Learning to Extract Entities from Labeled and Unlabeled Text

Rosie Jones

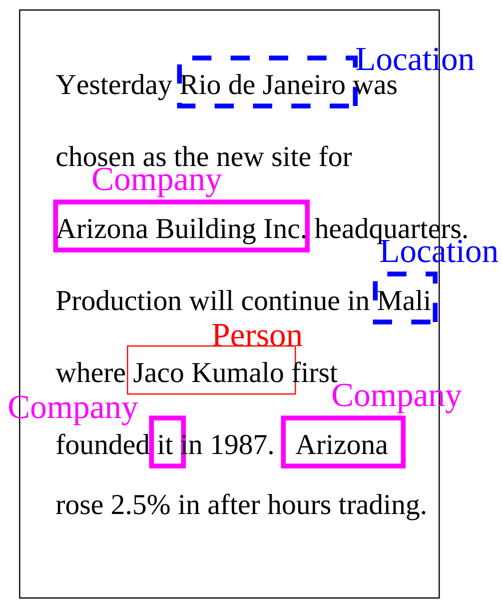
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

May 5th, 2005

Extracting Information from Text

Yesterday Rio de Janeiro was chosen as the new site for Arizona Building Inc. headquarters. Production will continue in Mali where Jaco Kumalo first founded it in 1987. Arizona rose 2.5% in after hours trading.

Extracting Information from Text



Information Extraction

- Set of rules for extracting words or phrases from sentences

  extract(X) if $p(location|X, context(X)) > \tau$

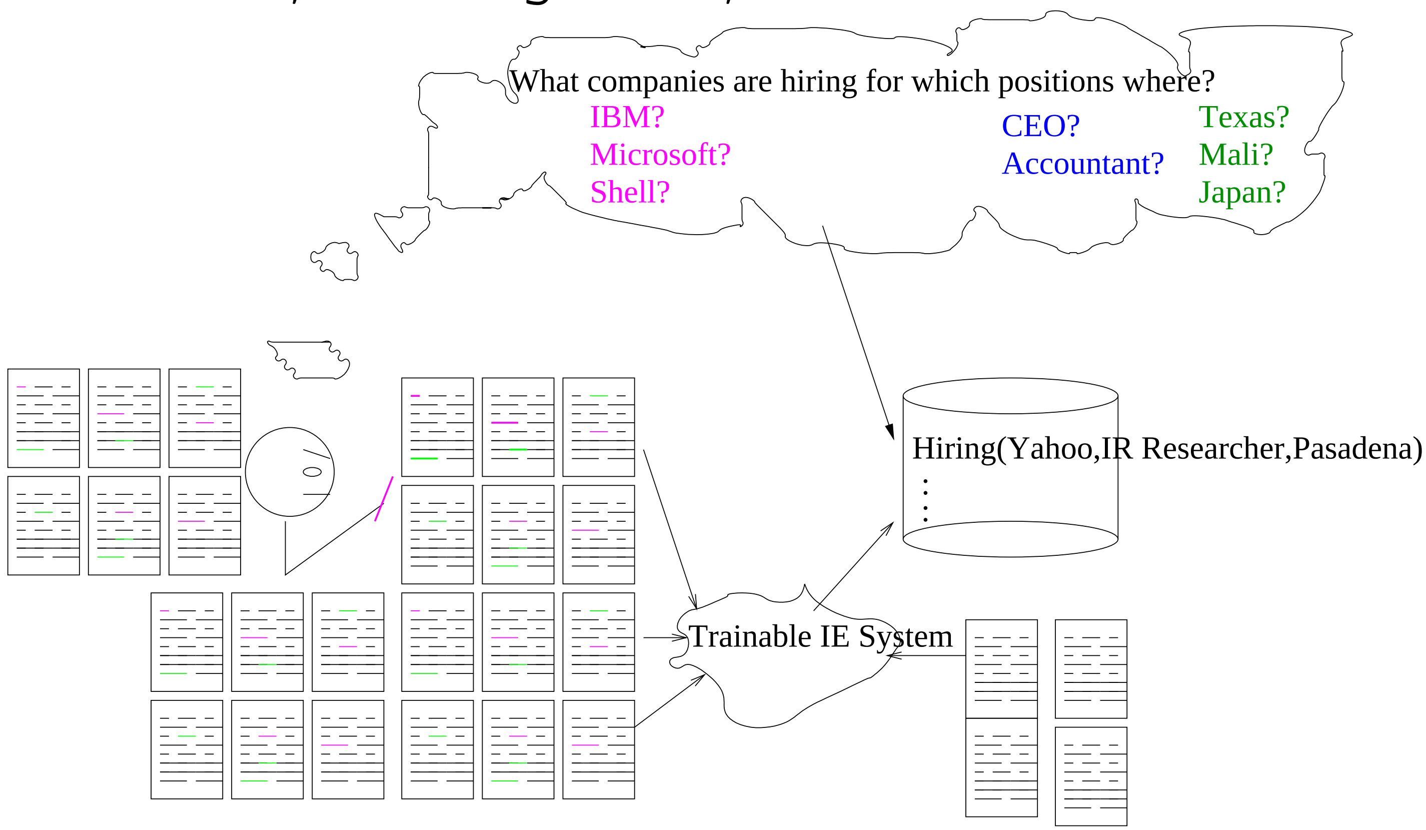
  - “hotel in paris”: X=”paris”, context(X) = “hotel in”
  - “paris hilton”: X = “paris”, “context(X) = “hilton”
  - $p_{location}$(“paris”) = 0.5
  - $p_{location}$(“hilton”) = 0.01
  - $p_{location}$(“hotel in”) = 0.9

Information Extraction II

- Types of Information:
  - “Locations”
  - “Organizations”
  - “People”
  - “Products”
  - “Job titles”
  - ...

Costs of Information Extraction

Data Collection, Labeling Time, Information Verification



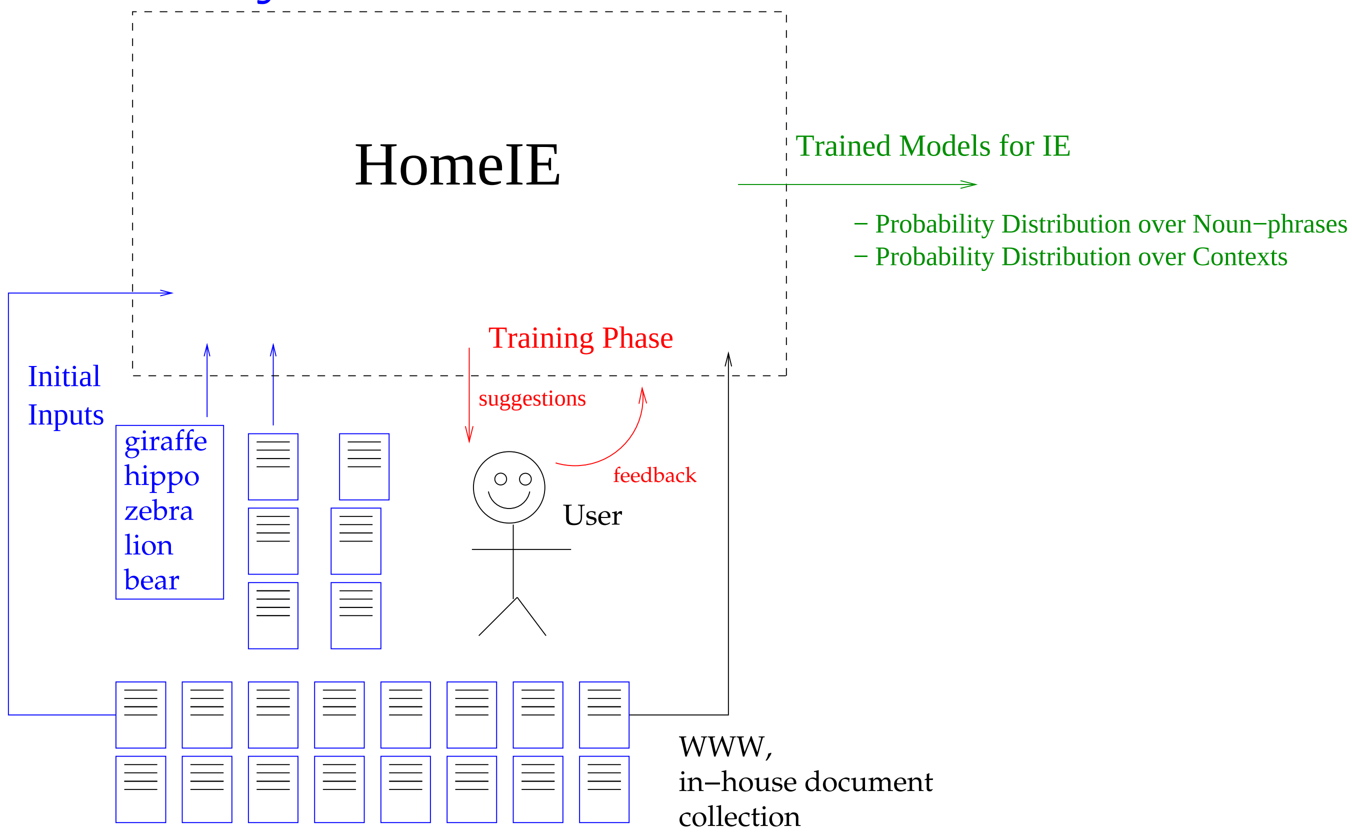
Costs of Information Extraction

- 3 - 6 months to port to new domain [Cardie 98]

