Learning Relations from the Web

Bill MacCartney
CS224U: Natural Language Understanding
31 January 2012
Stanford

Many thanks for lots of slides from many people,
including Dan Jurafsky, Jim Martin, Oren Etzioni,
Michele Banko, Rion Snow, Mike Mintz, and Steven Bills

Reminder!

- Lit Review paper due in 2 weeks!
- Start forming your project groups!
  - Working alone is fine, of course
  - But collaborating can be more fun!
- Bring your project ideas to office hours
- The ACL Anthology Searchbench may help find relevant literature: http://aclasb.dfki.de/

Extracting structured knowledge

Each article can contain hundreds or thousands of items of knowledge...


Log in / create account

article | discussion | edit this page | history

Lawrence Livermore National Laboratory

From Wikipedia, the free encyclopedia

Coordinates: 37.686024°N 121.709547°W

The **Lawrence Livermore National Laboratory (LLNL)** in Livermore, California is a scientific research laboratory founded by the University of California in 1952. It is funded by the United States Department of Energy (DOE) and managed by Lawrence Livermore National Security, LLC (LLNS), a partnership of the University of California, Bechtel Corporation, Babcock and Wilcox, the URS Corporation, and Battelle Memorial Institute. On October 1, 2007 LLNS assumed management of LLNL from the University of California, which had exclusively managed and operated the Laboratory since its inception 55 years before.

Contents [hide]

Lawrence Livermore National Laboratory

Motto	"Science in the national interest"
Established	1952 by the University of California
Research Type	National security, nuclear science
Budget	US$1.6 billion
Director	George H. Miller
Staff	6,800
Location	Livermore, California
Campus	3.2 km² (800 acres)
Operating Agency	Lawrence Livermore National Security, LLC
Website	www.llnl.gov

Background [edit]

LLNL is self-described as "a premier research and development institution for science and technology applied to national security."[1] Its principal responsibility is ensuring the safety, security and reliability of the nation's nuclear weapons through the application of advanced science, engineering and technology. The Laboratory also applies its special expertise and multidisciplinary capabilities to preventing the proliferation and use of weapons of mass destruction, bolstering homeland security and solving other nationally important problems, including energy and environmental security, basic science and economic competitiveness.

LLNL is home to many unique facilities and a number of the most powerful computer systems in the world, according to the TOP500 list, including Blue Gene/L, the world's fastest computer from 2004 until Los Alamos National Laboratory's Roadrunner supercomputer surpassed it in 2008. The Lab is a leader in technical innovation: since 1978, LLNL has received a total of 118 prestigious R&D 100 Awards, including

Aerial view of Lawrence Livermore National Laboratory

navigation: Main page, Contents, Featured content, Current events, Random article

search: Go, Search

interaction: About Wikipedia, Community portal, Recent changes, Contact Wikipedia, Donate to Wikipedia, Help

toolbox: What links here, Related changes, Upload file, Special pages, Printable version, Permanent link, Cite this page

languages: Deutsch, فارسی, Français, Bahasa Indonesia, Italiano, 日本語, Norsk (bokmål), Polski, Русский, 中文

"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

Goal: machine-readable summaries

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

5 easy methods for relation extraction

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

5 easy methods for relation extraction

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

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 grep for new tuples

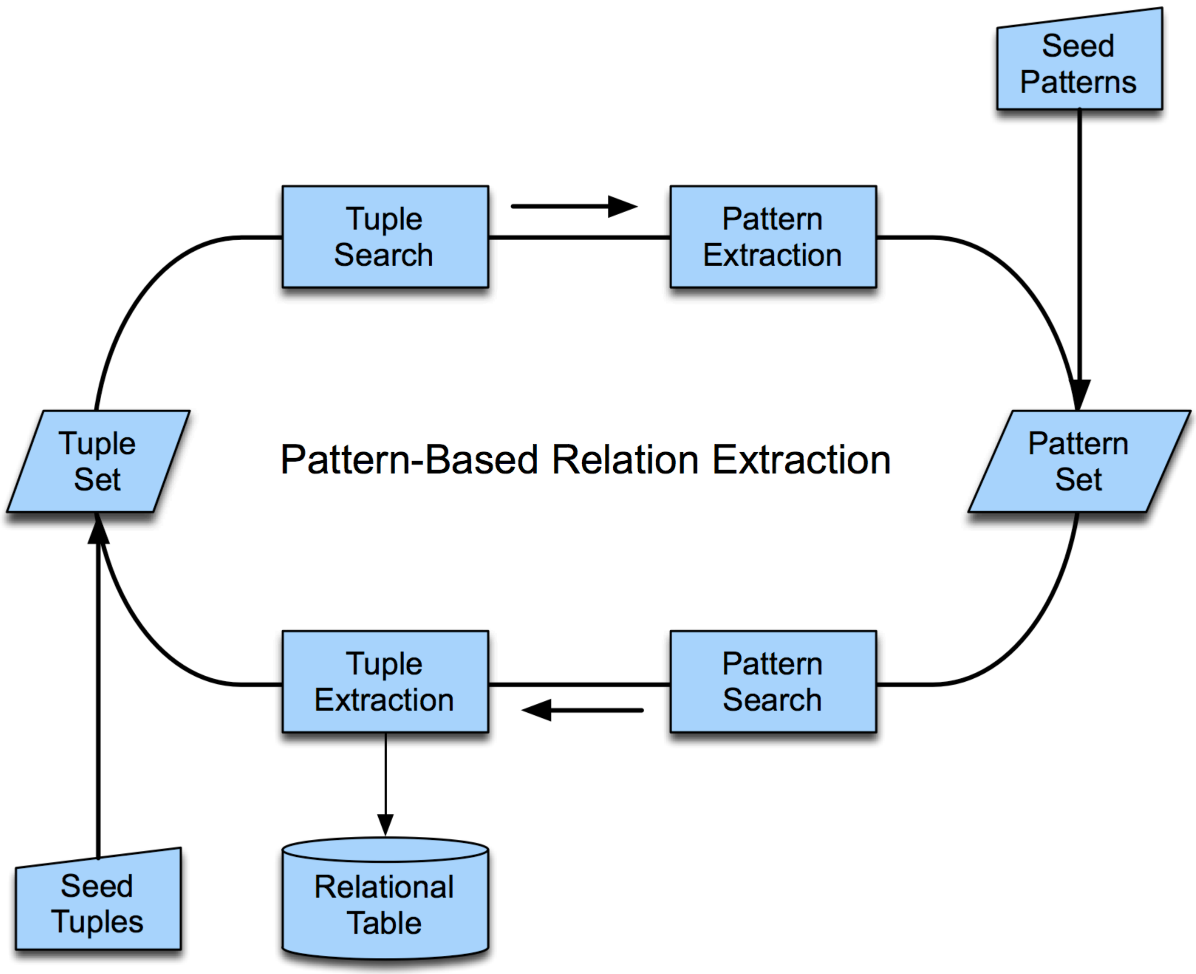
Slide adapted from Jim Martin

Bootstrapping à la Hearst

- Choose lexical relation *R*, e.g. hypernymy
- Gather a set of pairs that have this relation
- Find places in the corpus where these expressions occur near each other and record the environment
- Find the commonalities among these environments and hypothesize that common ones yield patterns that indicate the relation of interest

