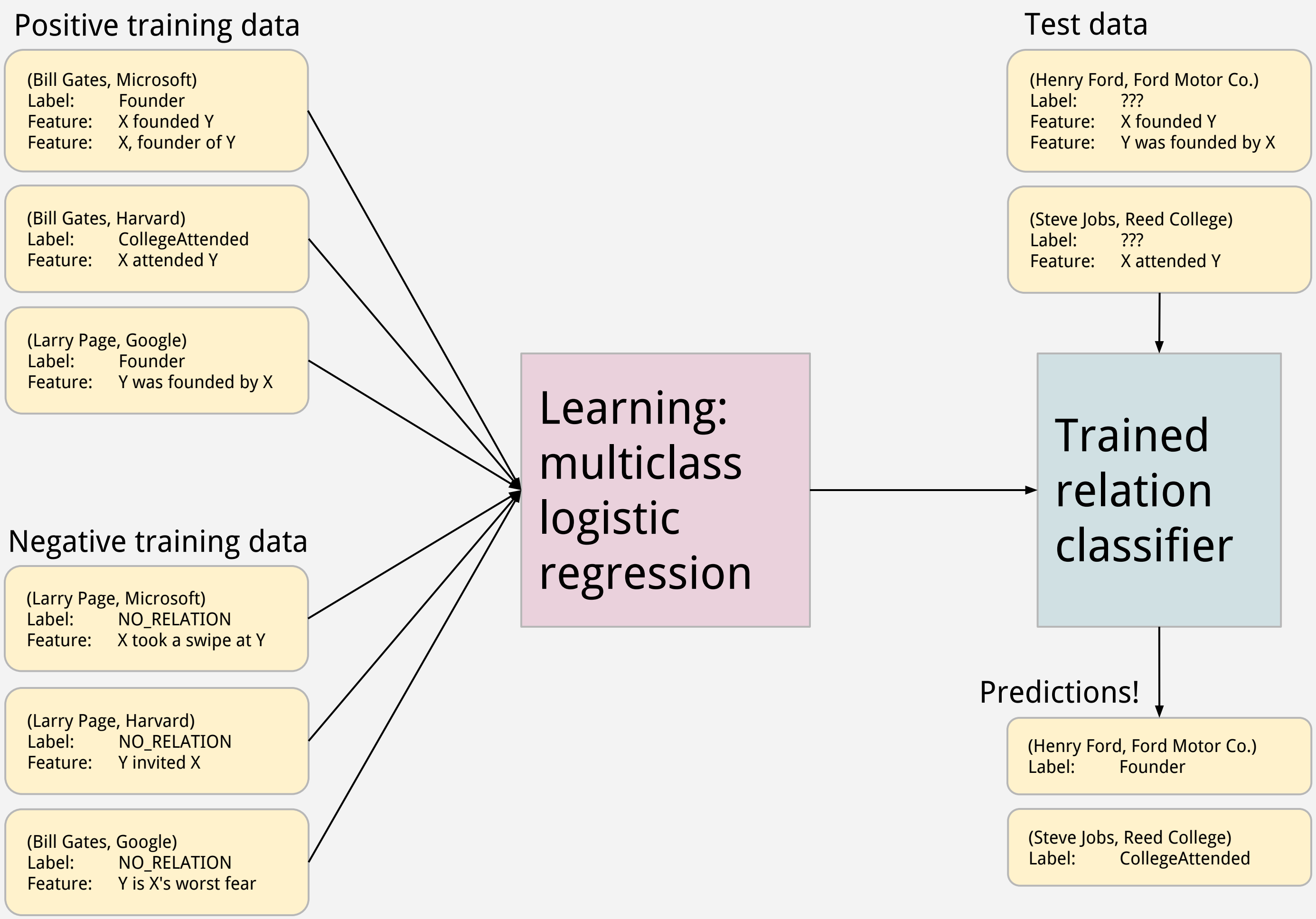
Training data

(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

The experiment



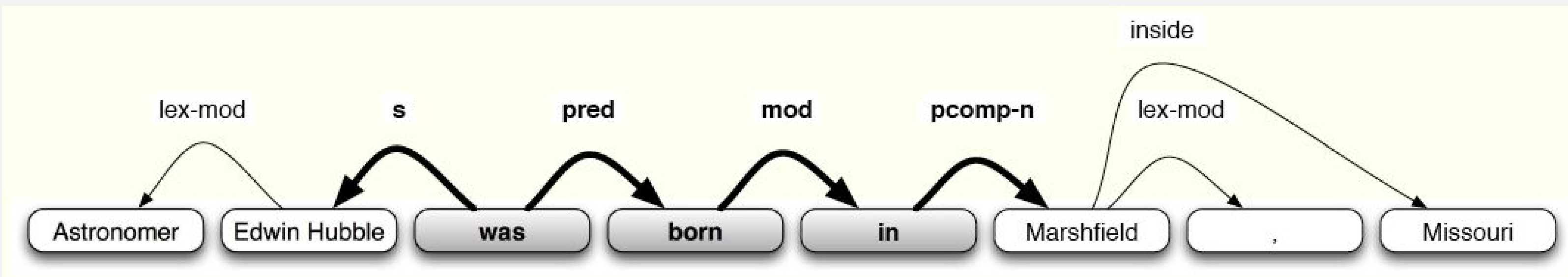
Benefits of distant supervision

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

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

Lexical and syntactic features

Astronomer Edwin Hubble was born in Marshfield, Missouri.



Feature type	Left window	NE1	Middle	NE2	Right window
Lexical	[]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[]
Lexical	[Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[,]
Lexical	[#PAD#, Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[, Missouri]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]

High-weight features

Relation	Feature type	Left window	NE1	Middle	NE2	Right window
/architecture/structure/architect	LEX$\curvearrowleft$		ORG	, the designer of the	PER	
	SYN	designed $\Uparrow_s$	ORG	$\Uparrow_s$ designed $\Downarrow_{by-subj}$ by $\Downarrow_{pcn}$	PER	$\Uparrow_s$ designed
/book/author/works_written	LEX		PER	s novel	ORG	
	SYN		PER	$\Uparrow_{pcn}$ by $\Uparrow_{mod}$ story $\Uparrow_{pred}$ is $\Downarrow_s$	ORG	
/book/book_edition/author_editor	LEX$\curvearrowleft$		ORG	s novel	PER	
	SYN		PER	$\Uparrow_{nn}$ series $\Downarrow_{gen}$	PER	
/business/company/founders	LEX		ORG	co - founder	PER	
	SYN		ORG	$\Uparrow_{nn}$ owner $\Downarrow_{person}$	PER	
/business/company/place_founded	LEX$\curvearrowleft$		ORG	- based	LOC	
	SYN		ORG	$\Uparrow_s$ founded $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/film/film/country	LEX		PER	, released in	LOC	
	SYN	opened $\Uparrow_s$	ORG	$\Uparrow_s$ opened $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ opened
/geography/river/mouth	LEX		LOC	, which flows into the	LOC	