- 20,000 words required to learn named entity extraction [Seymore et al 99]

- 7000 labeled examples: supervised learning of extraction rules for MUC task [Soderland 99]

Automated IE System Construction



Thesis Statement

We can train semantic class extractors from text using minimal supervision in the form of

- seed examples

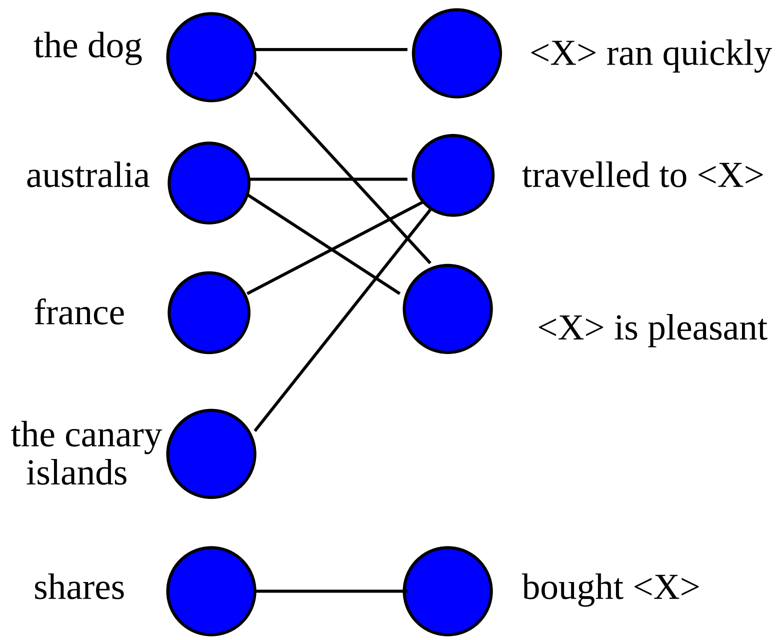
- actively labeled examples

by exploiting the graph structure of text cooccurrence relationships.

Talk Outline

- Information Extraction
- **Data Representation**
- Bootstrapping Algorithms: Learning From Almost Nothing
- Understanding the Data: Graph Properties
- Active learning: Effective Use of User Time

Data Representation



noun-phrases	lexico-syntactic contexts
the dog	X ran quickly
the dog	X is pleasant
australia	X is pleasant
shares	bought X
australia	travelled to X
france	travelled to X
the canary islands	travelled to X

Information Extraction Approaches

- Hand-constructed
- Supervised learning from many labeled examples
- Semi-supervised learning

The Semi-supervised IE Learning Task

Given:

- A large collection of unlabeled documents

- A small set (10) of nouns representing the target class

Learn:
A set of rules for extracting members of the target class from novel unseen documents (test collection)

Initialization from Seeds

- foreach instance in unlabeled docs
  - if matchesSeed(noun-phrase)
  - hardlabel(instance) = 1
  - else softlabel(instance) = 0
- hardlabel(australia, located-in) = 1
- softlabel(the canary-islands, located-in) = 0

Bootstrapping Approach to Semi-supervised Learning

- learn two models:
  - noun-phrases: {New York, Timbuktu, China, the place we met last time, the nation's capitol ...}
  - contexts: {located-in <X>, travelled to <X>...}
- Use redundancy in two models:
  - noun-phrases can label contexts
  - contexts can label noun-phrases

$\Rightarrow$ bootstrapping

Space of Bootstrapping Algorithms

- Incremental (label one-at-a-time) / **All at once**
  [Cotraining: Blum & Mitchell, 1998]
  [coEM: Nigam & Ghani, 2000]

- asymmetric/**symmetric**

- heuristic/**probabilistic**

- **use knowledge about language** /assume nothing about language

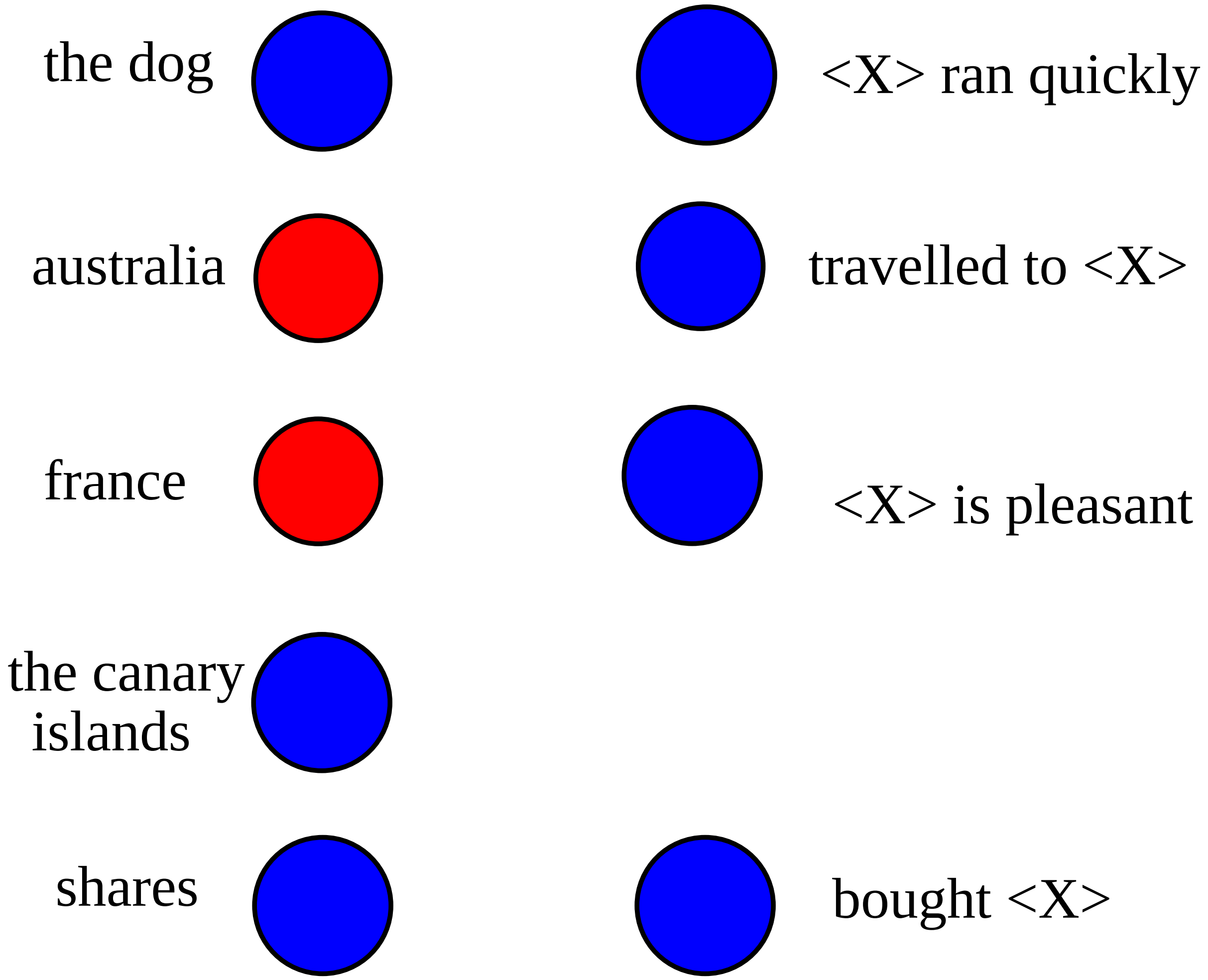
Bootstrapping Inputs

- corpus
  - 4160 company web pages
  - parsed [Riloff 1996] into noun-phrases and contexts (around 200,000 instances)
    - "Ultramar Diamond Shamrock has a strong network of approximately 4,400 locations in 10 Southwestern states and eastern Canada."
    - Ultramar Diamond Shamrock - <X> has network
    - 10 Southwestern states and eastern Canada - locations in <X>