Shakespeare and other authors	X and other Ys
metals such as tin and lead	Ys such as X
such diseases as malaria	such Ys as X
regulators including the SEC	Ys including X

Bootstrapping relations



Slide adapted from Jim Martin

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:

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

- Now iterate, using these patterns to get more instances and 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


	ORGANIZATION	’s$_{0.4}$ headquarters$_{0.4}$ in$_{0.1}$	LOCATION	

	LOCATION	-$_{0.75}$ based$_{0.75}$	ORGANIZATION	

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
- Don't have a probabilistic interpretation
  - Hard to know how confident to be in each result

5 easy methods for relation extraction

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

KnowItAll (Etzioni et al. 2005)

- Input: target class labels and relation labels

Predicate	class label	relation label
City	"city", "town"	——
Country	"country", "nation"	——
capitalOf(City,Country)	——	"capital of"

- Use Hearst patterns to find instances of classes
  - ENTITY and/or other CLASS, such CLASS as ENTITY, etc
- Now use new pattern templates to find relations
  - CLASS1 is the RELATION CLASS2
  - CLASS1, RELATION CLASS2
- So once you learn Paris and Berlin are cities
  - Can use "Paris is the capital of France" to extract capitalOf(Paris, France)

KnowItAll PMI-based Assessor

- Validate candidate instances using “discriminators”
- Compare # search engine hits for
  - instance alone
  - Instance + discriminator

$$\mathrm{PMI}(I, D) = \frac{|\mathrm{Hits}(D + I)|}{|\mathrm{Hits}(I)|}$$

- (This is *not* the conventional definition of PMI!)
- Use “PMI” scores as features for Naïve Bayes classifier
- Example: linguists such as
  - PMI(linguists such as, Chomsky) = 4000 / 17.5M = 2.23 E-04
  - PMI(linguists such as, Potts) = 1 / 26.8M = 3.73 E-08

TextRunner (Banko et al. 2007)

1. **Self-Supervised Learner**: automatically labels +/– examples & learns an extractor
2. **Single-Pass Extractor:** single pass over corpus, identifying extractions in each sentence
3. **Redundancy-Based Assessor:** Assign a probability to each extraction

Step 1: Self-Supervised Learner

- Run a parser over 2000 sentences
  - expensive (0.5 seconds/parse) 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)
- Now label each tuple t as **positive** if and only if:
  - The dependency path between entities is short
  - The dependency path doesn't cross a clause boundary
  - Neither NP is a pronoun
- Now train a Naïve Bayes classifier to distinguish them
  - using features like POS tags nearby, stop words, etc. etc.

Step 2: Single-Pass Extractor

Over a huge (web-sized) corpus:

- Run a dumb POS tagger
- Run a dumb Base Noun Phrase finder
- 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 tuple to classifier
- Save only those predicted to be "trustworthy"

Step 3: Redundancy-Based Assessor

- Collect counts for each simplified relation

  ⟨*scientists*, *are studying*, *stars*⟩ → 17

- Given the counts for each relation, and the number of sentences, they use a combinatoric balls-and-urns model to compute probability of each relation [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 essentially the same as before.)

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

Results from TextRunner

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

Evaluating TextRunner

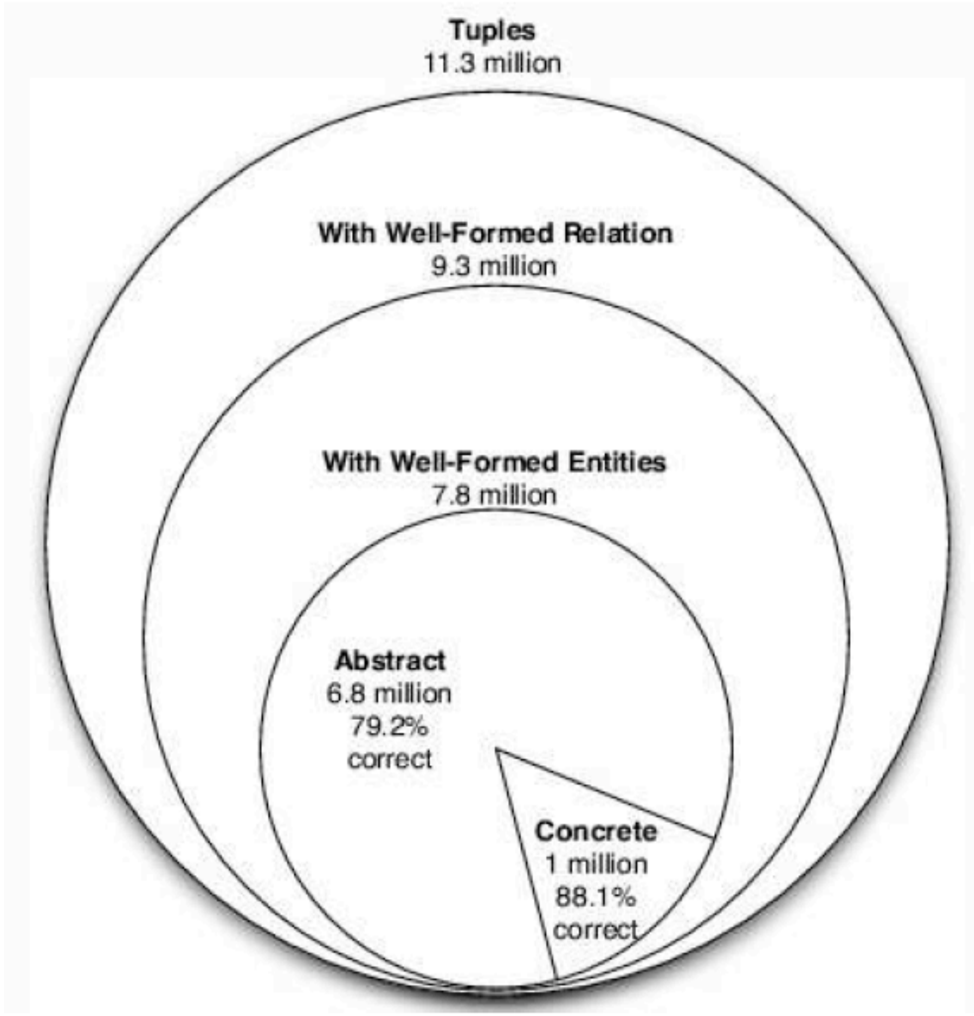


figure from Banko et al. 2007

5 easy methods for relation extraction

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

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 are known to belong to a certain relation, any sentence containing those two entities is likely to express that relation.
- Distant supervision: use *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)

Distant supervision paradigm

- For each pair of entities in a large database:
  - Grab sentences containing these entities from a corpus
  - Extract lots of noisy features from the sentences
    - Lexical features, syntactic features, named entity tags
  - Combine in a classifier

Distant supervision paradigm

- Has advantages of supervised approach:
  - use of rich of hand-created knowledge
  - relations have canonical names
  - can use rich features (e.g. syntactic features)
- Has advantages of unsupervised approach:
  - unlimited amounts of data
  - allows for very large number of weak features
  - 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 an IS-A pair according to 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*..."