	SYN	the $\Downarrow_{det}$	LOC	$\Uparrow_s$ is $\Downarrow_{pred}$ tributary $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	$\Downarrow_{det}$ the
/government/political_party/country	LEX$\curvearrowleft$		ORG	politician of the	LOC	
	SYN	candidate $\Uparrow_{nn}$	ORG	$\Uparrow_{nn}$ candidate $\Downarrow_{mod}$ for $\Downarrow_{pcn}$	LOC	$\Uparrow_{nn}$ candidate
/influence/influence_node/influenced	LEX$\curvearrowleft$		PER	, a student of	PER	
	SYN	of $\Uparrow_{pcn}$	PER	$\Uparrow_{pcn}$ of $\Uparrow_{mod}$ student $\Uparrow_{appo}$	PER	$\Uparrow_{pcn}$ of
/language/human_language/region	LEX		LOC	- speaking areas of	LOC	
	SYN		LOC	$\Uparrow_{lex-mod}$ speaking areas $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	
/music/artist/origin	LEX$\curvearrowleft$		ORG	based band	LOC	
	SYN	is $\Uparrow_s$	ORG	$\Uparrow_s$ is $\Downarrow_{pred}$ band $\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ is
/people/deceased_person/place_of_death	LEX		PER	died in	LOC	
	SYN	hanged $\Uparrow_s$	PER	$\Uparrow_s$ hanged $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ hanged
/people/person/nationality	LEX		PER	is a citizen of	LOC	
	SYN		PER	$\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	
/people/person/parents	LEX		PER	, son of	PER	
	SYN	father $\Uparrow_{gen}$	PER	$\Uparrow_{gen}$ father $\Downarrow_{person}$	PER	$\Uparrow_{gen}$ father
/people/person/place_of_birth	LEX$\curvearrowleft$		PER	is the birthplace of	PER	
	SYN		PER	$\Uparrow_s$ born $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/people/person/religion	LEX		PER	embraced	LOC	
	SYN	convert $\Downarrow_{appo}$	PER	$\Downarrow_{appo}$ convert $\Downarrow_{mod}$ to $\Downarrow_{pcn}$	LOC	$\Downarrow_{appo}$ convert