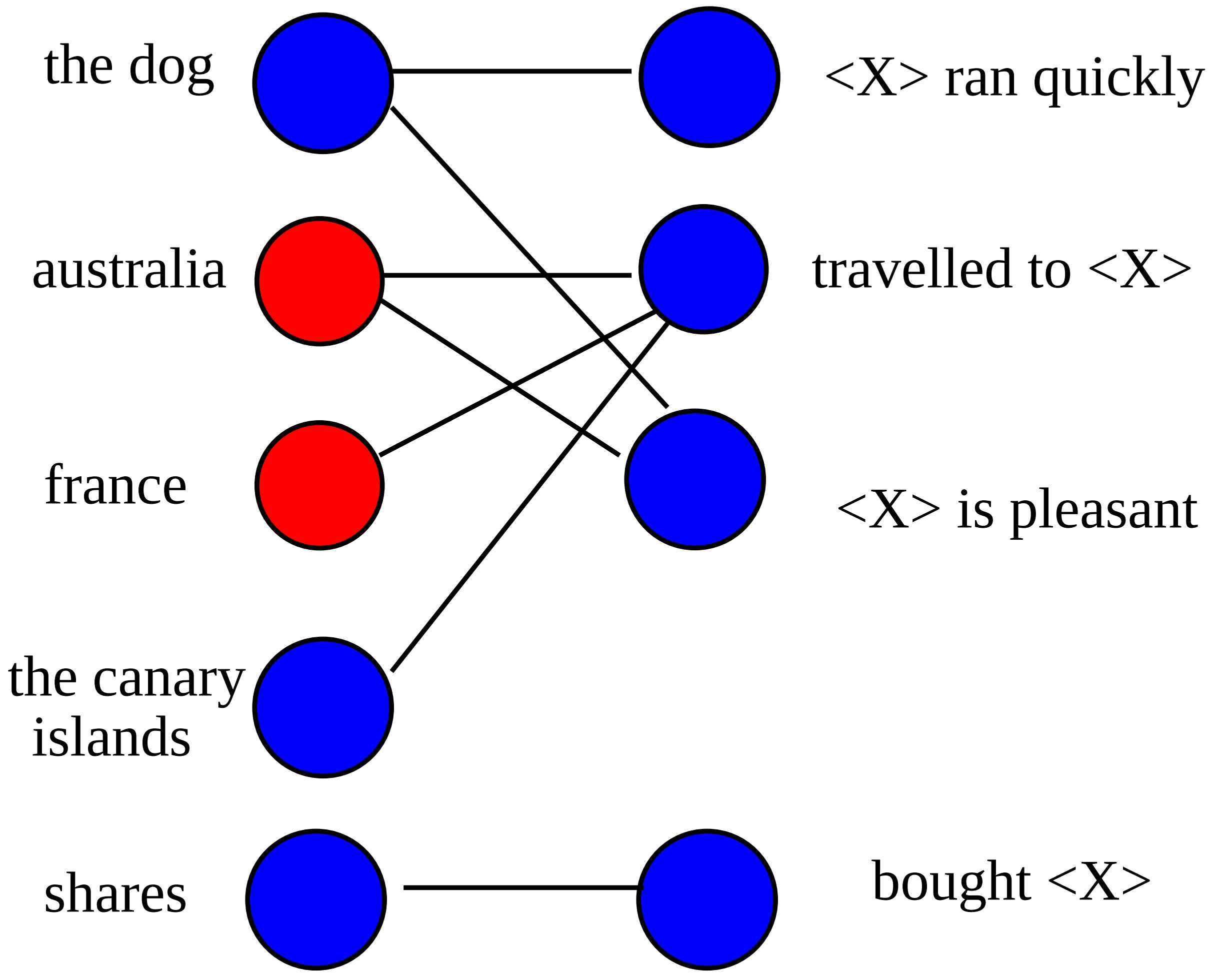
Seeds

- locations : {australia, canada, china, england, france, germany, japan, mexico, switzerland, united states }
- people : {customers, subscriber, people, users, shareholders, individuals, clients, leader, director, customer }
- organizations: {inc., praxair, company, companies, dataram, halter marine group, xerox, arco, rayonier timberlands, puretec}