Slide adapted from Rion Snow

Hypernyms via distant supervision

We construct a noisy training set consisting of occurrences from our corpus that contain an IS-A pair according to 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..."

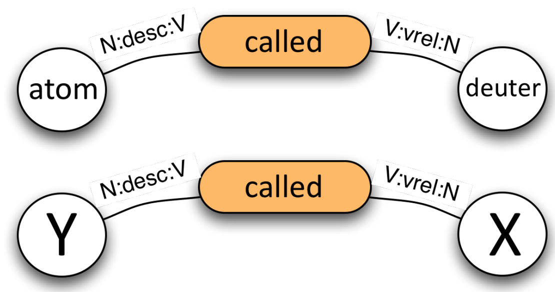
Training set (TREC and Wikipedia):
14,000 hypernym pairs, ~600,000 total pairs

Slide adapted from Rion Snow

Learning patterns

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

1. Take corpus sentences … doubly heavy hydrogen **atom called deuterium…**
2. Collect noun pairs (atom, deuterium)
   - 752,311 pairs from 6M words of newswire
3. Is pair an IS-A in WordNet? YES
   - 14,387 yes, 737,924 no
4. Parse the sentences
5. Extract patterns
   - 69,592 dependency paths >5 pairs)
6. Train classifier on these patterns
   - **Logistic regression** with 70K features (actually converted to 974,288 bucketed binary features)



One of 70,000 patterns

“<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:

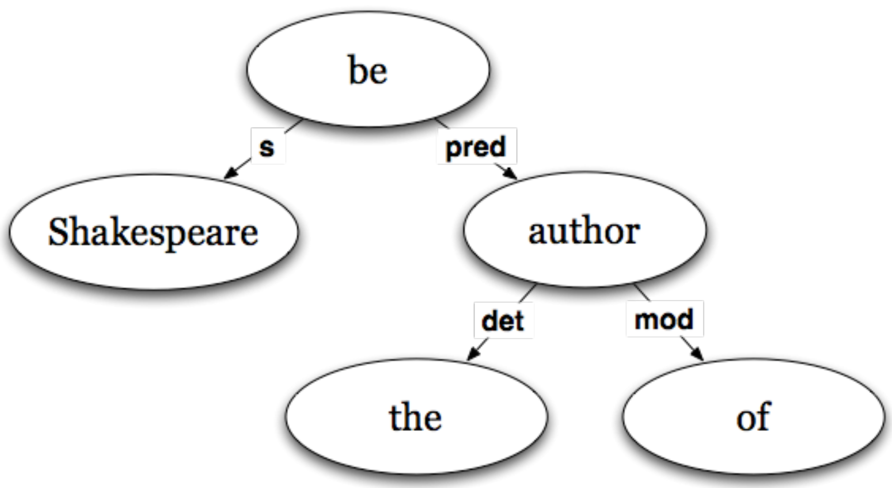
“*efflorescence / condition*”: …and a **condition called efflorescence** are other reasons for…
“*’neal_inc / company*” …The **company, now called O'Neal Inc.**, was sole distributor of E-Ferol…
“*hat_creek_outfit / ranch*” …run a small **ranch called the Hat Creek Outfit**.
“*hiv-1 / aids_virus*” …infected by the **AIDS virus, called HIV-1**.
“*bateau_mouche / attraction*” …local sightseeing **attraction called the Bateau Mouche**...
“*kibbutz_malkiyya / collective_farm*” …an Israeli **collective farm called Kibbutz Malkiyya**…

Recording the Lexico-Syntactic Environment with Syntactic Dependency Paths

MINIPAR: A principle-based dependency parser (Lin, 1998)

Example Word Pair: "**Shakespeare / author**"
Example Sentence: "Shakespeare was the author of several plays..."

Minipar Parse:



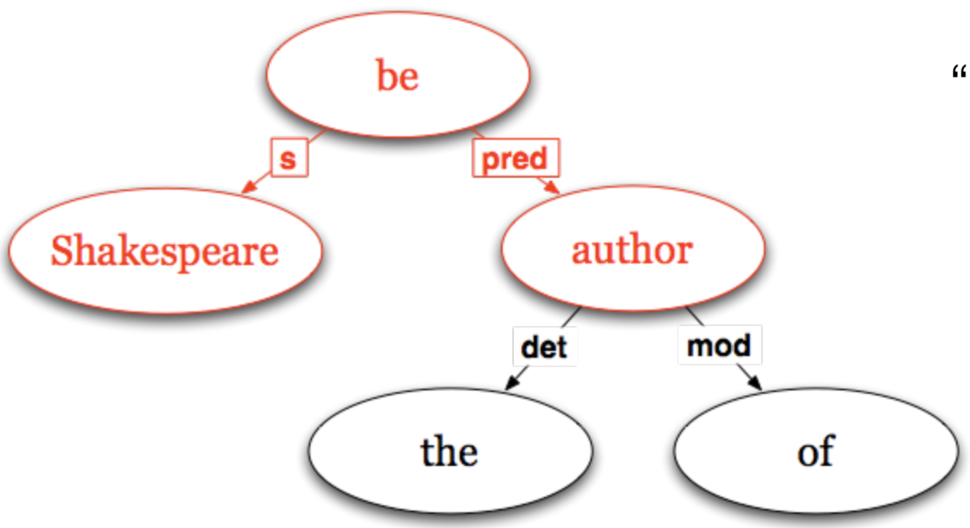
Slide from Rion Snow

Recording the Lexico-Syntactic Environment with Syntactic Dependency Paths

MINIPAR: A principle-based dependency parser (Lin, 1998)

Example Word Pair: “**Shakespeare / author**”
Example Sentence: “Shakespeare was the author of several plays...”