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

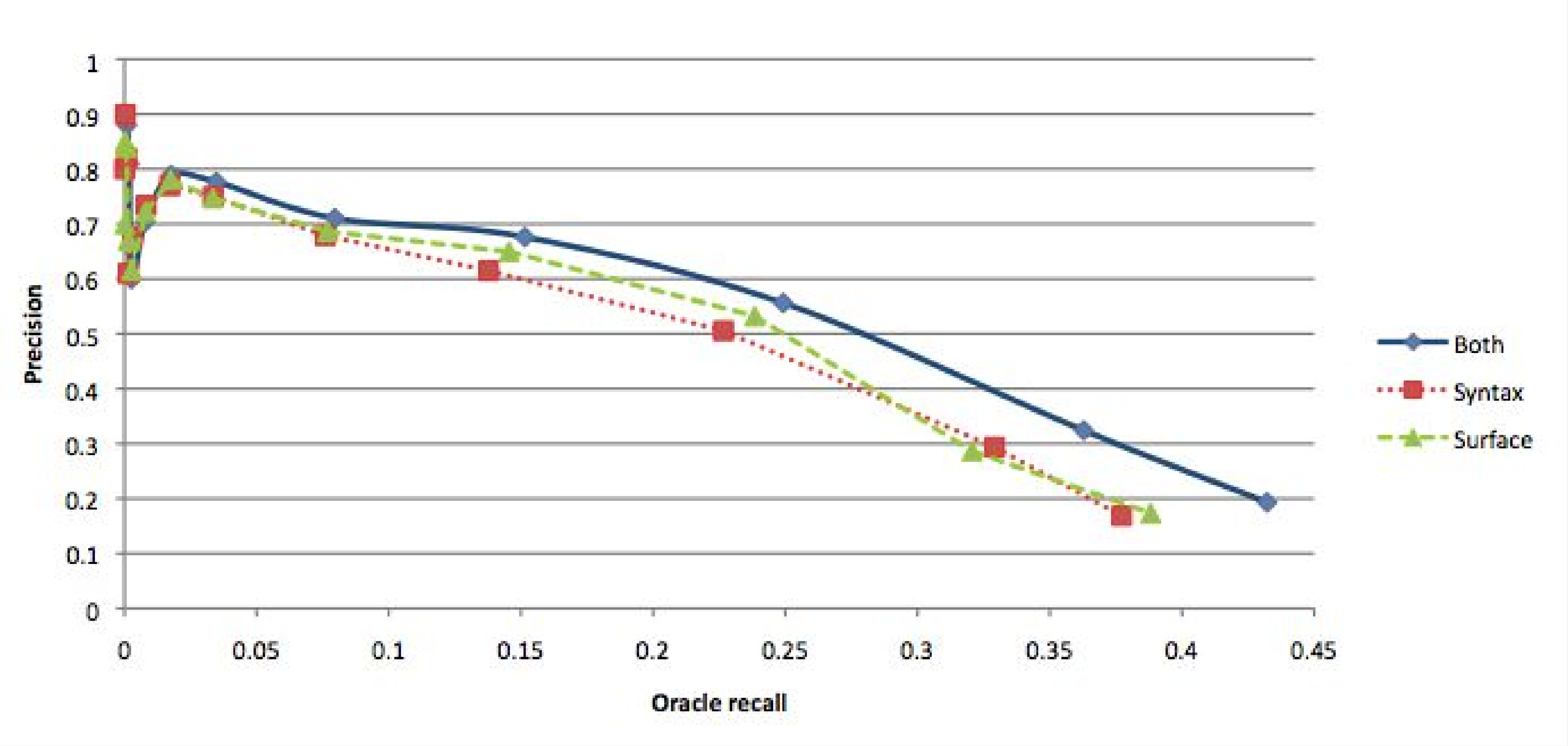
Relation name	New instance
/location/location/contains	Paris, Montmartre
/location/location/contains	Ontario, Fort Erie
/music/artist/origin	Mighty Wagon, Cincinnati
/people/deceased_person/place_of_death	Fyodor Kamensky, Clearwater
/people/person/nationality	Marianne Yvonne Heemskerk, Netherlands
/people/person/place_of_birth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

Evaluation

- Held-out evaluation
  - Train on 50% of gold-standard Freebase relation instances, test on other 50%
  - Used to tune parameters quickly without having to wait for human evaluation

- Human evaluation
  - Performed by evaluators on Amazon Mechanical Turk
  - Calculated precision at 100 and 1000 recall levels for the ten most common relations

Held-out evaluation



Automatic evaluation on 900K instances of 102 Freebase relations.
Precision for three different feature sets is reported at various recall levels.