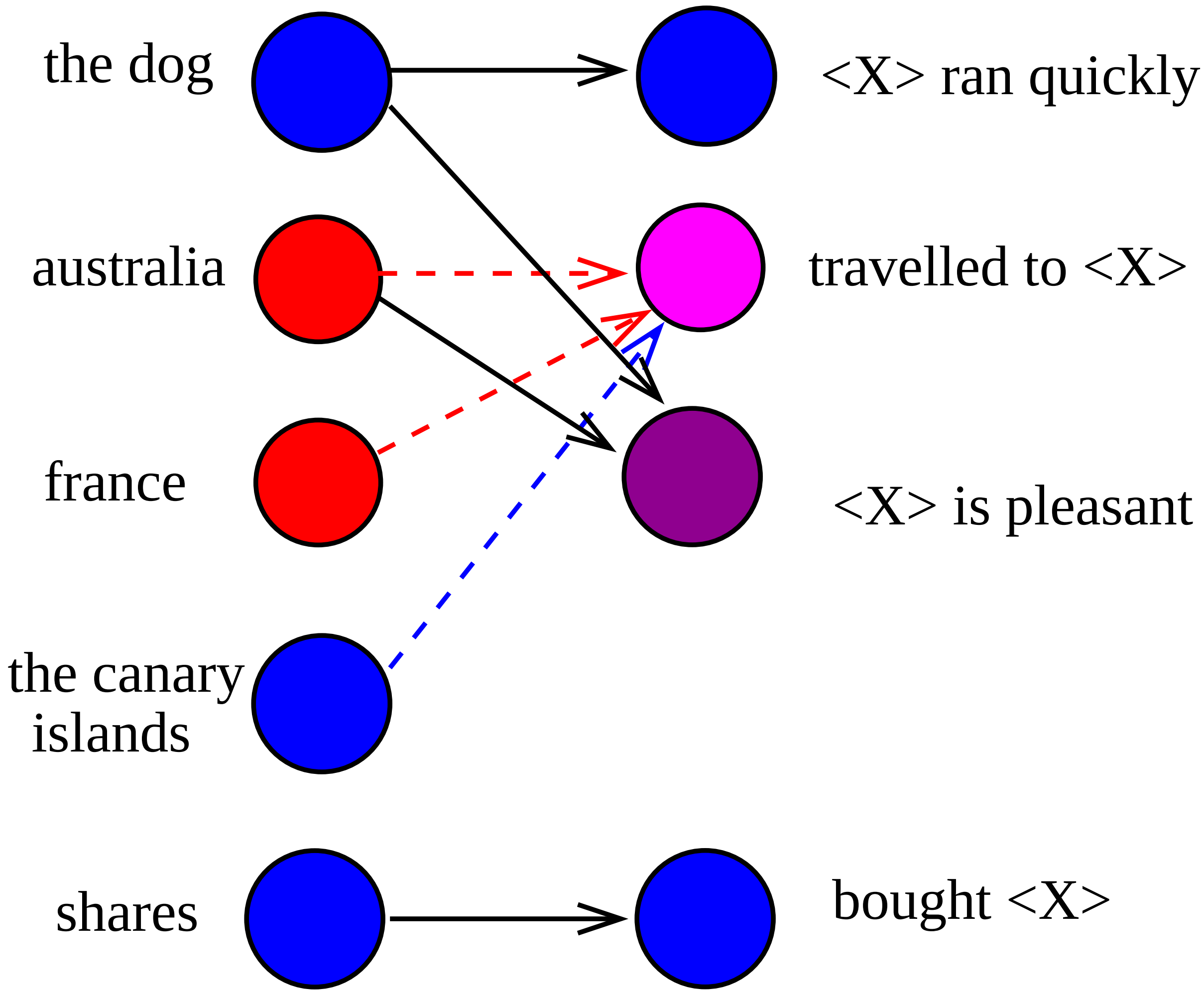
CoEM for Information Extraction



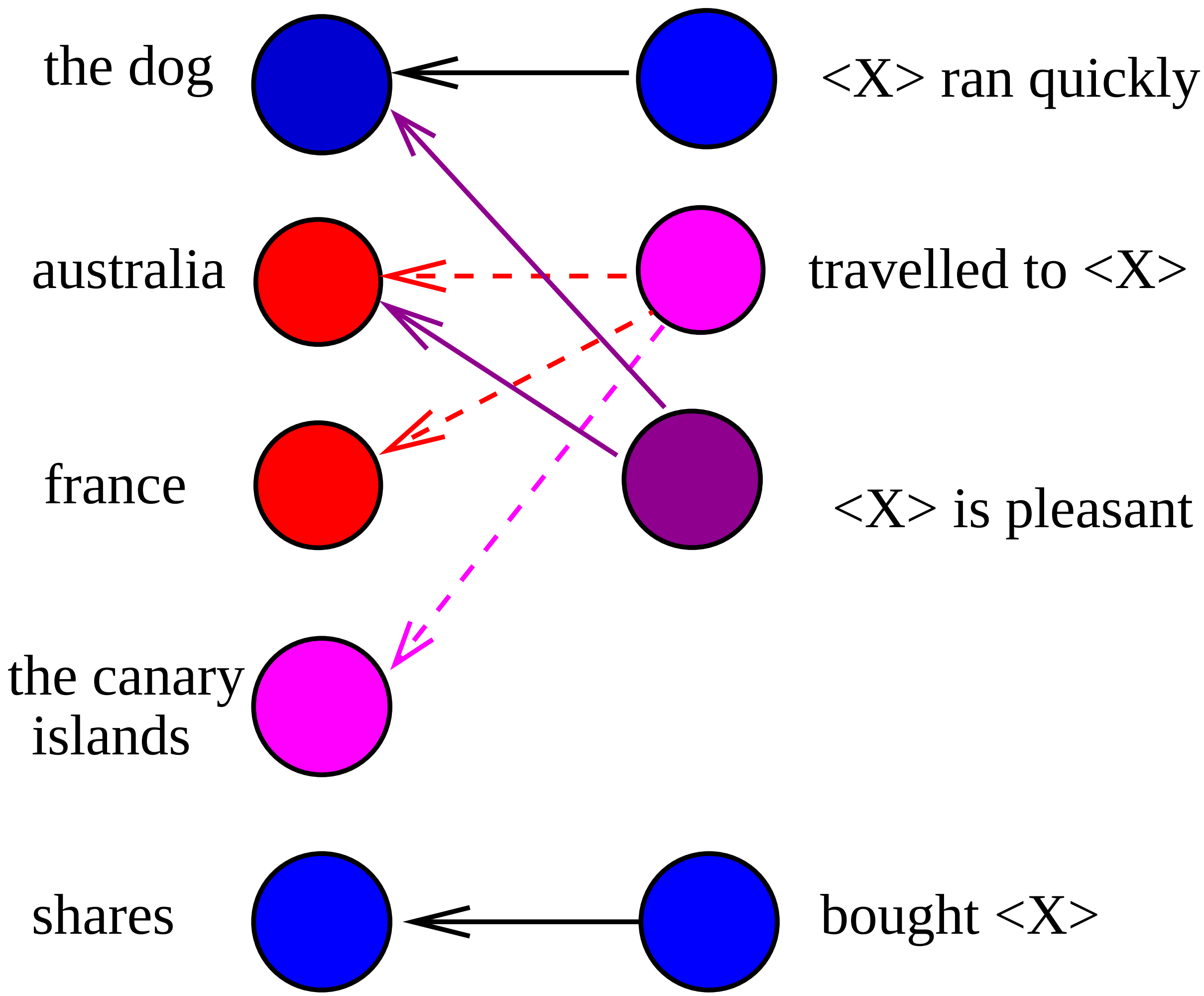
CoEM for Information Extraction



CoEM for Information Extraction



CoEM



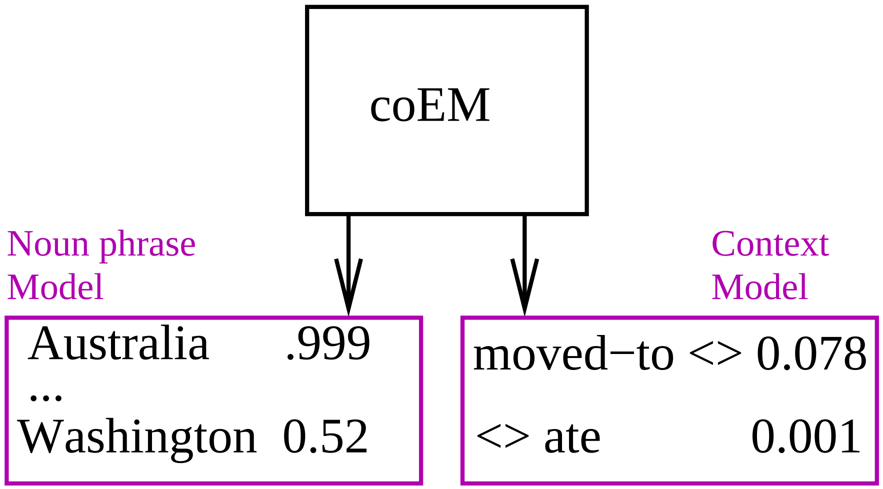
the dog
<X> ran quickly
australia
travelled to <X>
france
<X> is pleasant
the canary islands
shares
bought <X>

coEM Update Rules

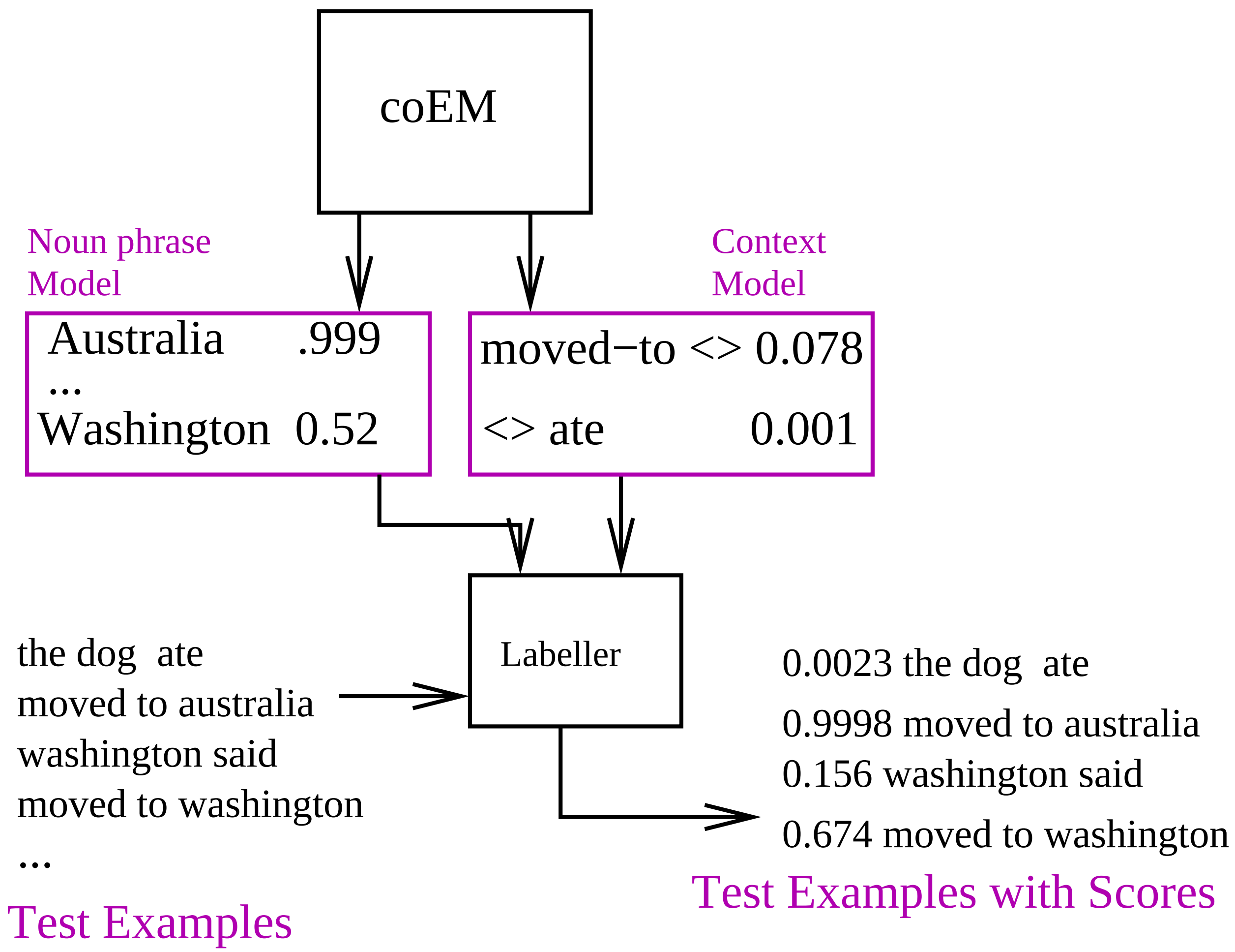
$$P(class|context_i) = \sum_j P(class|NP_j)P(NP_j|context_i) \quad (1)$$

$$P(class|NP_i) = \sum_j P(class|context_j)P(context_j|NP_i) \quad (2)$$

Evaluation

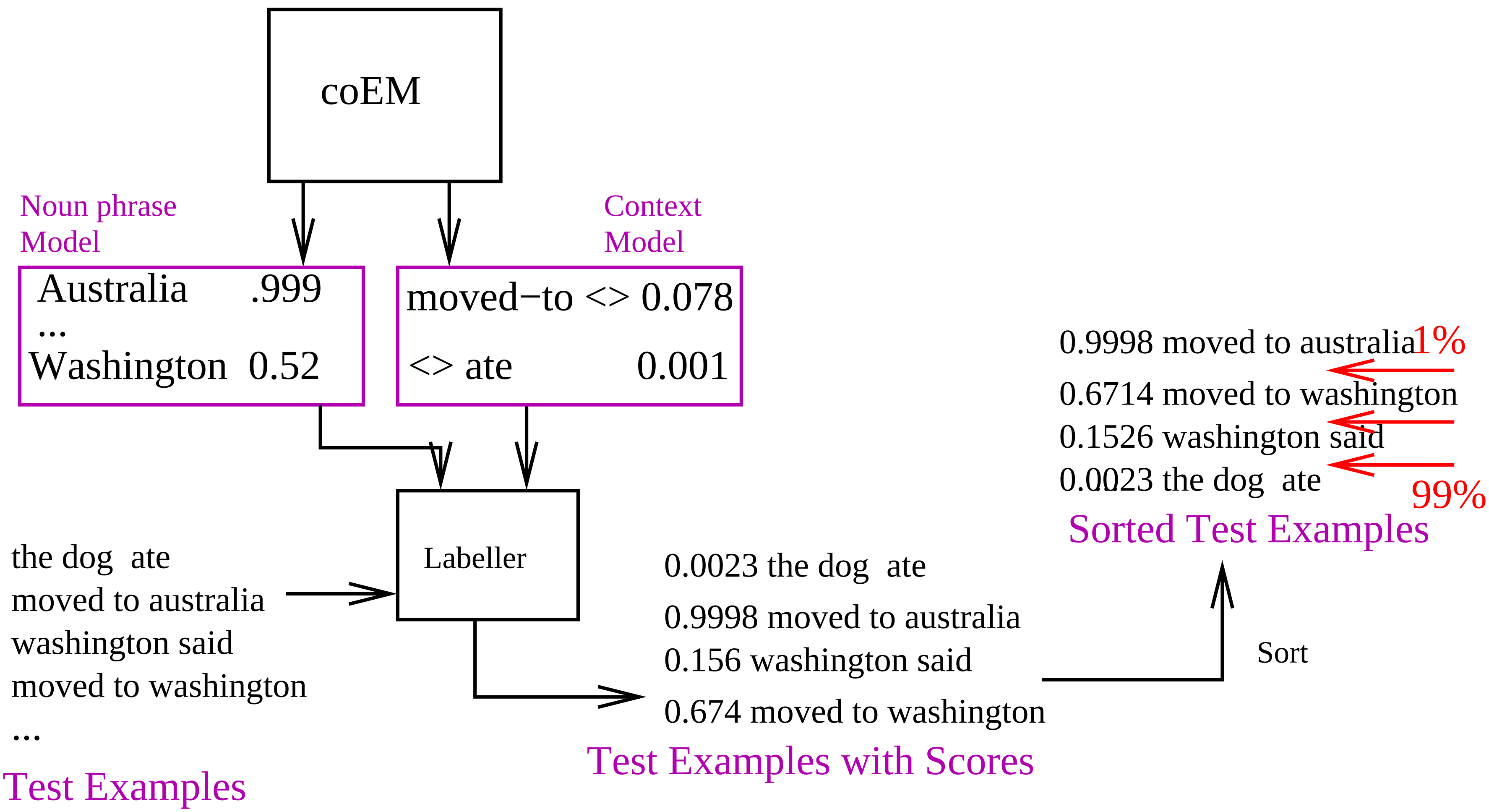

coEM
Noun phrase Model
Context Model
Australia .999
...
Washington 0.52
moved−to <> 0.078
<> ate 0.001