Minipar Parse:

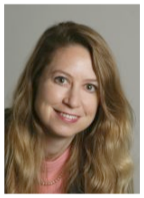


Extract shortest path:

“-N:s:VBE, “be”, VBE:pred:N”

Slide from Rion Snow

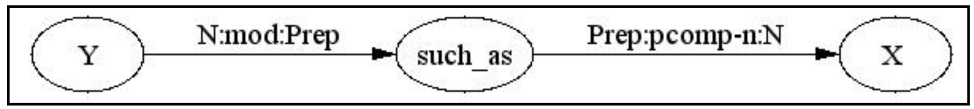
Hearst patterns to MINIPAR dependency paths



Hearst Pattern

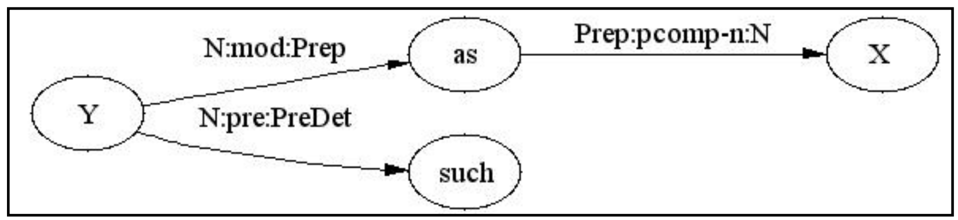
MINIPAR Representation

Y such as X…



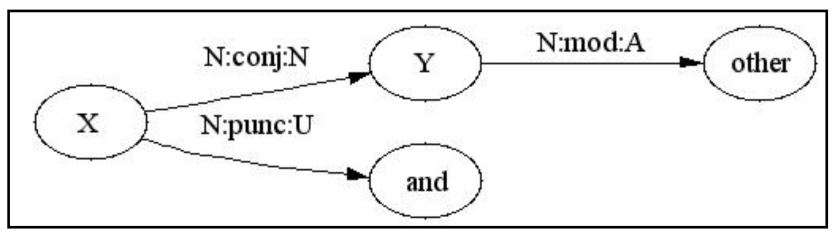
-N:pcomp-n:Prep,such_as,such_as,-Prep:mod:N

Such Y as X…



-N:pcomp-n:Prep,as,as,-Prep:mod:N,(such,PreDet:pre:N)}

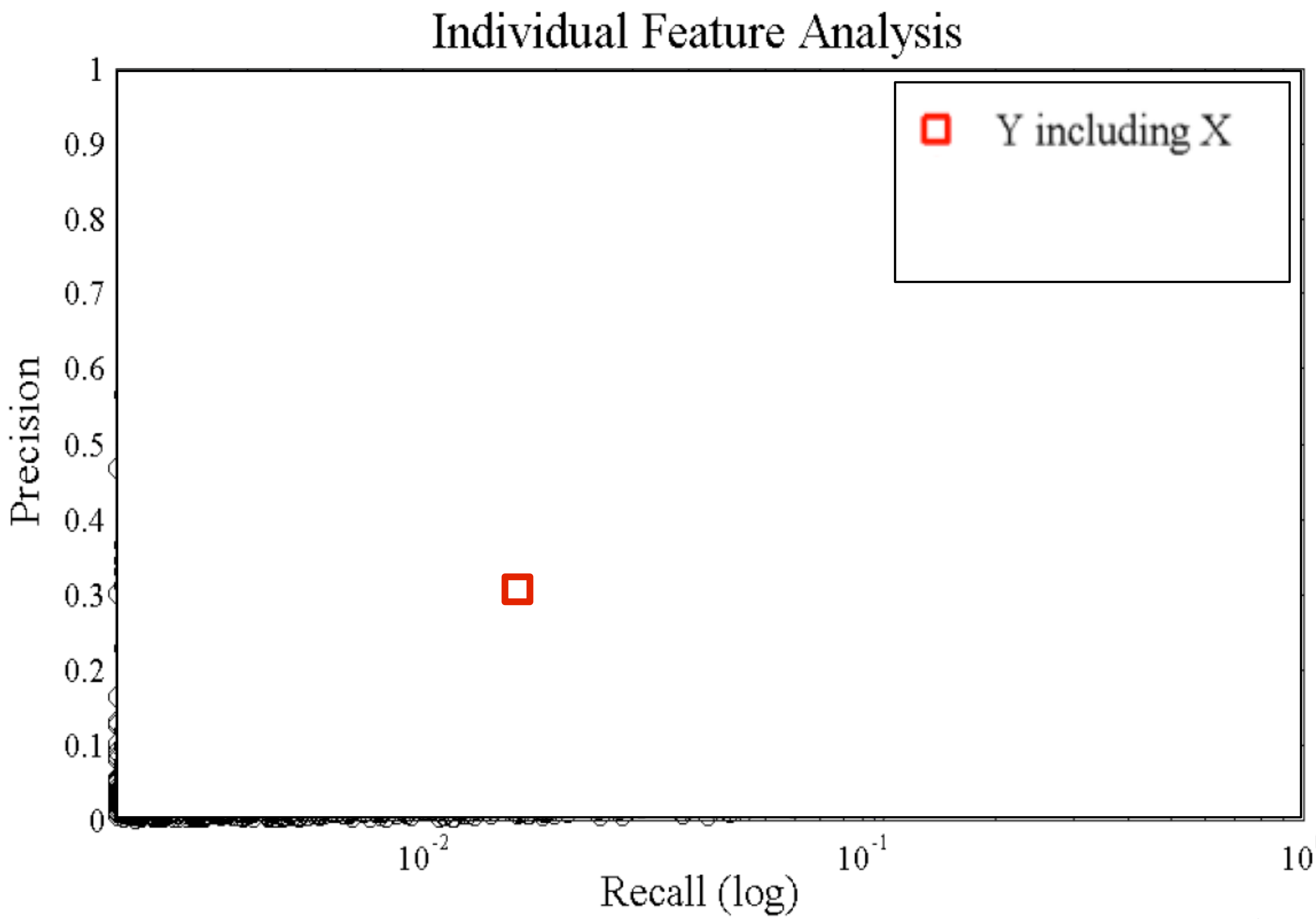
X… and other Y



(and,U:punc:N),N:conj:N, (other,A:mod:N)