Human evaluation

Precision, using Mechanical Turk labelers:

Relation name	100 instances			1000 instances		
	Syn	Lex	Both	Syn	Lex	Both
/film/director/film	**0.49**	0.43	0.44	**0.49**	0.41	0.46
/film/writer/film	**0.70**	0.60	0.65	**0.71**	0.61	0.69
/geography/river/basin_countries	0.65	0.64	**0.67**	**0.73**	0.71	0.64
/location/country/administrative_divisions	0.68	0.59	**0.70**	**0.72**	0.68	**0.72**
/location/location/contains	0.81	**0.89**	0.84	**0.85**	0.83	0.84
/location/us_county/county_seat	0.51	0.51	**0.53**	0.47	**0.57**	0.42
/music/artist/origin	0.64	0.66	**0.71**	0.61	**0.63**	0.60
/people/deceased_person/place_of_death	0.80	0.79	**0.81**	0.80	**0.81**	0.78
/people/person/nationality	0.61	0.70	**0.72**	0.56	0.61	**0.63**
/people/person/place_of_birth	**0.78**	0.77	**0.78**	0.88	0.85	**0.91**
Average	0.67	0.66	**0.69**	**0.68**	0.67	0.67

- 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

Distant supervision: takeaways

- The distant supervision approach uses a database of known relation instances as a source of supervision
- We're classifying pairs of entities, not pairs of entity mentions
- The features for a pair of entities describe the patterns in which the two entities have co-occurred across many sentences in a large corpus
- Can make use of 100x or even 1000x more data than in the supervised paradigm

Relation extraction: 5 easy methods

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

OpenIE at U. Washington

- Influential work by Oren Etzioni's group
- 2005: KnowItAll
  - Generalizes Hearst patterns to other relations
  - Requires zillions of search queries; very slow
- 2007: TextRunner
  - No predefined relations; highly scalable; imprecise
- 2011: ReVerb
  - Improves precision using simple heuristics
- 2012: Ollie
  - Operates on Stanford dependencies, not just tokens
- 2013: OpenIE 4.0

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
  - Parsing is relatively expensive, so can't run on whole web
  - For each pair of base noun phrases NP_i and NP_j
  - Extract all tuples t = (NP_i, $relation_{i,j}$, NP_j)
- Label each tuple based on features of parse:
  - Positive iff the dependency path between the NPs is short, and 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 tag sequences, number of 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

    → ⟨*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
  ⟨*scientists*, *are studying*, *stars*⟩ → 17
- Compute likelihood of each tuple
  - given the counts for each relation
  - and the number of sentences
  - and a combinatoric balls & 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 examples

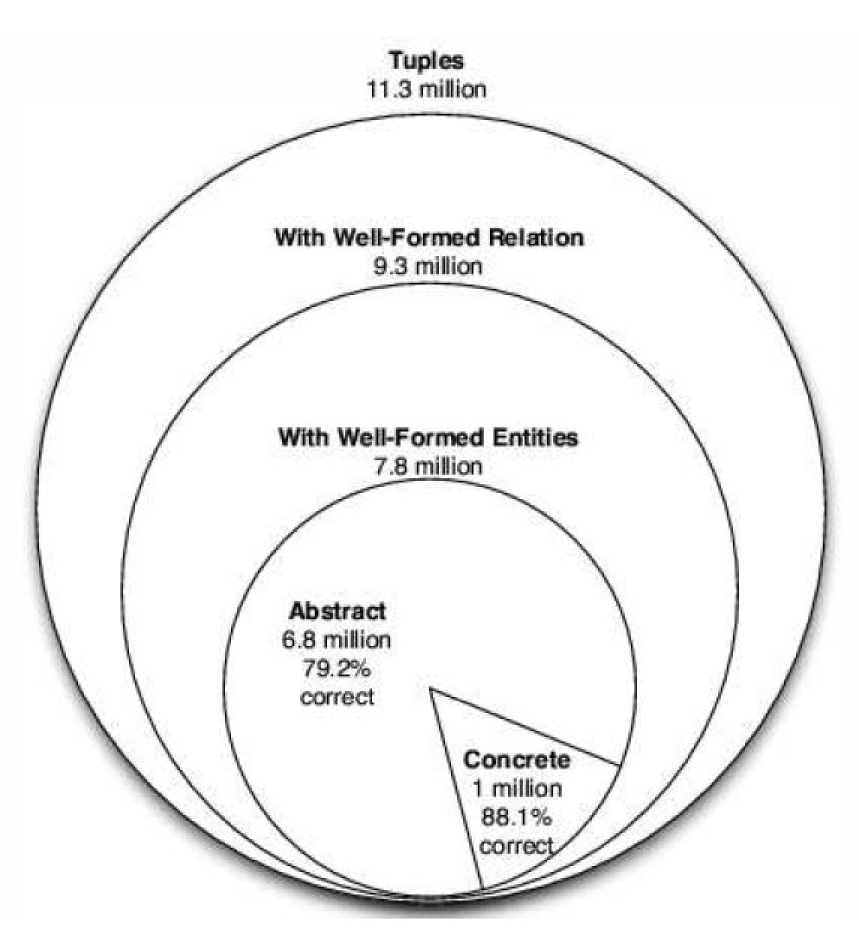
Probability	Count	Arg1	Predicate	Arg2
0.98	59	Smith	invented	the margherita
0.97	49	Al Gore	invented	the Internet
0.97	44	manufacturing plant	first invented	the automatic revolver
0.97	41	Alexander Graham Bell	invented	the telephone
0.97	36	Thomas Edison	invented	light bulbs
0.97	29	Eli Whitney	invented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