Evaluation



coEM
Noun phrase
Model
Context
Model
Australia .999
...
Washington 0.52
moved−to <> 0.078
<> ate 0.001
Labeller
the dog ate
moved to australia
washington said
moved to washington
...
Test Examples
0.0023 the dog ate
0.9998 moved to australia
0.156 washington said
0.674 moved to washington
Test Examples with Scores

Evaluation



coEM
Noun phrase Model
Context Model
Australia .999
...
Washington 0.52
moved−to <> 0.078
<> ate 0.001
Labeller
the dog ate
moved to australia
washington said
moved to washington
...
Test Examples
0.0023 the dog ate
0.9998 moved to australia
0.156 washington said
0.674 moved to washington
Test Examples with Scores
Sort
0.9998 moved to australia
0.6714 moved to washington
0.1526 washington said
0.0023 the dog ate
1%
99%
Sorted Test Examples

Evaluation

- $\widehat{P}(location|example) \sim \widehat{P}(location|NP) * \widehat{P}(location|context)$ for test collection

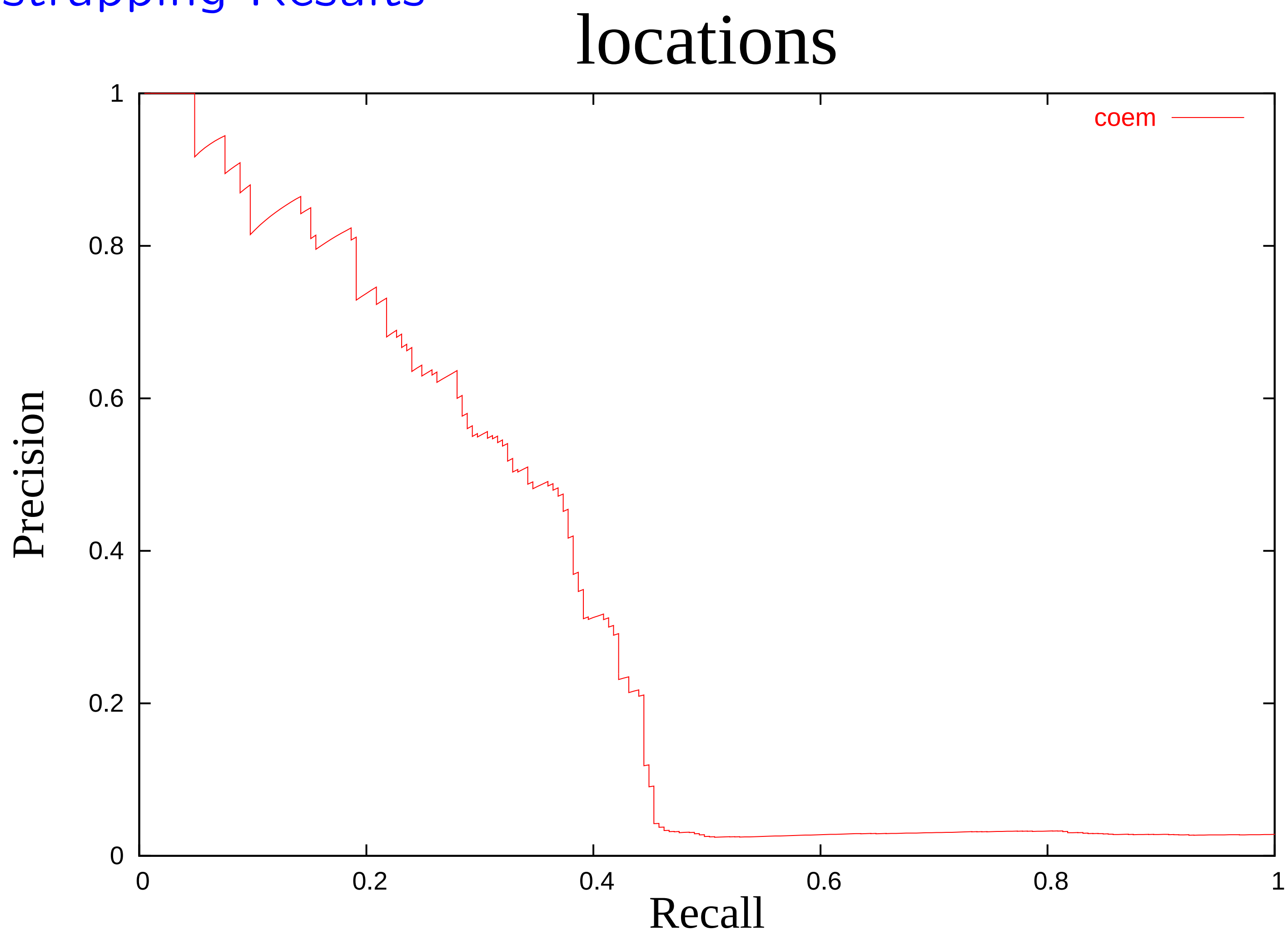
- sort test examples by $\widehat{P}(location|example)$: 800 cut points for precision-recall calculation

Precision and Recall at each of 800 points:

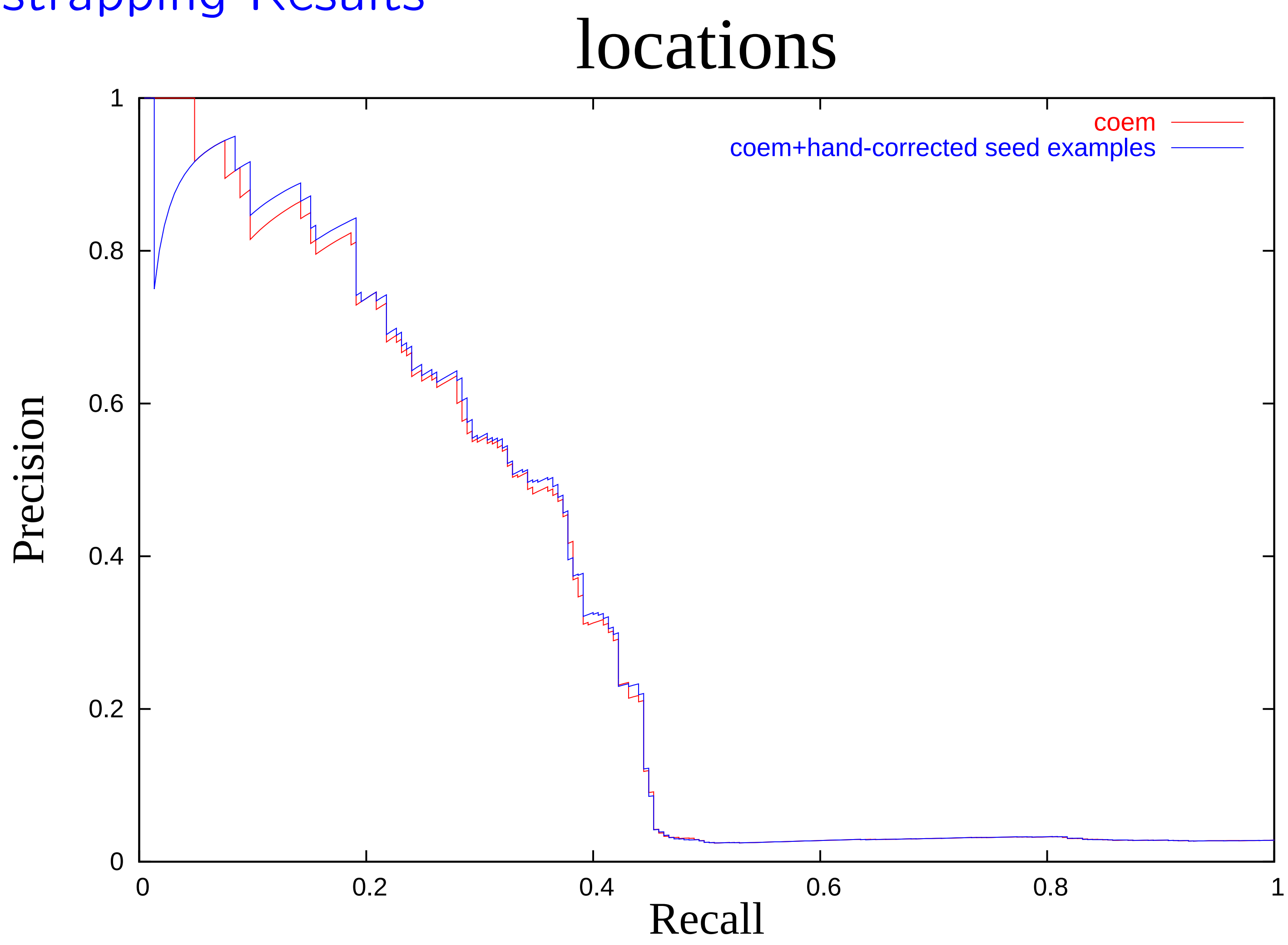
$$Precision = \frac{TargetClassRetrieved}{AllRetrieved}$$