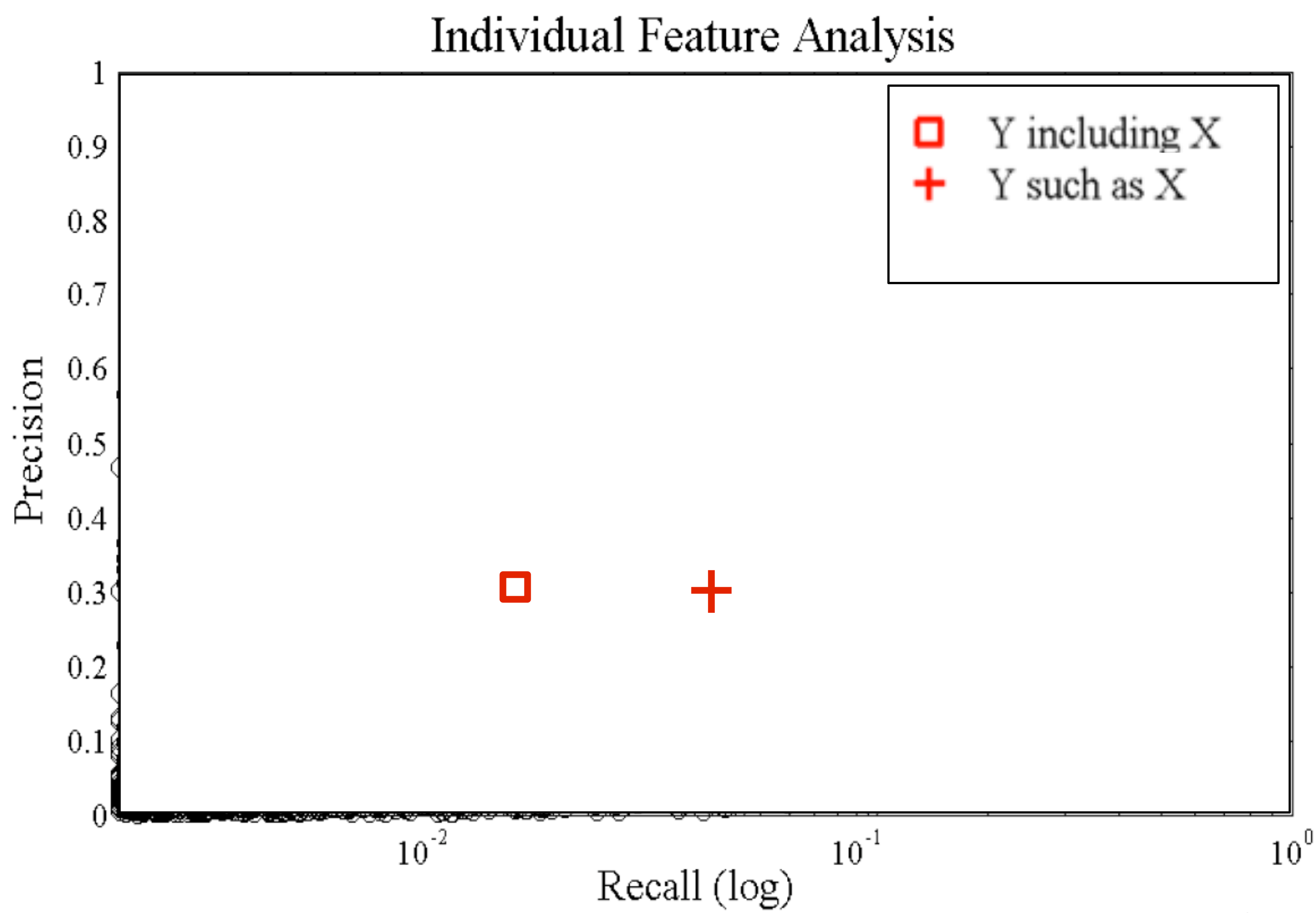
Slide from Rion Snow

Hypernym Precision / Recall for all Features



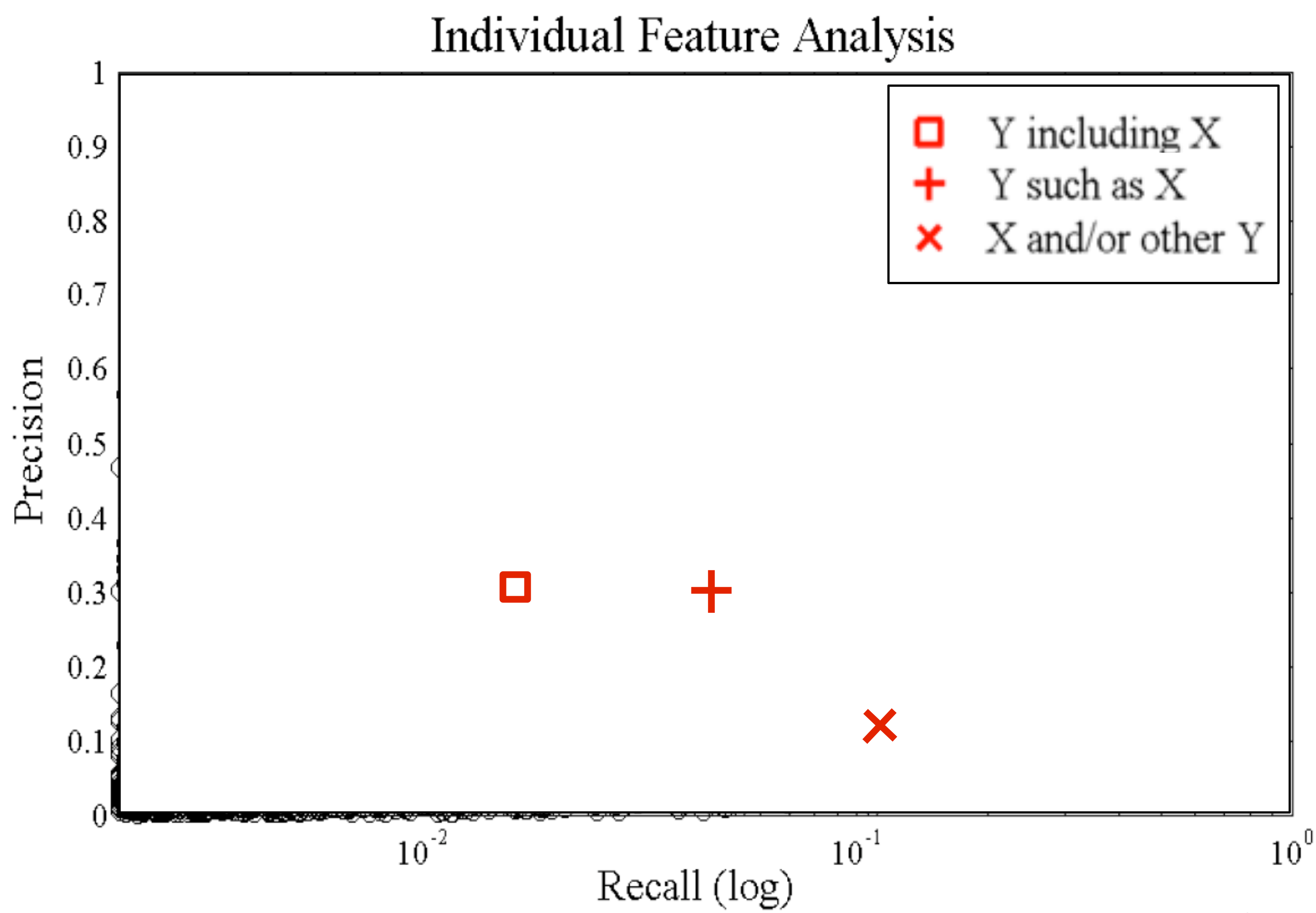
Slide from Rion Snow

Precision/recall for various patterns



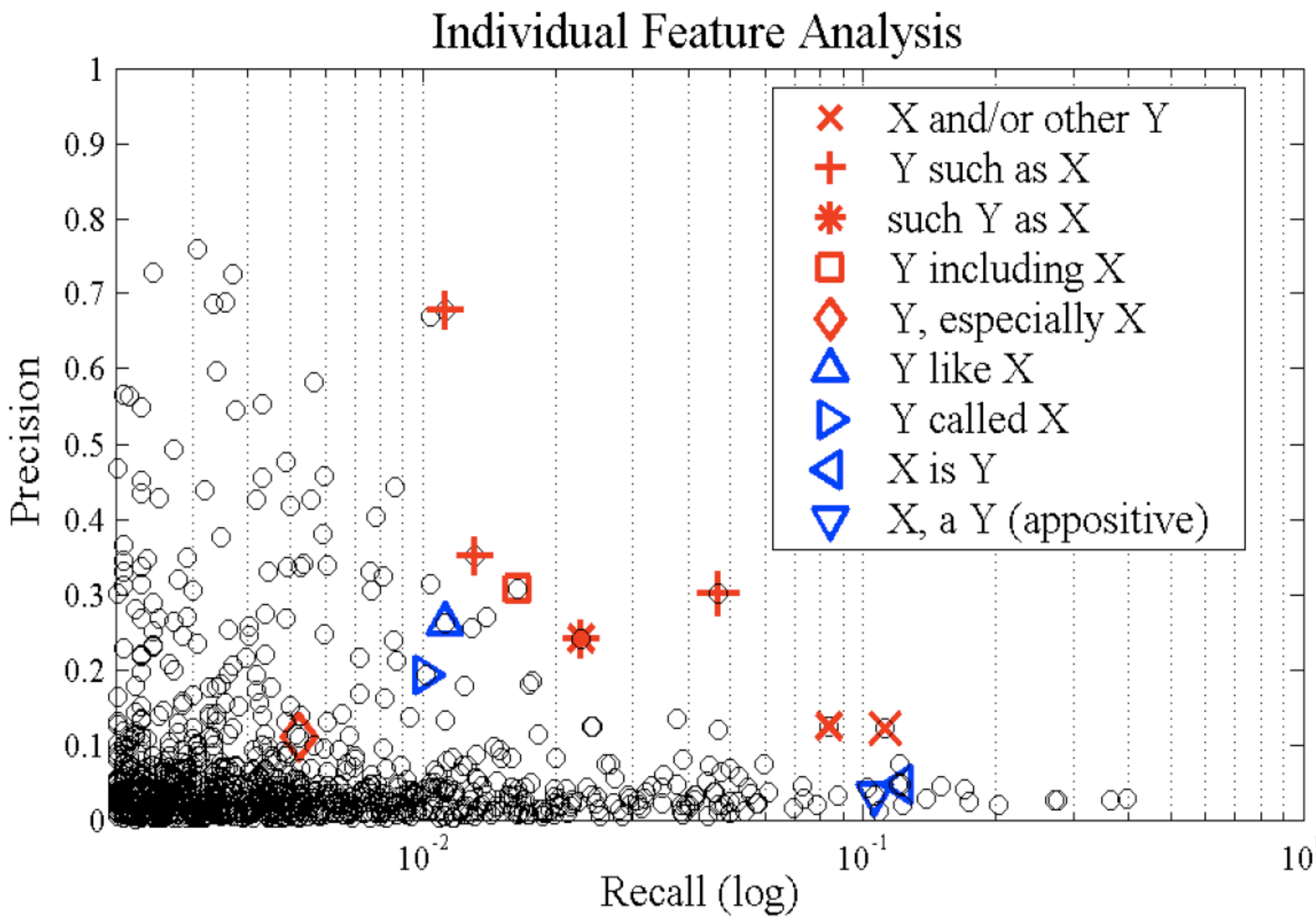
Slide from Rion Snow

Precision/recall for various patterns