slide from Oren Etzioni

TextRunner results

- From corpus of 9M web pages = 133M sentences
- Extracted 60.5M tuples
- Filtered down to 11.3M tuples
  - High probability, good support, but not *too* frequent
- Evaluated by manually inspecting a sample
  - Not well formed:
    ⟨*demands, of securing, border*⟩⟨*29, dropped, instruments*⟩
  - Abstract:
    ⟨*Einstein, derived, theory*⟩⟨*executive, hired by, company*⟩
  - Concrete:
    ⟨*Tesla, invented, coil transformer*⟩

Evaluating TextRunner



Problems with TextRunner

TextRunner's extractions are not very precise!

Many of TextRunner's problems with precision come from two sources:

- Incoherent relations (~13%)
- Uninformative extractions (~7%)

(ReVerb aims to fix these problems …)

Incoherent relations

Extraction and simplification heuristics often yield relations that make no sense:

Extendicare agreed to buy Arbor Health Care for about US $432 million in cash and assumed debt.

→ (Arbor Health Care, for assumed, debt)

Uninformative extractions

Light-verb constructions (LVCs) are not handled properly, and critical information is lost:

Faust made a deal with the devil.

→ (Faust, made, a deal)
vs. (Faust, made a deal with, the devil)

is has made took gave got	vs.	is an album by, is the author of, is a city in has a population of, has a Ph.D. in, has a cameo in made a deal with, made a promise to took place in, took control over, took advantage of gave birth to, gave a talk at, gave new meaning to got tickets to, got a deal on, got funding from

ReVerb's syntactic constraint

ReVerb fixes both problems with a syntactic constraint.
A relation phrase must be longest match to this regexp:

(V | V P | V W* P)+

V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

matches:
invented
located in
has atomic weight of
wants to extend

but not: *for assumed*

ReVerb's lexical constraint

The syntactic constraint has an unfortunate side-effect: matching very long and overly-specific relations.

> *The Obama administration is offering only modest greenhouse gas reduction targets at the conference.*

ReVerb avoids this by imposing a lexical constraint:

Valid relational phrases should take ≥ 20 distinct argument pairs over a large corpus (500M sentences).

ReVerb's confidence function

To assign probabilities to candidate extractions, and improve precision, ReVerb uses a simple classifier.

- Logistic regression
- Trained on 1,000 manually labeled examples
- Few features
- Lightweight features
- Relation-independent

Weight	Feature
1.16	(x, r, y) covers all words in s
0.50	The last preposition in r is *for*
0.49	The last preposition in r is *on*
0.46	The last preposition in r is *of*
0.43	$len(s) \leq 10$ words
0.43	There is a WH-word to the left of r
0.42	r matches VW*P from Figure 1
0.39	The last preposition in r is *to*
0.25	The last preposition in r is *in*
0.23	10 words $< len(s) \leq$ 20 words
0.21	s begins with x
0.16	y is a proper noun
0.01	x is a proper noun
-0.30	There is an NP to the left of x in s
-0.43	20 words $< len(s)$
-0.61	r matches V from Figure 1
-0.65	There is a preposition to the left of x in s
-0.81	There is an NP to the right of y in s
-0.93	Coord. conjunction to the left of r in s

ReVerb relation extraction

Given input sentence with POS tags and NP chunks:

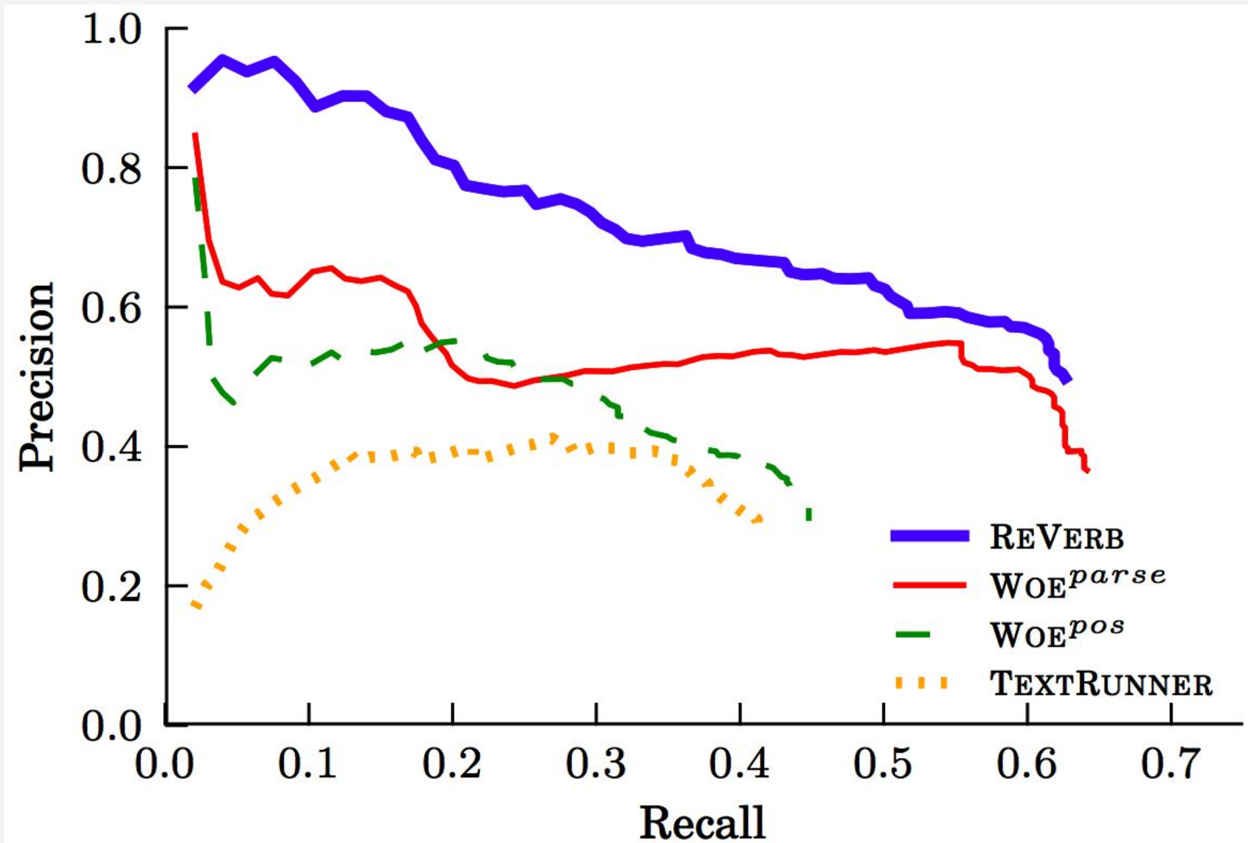
- *Relation extraction*: for each verb *v*, find longest phrase starting with *v* and satisfying both the syntactic constraint and the lexical constraint.
- *Argument extraction*: for each relation phrase, find nearest non-pronoun NPs to left and right.
- *Confidence estimation*: apply classifier to candidate extraction to assign confidence and filter.

ReVerb example

Hudson was born in Hampstead, which is a suburb of London.

→ (Hudson, was born in, Hampstead)
→ (Hampstead, is a suburb of, London)

ReVerb results



Manual evaluation over 500 sentences.

OpenIE demo

http://openie.allenai.org/

Synonymy of relations

TextRunner and ReVerb don't pay much attention to the issue of *synonymy* between relation phrases.

(*airlift*, *alleviates*, *hunger crisis*)
(*hunger crisis*, *is eased by*, *airlift*)
(*airlift*, *helps resolve*, *hunger crisis*)
(*airlift*, *addresses*, *hunger crisis*)

Have we learned four facts, or one?

How to identify (& combine) synonymous relations?

DIRT (Lin & Pantel 2001)

- 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
  - 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.

- X is a solution to Y
  - (5, 1) is a solution to the equation 2x – 3y = 7
  - Nuclear energy is a solution to the energy crisis.