$$Recall = \frac{TargetClassRetrieved}{TargetClassInCollection}$$

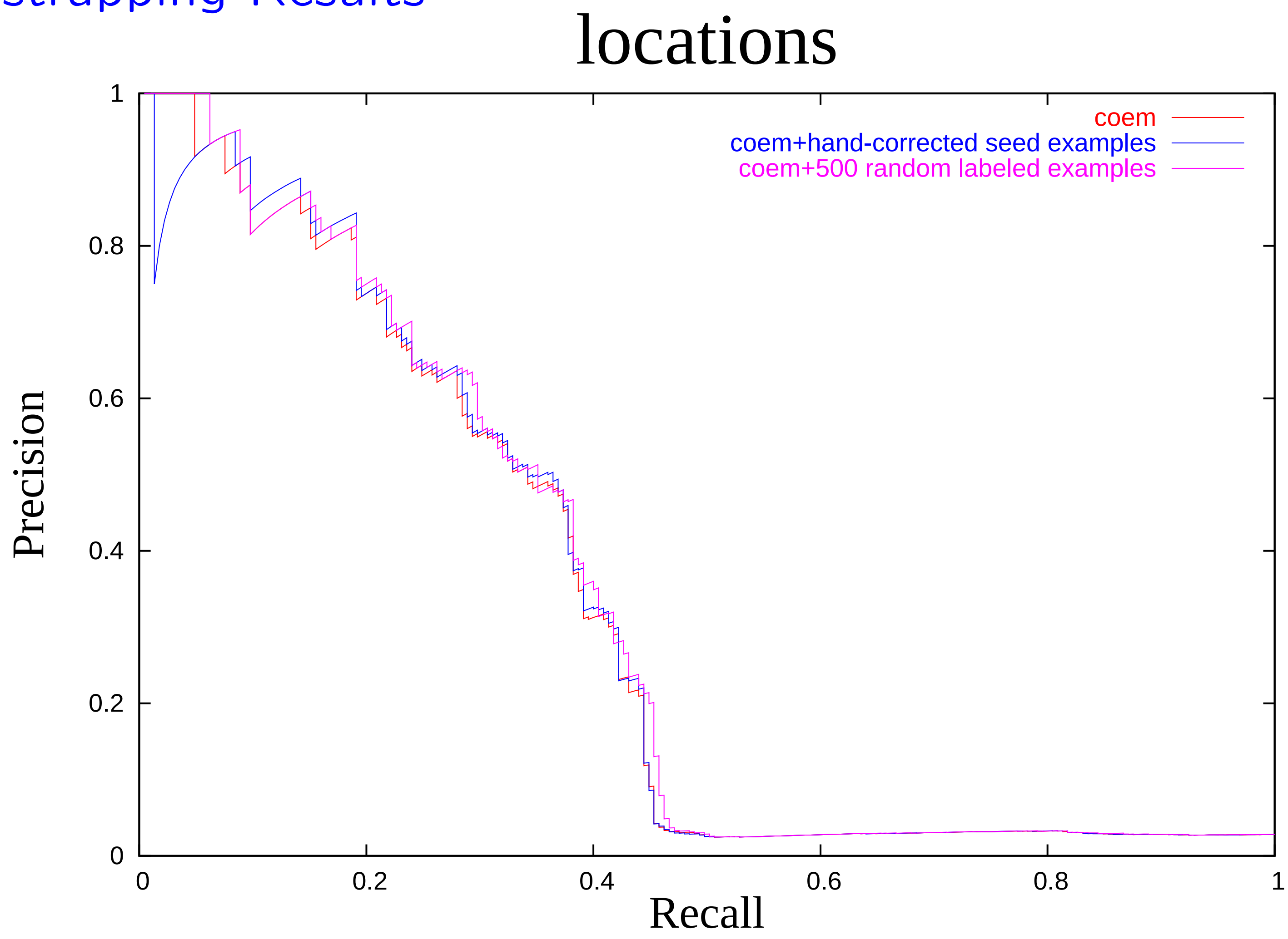
Bootstrapping Results



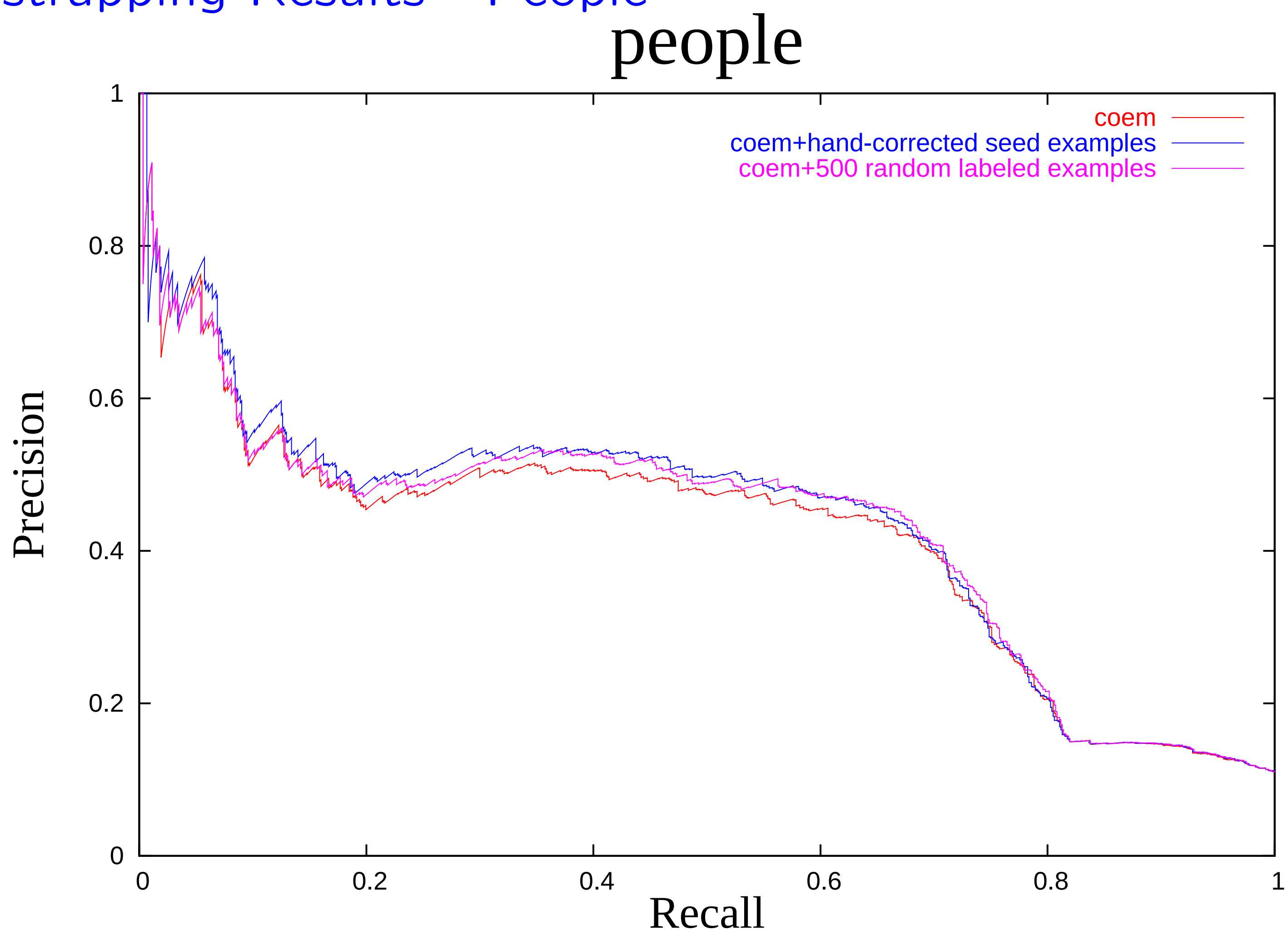
Bootstrapping Results



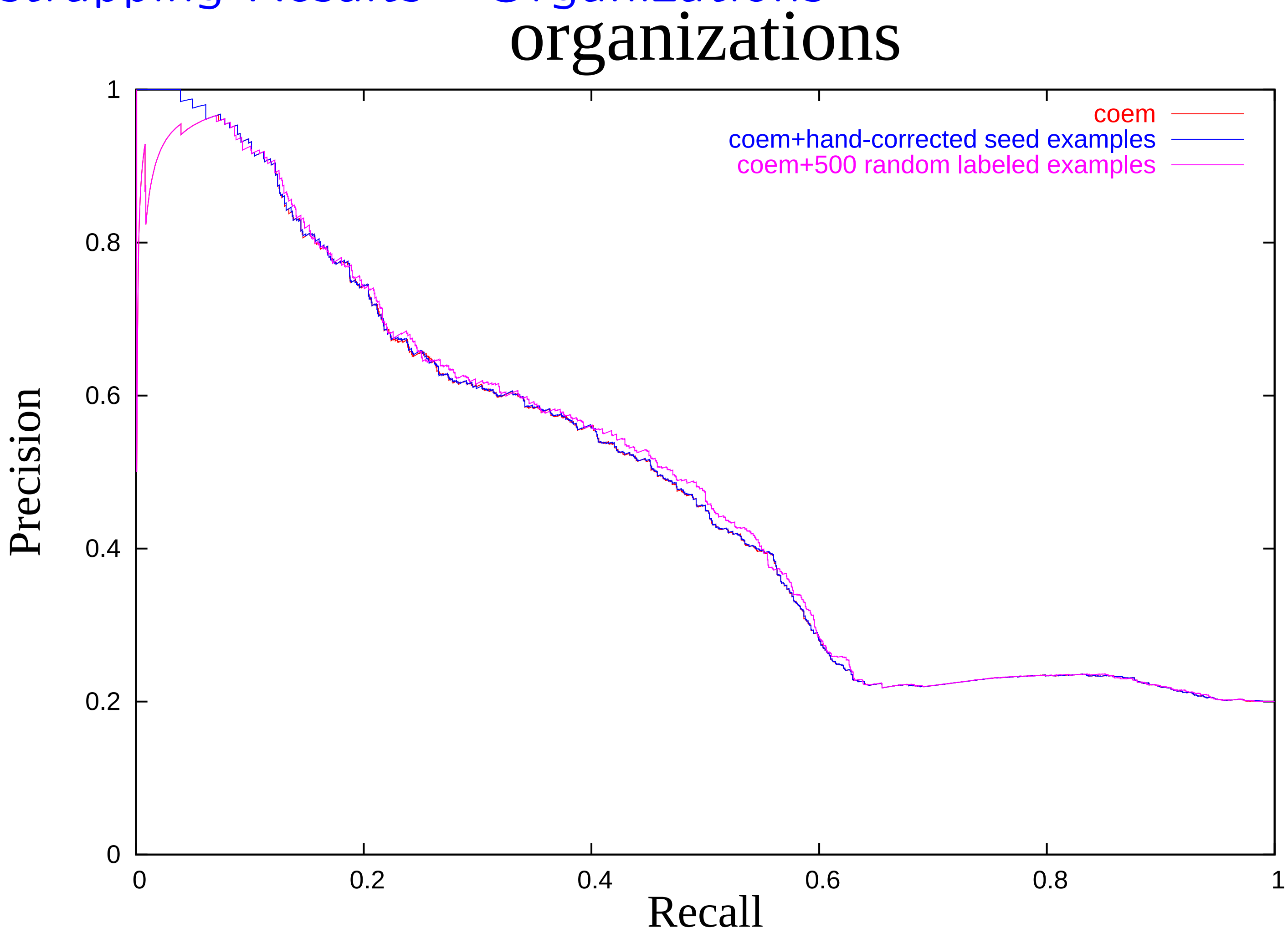
Bootstrapping Results



Bootstrapping Results - People



Bootstrapping Results - Organizations



We can Learn Simple Extraction Without Extensive Labeling

- Using just 10 seeds, we learned to extract from an unseen collection of documents
- No significant improvements from hand-correcting these examples