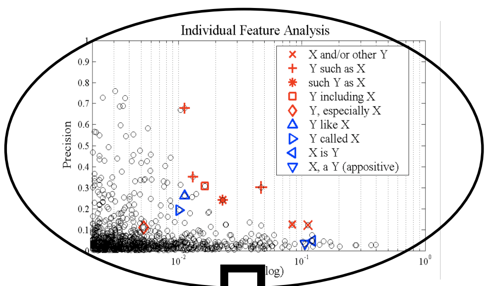
Slide from Rion Snow

Precision/recall for various patterns



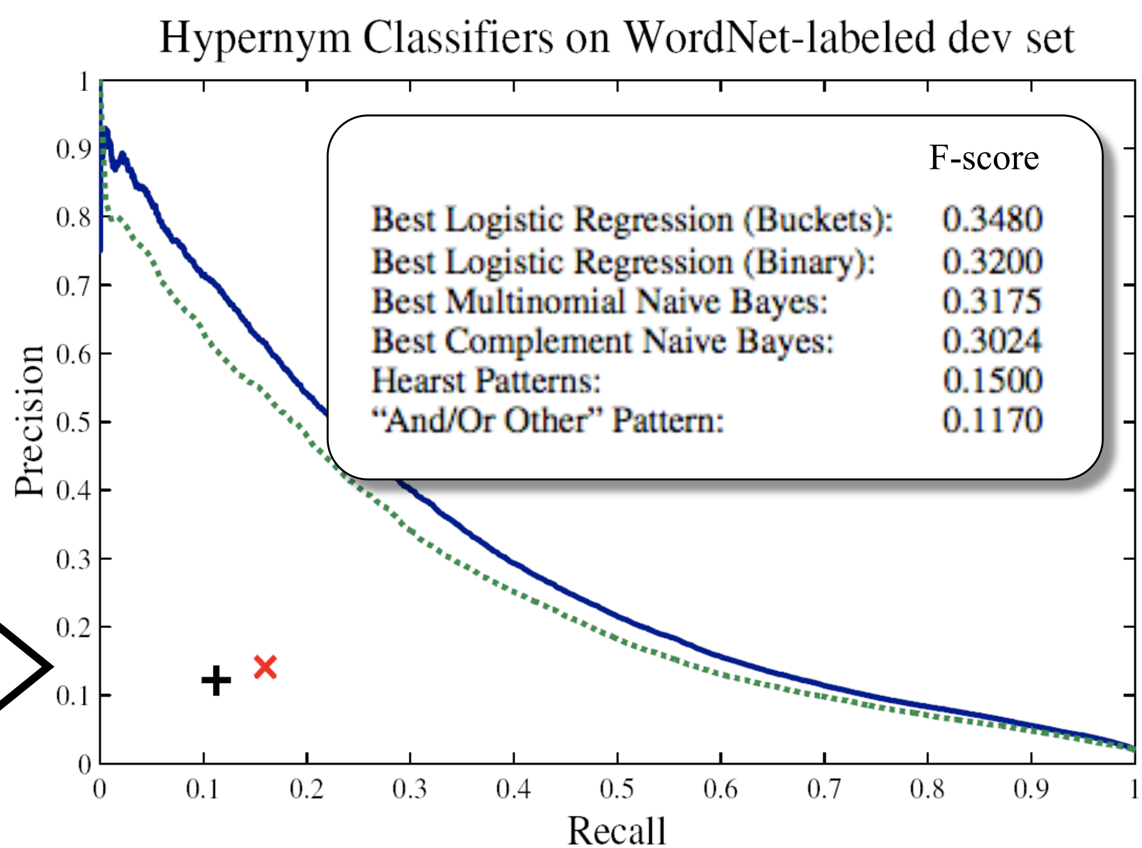
Slide from Rion Snow

Precision/recall for 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 from Rion Snow

What about other relations?