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. Split 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:
  - Extract dependency path between them, as in DIRT
  - 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

Background: LDA topic models

- LDA = Latent Dirichlet Allocation [Blei, Ng, & Jordan 2003]
- A generative model of documents, topics, and words
  - A topic is a multinomial distribution over words
  - Each document has a mixture of topics, sampled from a Dirichlet
  - Each word in the document is sampled from one topic

α : parameter of Dirichlet prior on per-document topic distributions

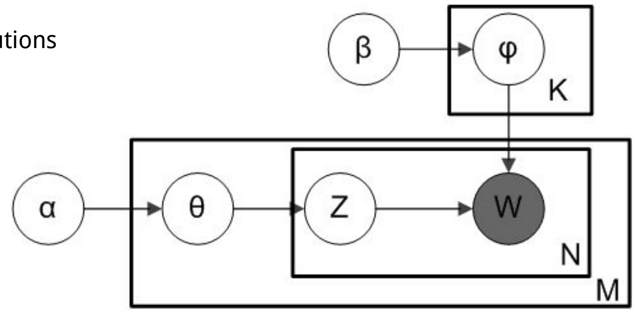
β : parameter of Dirichlet prior on per-topic word distribution

θ_i : topic distribution for document i

φ_k : word distribution for topic k

z_{ij} : topic for jth word in document i

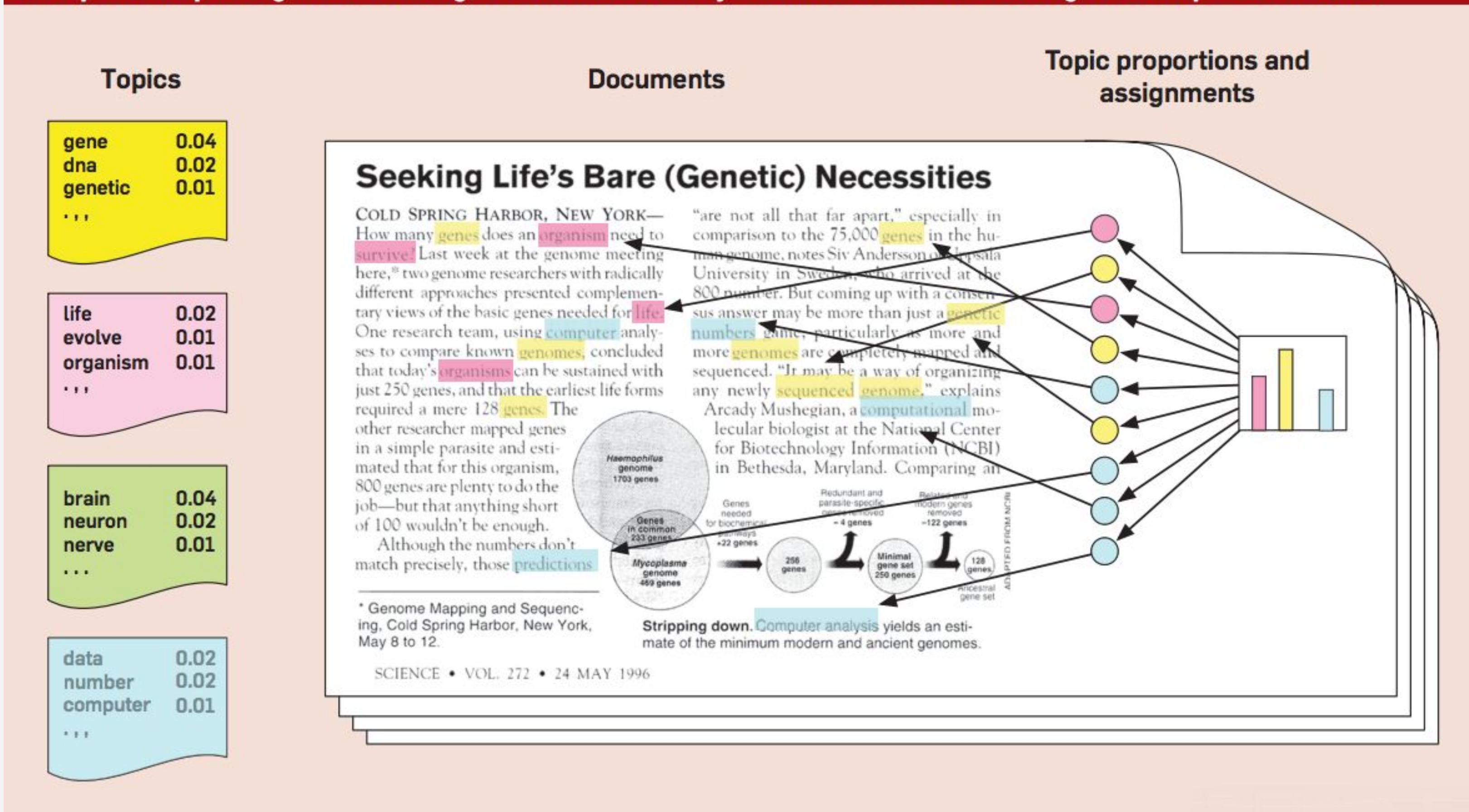
w_{ij} : the specific word



- Inference via variational Bayes or Gibbs sampling
- Off-the-shelf software packages are available

LDA topic models

Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.