- No significant improvements from adding 500 labeled examples selected uniformly at random

We can Learn Simple Extraction Without Extensive Labeling

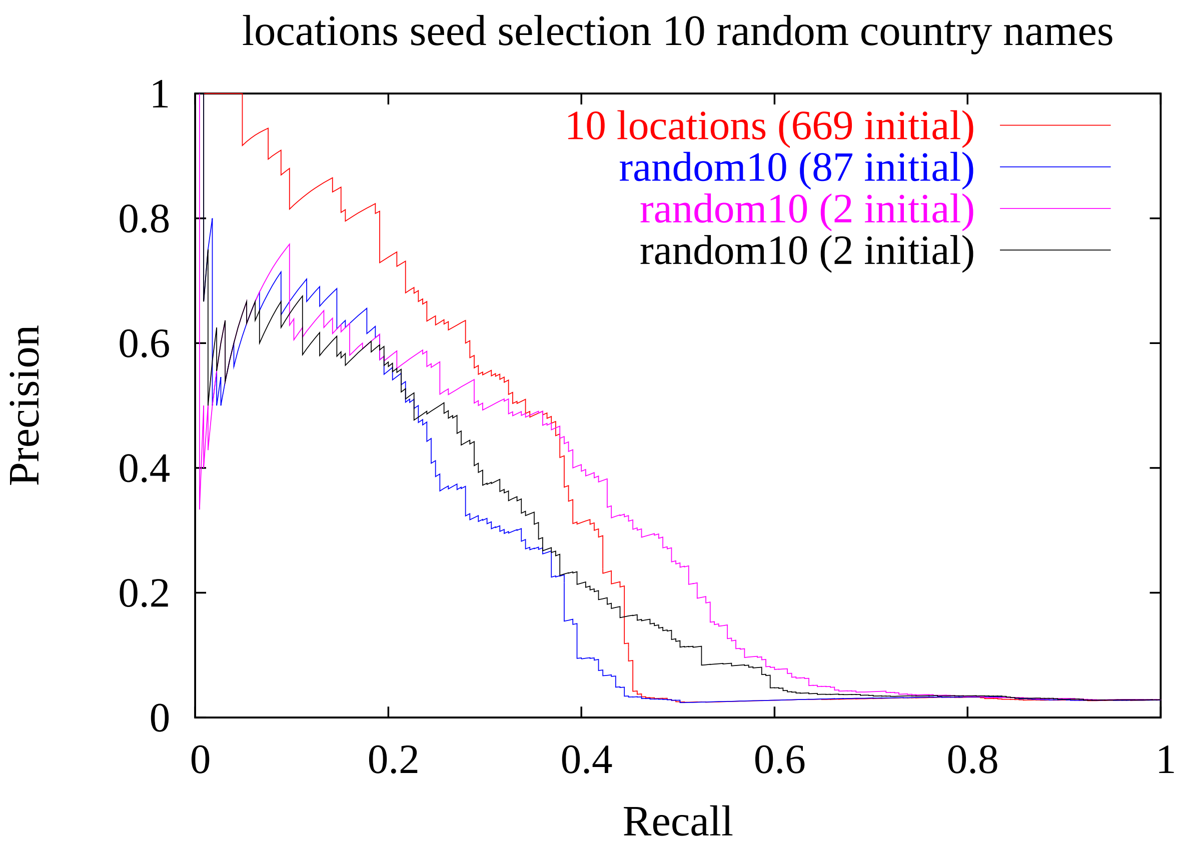
- Using just 10 seeds, we learned to extract from an unseen collection of documents

- No significant improvements from hand-correcting these examples

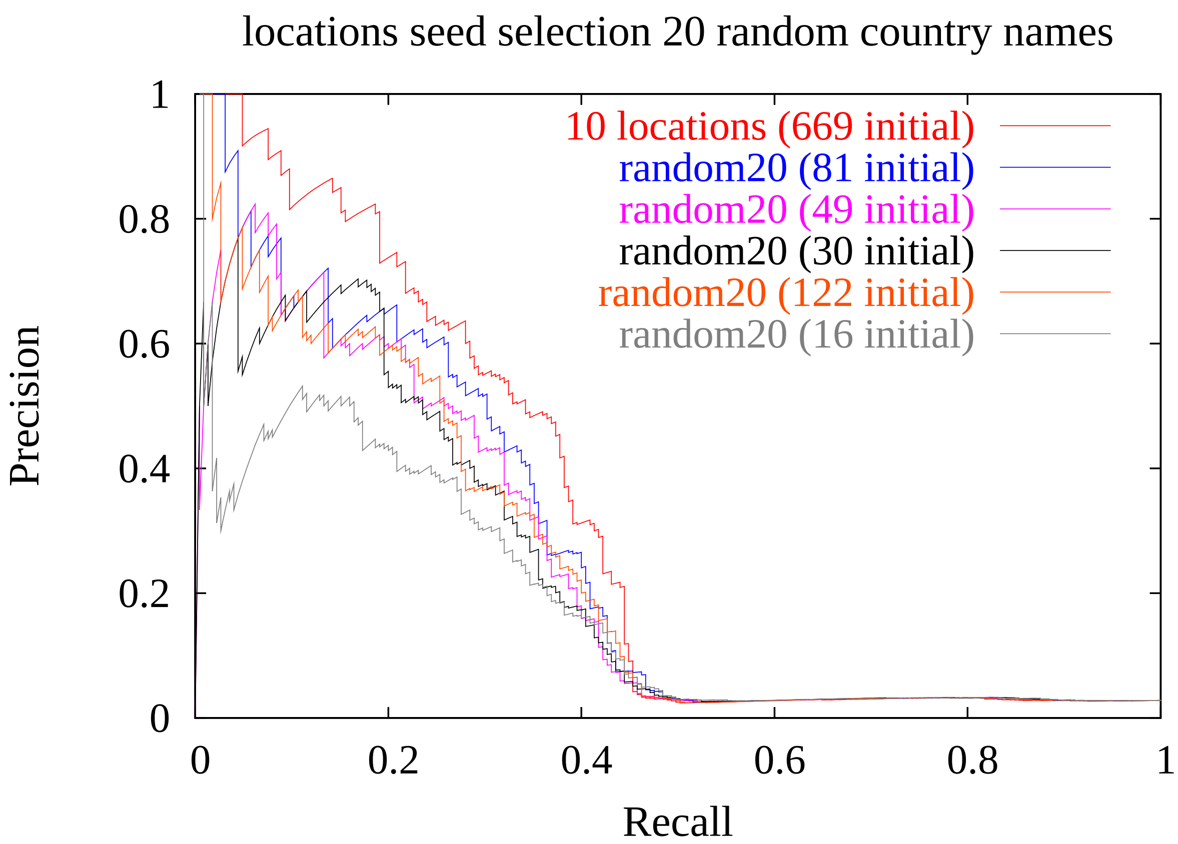
- No significant improvements from adding 500 labeled examples selected uniformly at random

- Did we just get lucky with the seeds?