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

Training Set

Freebase

102 relations
940,000 entities
1.8 million instances

Corpus

Wikipedia

1.8 million articles
25.7 million sentences

Slide adapted from Rion Snow

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

Training

Text

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

Freebase relations

Founder:	<Bill Gates, Microsoft>
Founder:	<Larry Page, Google>
CollegeAttended:	<Bill Gates, Harvard>

Extracted training data

Training

Text

<u>Bill Gates</u> founded <u>Microsoft</u> in 1975.
Bill Gates, founder of Microsoft, …
Bill Gates attended Harvard from…
Google was founded by Larry Page and…

Freebase relations

Founder: <**<u>Bill Gates</u>**, **<u>Microsoft</u>**>
Founder: <Larry Page, Google>
CollegeAttended: <Bill Gates, Harvard>

Extracted training data

[Founder]

- "X founded Y"

Training

Text

Bill Gates founded Microsoft in 1975.
<u>Bill Gates</u>, founder of <u>Microsoft</u>, …
Bill Gates attended Harvard from…
Google was founded by Larry Page and…

Freebase relations

Founder: <**<u>Bill Gates</u>**, **<u>Microsoft</u>**>
Founder: <Larry Page, Google>
CollegeAttended: <Bill Gates, Harvard>

Extracted training data

[Founder]

- "X founded Y"
- "X, founder of Y"

Training

Text

Bill Gates founded Microsoft in 1975.
Bill Gates, founder of Microsoft, …
<u>Bill Gates</u> attended <u>Harvard</u> from…
Google was founded by Larry Page and…

Freebase relations

Founder: <Bill Gates, Microsoft>
Founder: <Larry Page, Google>
CollegeAttended: <**<u>Bill Gates</u>**, **<u>Harvard</u>**>

Extracted training data

[Founder]

- "X founded Y"
- "X, founder of Y"

[CollegeAttended]

- "X attended Y"

Training

Text

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

Freebase relations

Founder: <Bill Gates, Microsoft>
Founder: <**Larry Page**, **Google**>
CollegeAttended: <Bill Gates, Harvard>

Extracted training data

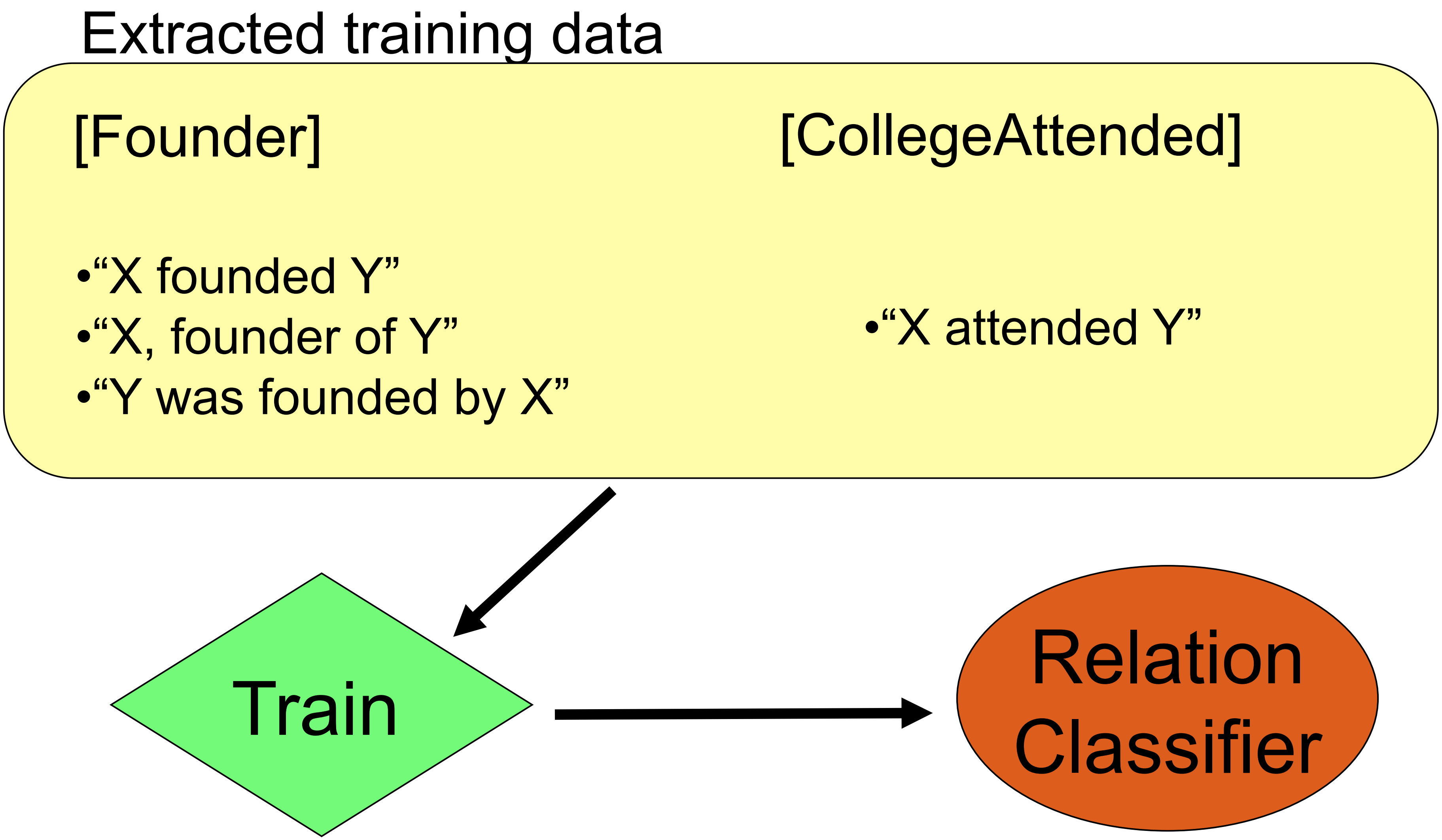
[Founder]

- "X founded Y"
- "X, founder of Y"
- "Y was founded by X"

[CollegeAttended]

- "X attended Y"

Training


Extracted training data
[Founder]
•"X founded Y"
•"X, founder of Y"
•"Y was founded by X"
[CollegeAttended]
•"X attended Y"
Train
Relation Classifier

Testing

Corpus text

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

Extracted testing data

Testing

Corpus text

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

Extracted testing data

<Henry Ford, Ford Motor Co.>

[???]

- "X founded Y"

Testing

Corpus text

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

Extracted testing data

<Henry Ford, Ford Motor Co.>

[???]