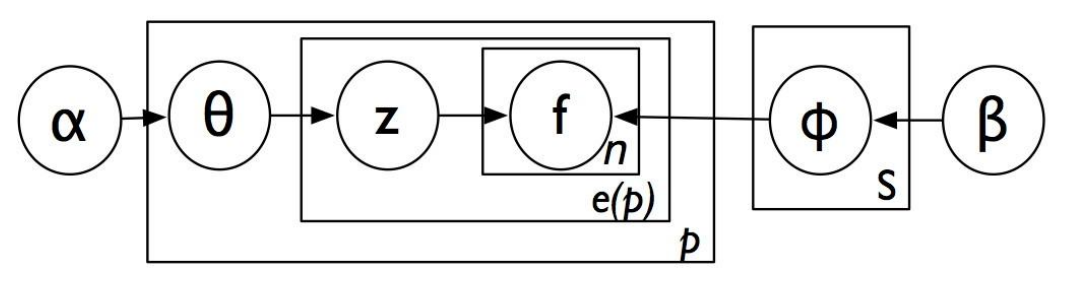
graphic from [Blei 2012]()

Clustering tuples into senses

- Goal: group tuples for each path into coherent sense clusters
- To do this, we apply yet another LDA topic model
  - Not vanilla LDA this time — rather, a slight variant
  - Details on next slide
- Use Gibbs sampling for inference
- Result: each tuple is assigned one topic/sense
- Tuples with the same topic/sense constitute a cluster

The Sense-LDA model

$$\theta_{p_i} \sim Dirichlet(\alpha)$$
$$\phi_z \sim Dirichlet(\beta)$$
$$z_e \sim \text{Multinomial}(\theta_{p_i})$$
$$f_k \sim \text{Multinomial}(\phi_{z_e})$$



- A slight variation on standard LDA (Blei et al. 2003)
- For each path, form ”document” of all its tuples, with features
- For each path/document, sample a multinomial distribution θ over topics/senses from a Dirichlet prior
- For each tuple, sample a topic/sense from θ
- Features are sampled from a topic/sense-specific multinomial
- Features are conditionally independent, given topic/sense

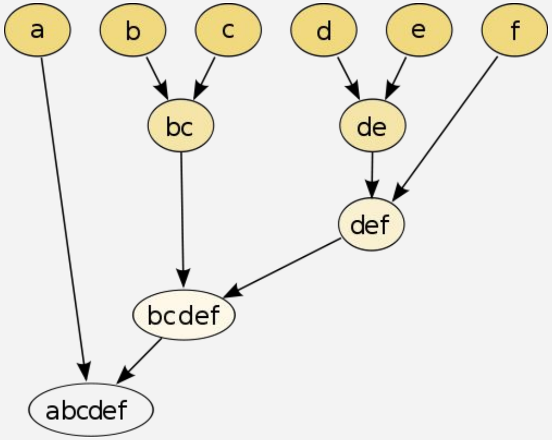
Sense cluster examples

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

Sense clusters for path ”A play B”,
along with sample entity pairs and top features.

Clustering the clusters!

- Now cluster sense clusters from different paths into semantic relations — this is the part most similar to DIRT
- Uses Hierarchical Agglomerative Clustering (HAC)
- Start with minimal clustering, then merge progressively
- Uses cosine similarity between sense-cluster feature vectors
- Uses complete-linkage strategy: similarity between two clusters is min similarity between any pair of items



Semantic relation 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

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

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	$\boxed{0.567}$	0.152	$\boxed{0.367}$	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	$\boxed{\mathbf{0.736}}$	**0.156**	$\boxed{\mathbf{0.422}}$	**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

Automatic evaluation against Freebase

HAC = hierarchical agglomerative clustering alone

(i.e. no sense disambiguation — most similar to DIRT)

Sense clustering adds 17% to precision!

Tell me again why this matters?

The OpenIE approaches (TextRunner, ReVerb) don't have any way to canonicalize relation phrases.

(*Google*, *is based in*, *Mountain View*)
(*Mountain View*, *is home to*, *Google*)
(*Google*, *has its headquarters in*, *Mountain View*)
(*Google*, *is located in*, *Mountain View*)

If your goal is to populate a knowledge base from text on the web, what relation do you add these tuples to?

Yao et al. 2012 helps to resolve this problem.