- "X founded Y"
- "Y was founded by X"

Testing

Corpus text

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

Extracted testing data

<Henry Ford, Ford Motor Co.>

[???]

- “X founded Y”
- “Y was founded by X”

<Steve Jobs, Reed College>

[???]

- “X attended Y”

Testing

Extracted testing data

<Henry Ford, Ford Motor Co.>

[???]

- "X founded Y"
- "Y was founded by X"

<Steve Jobs, Reed College>

[???]

- "X attended Y"

Trained Relation Classifier

Results!

<Henry Ford, Ford Motor Co.>: Founder

<Steve Jobs, Reed College>: CollegeAtt.

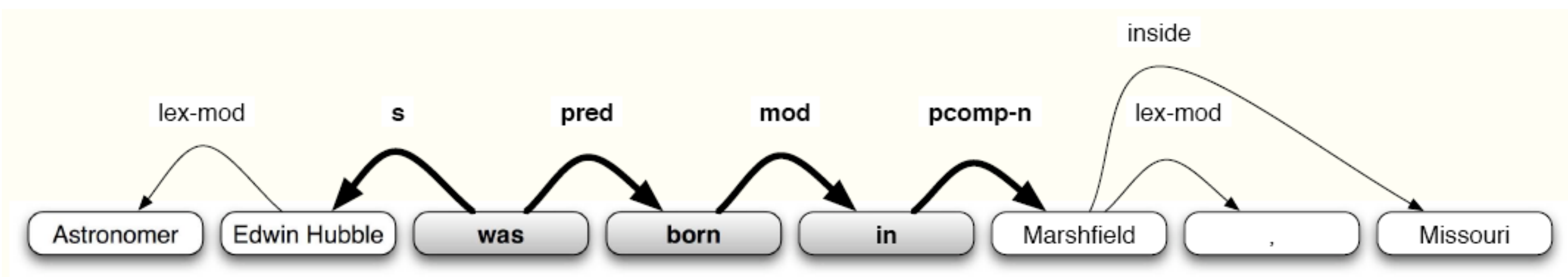
Advantage

- ACE paradigm: labeling sentences
- Our 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

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]



Examples of 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 {person, location, organization, miscellaneous, none} (Finkel et al. 2005)
- 3 configurations: lexical features, syntax features, both

Experiment

- 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
- We only extract relation instances that are not already in Freebase

Newly discovered relation instances

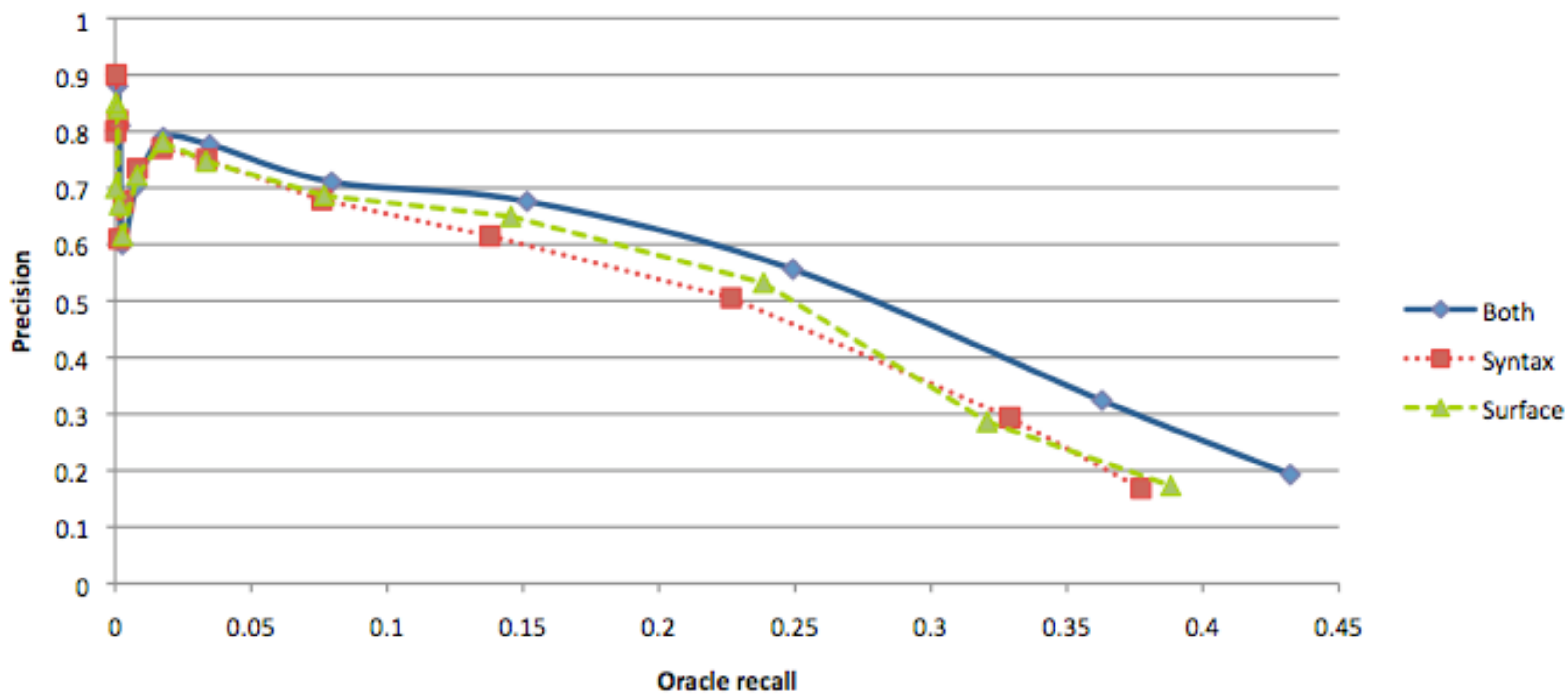
Relation name	New instance
/location/location/contains	Paris, Montmartre
/location/location/contains	Ontario, Fort Erie
/music/artist/origin	Mighty Wagon, Cincinnati
/people/deceased_person/place_of_death	Fyodor Kamensky, Clearwater
/people/person/nationality	Marianne Yvonne Heemskerk, Netherlands
/people/person/place_of_birth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

Ten relation instances extracted by the system that did not appear in Freebase

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 10 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

Human evaluation

- 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

Where syntax helps

Back Street *is a 1932 film made by*
Universal Pictures, directed by ***John M. Stahl****,*
and produced by Carl Laemmle Jr.

Back Street and **John M. Stahl** are far apart in surface string

But close together in dependency parse

Where syntax doesn't help

Beaverton *is a city in* ***Washington County****, Oregon …*

Beaverton and **Washington County** are close together in surface string

Conclusions

- Distant supervision extracts high-precision patterns for a variety of relations.
- Can make use of 1000 times 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.

Discussion

- Relation extraction → learning by reading
- Suppose we could do relation extraction perfectly?
- What would we still be missing?
- What knowledge could we still not gather from the web?