Random Sets of Seeds Not So Good



Doubling the Number of Random Seeds Doesn't Help

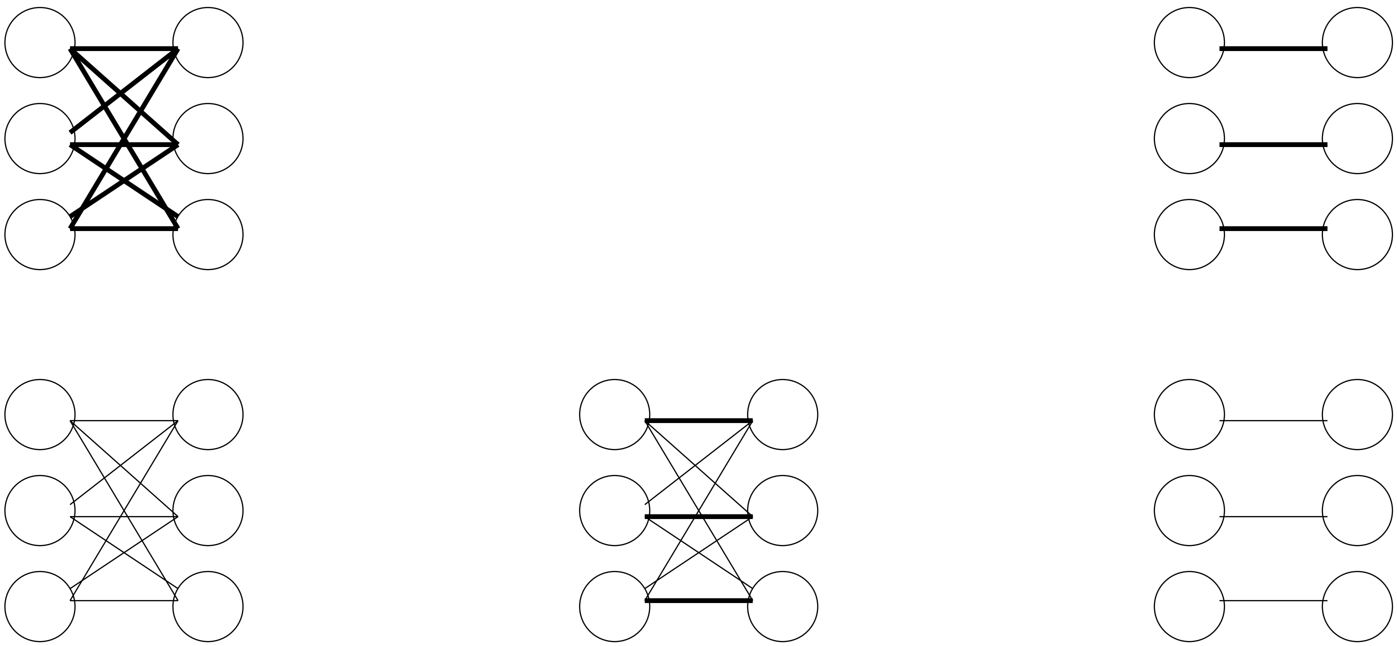


How does the set of seeds affect the performance? Something about the data?

Talk Outline

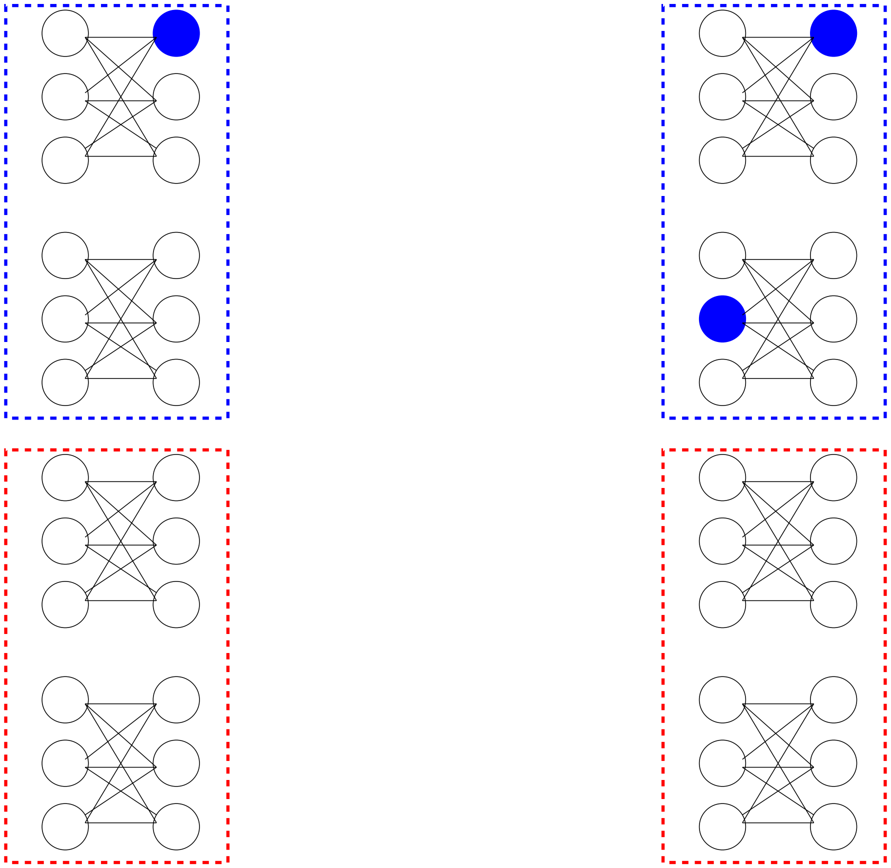
- Information Extraction
- Bootstrapping algorithm: coEM
- Understanding the Data: Graph Properties
- Active learning: Effective Use of User Time

What Properties of the Graph Might Affect Learning?



- Connectivity
- Mutual Information Given Class

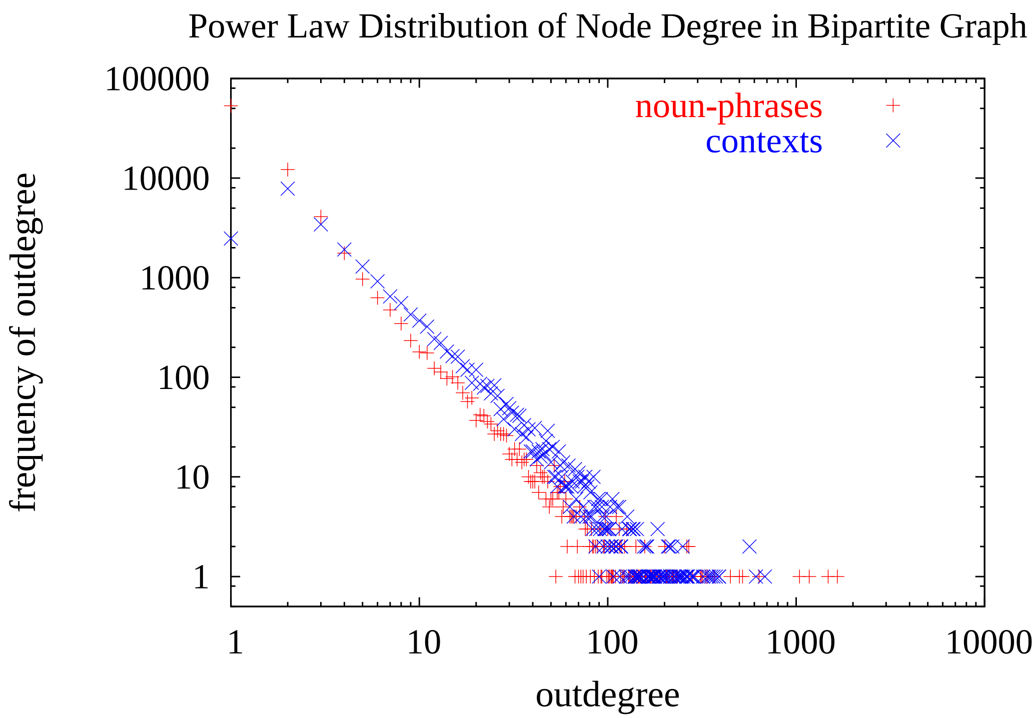
What about the Distribution of Initial Seeds?



What kind of Graph Structure Does Our Data Exhibit?

- How many components?
- What size components?
- Distribution of node degree?

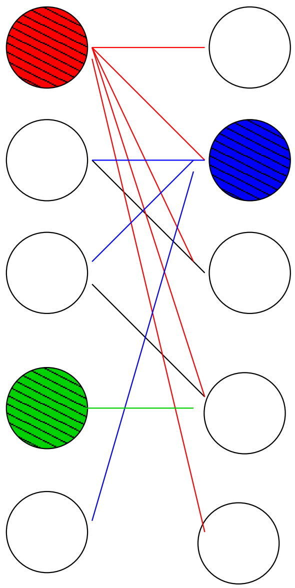
Node Degree is Power-Law Distributed



$$
\begin{aligned}
p_k &= ck^{-\alpha} \\
\log(p_k) &= \log(c) - \alpha \log(k)
\end{aligned}
$$

Power law coefficient $\alpha = 2.24$ for noun-phrases, 1.95 for contexts

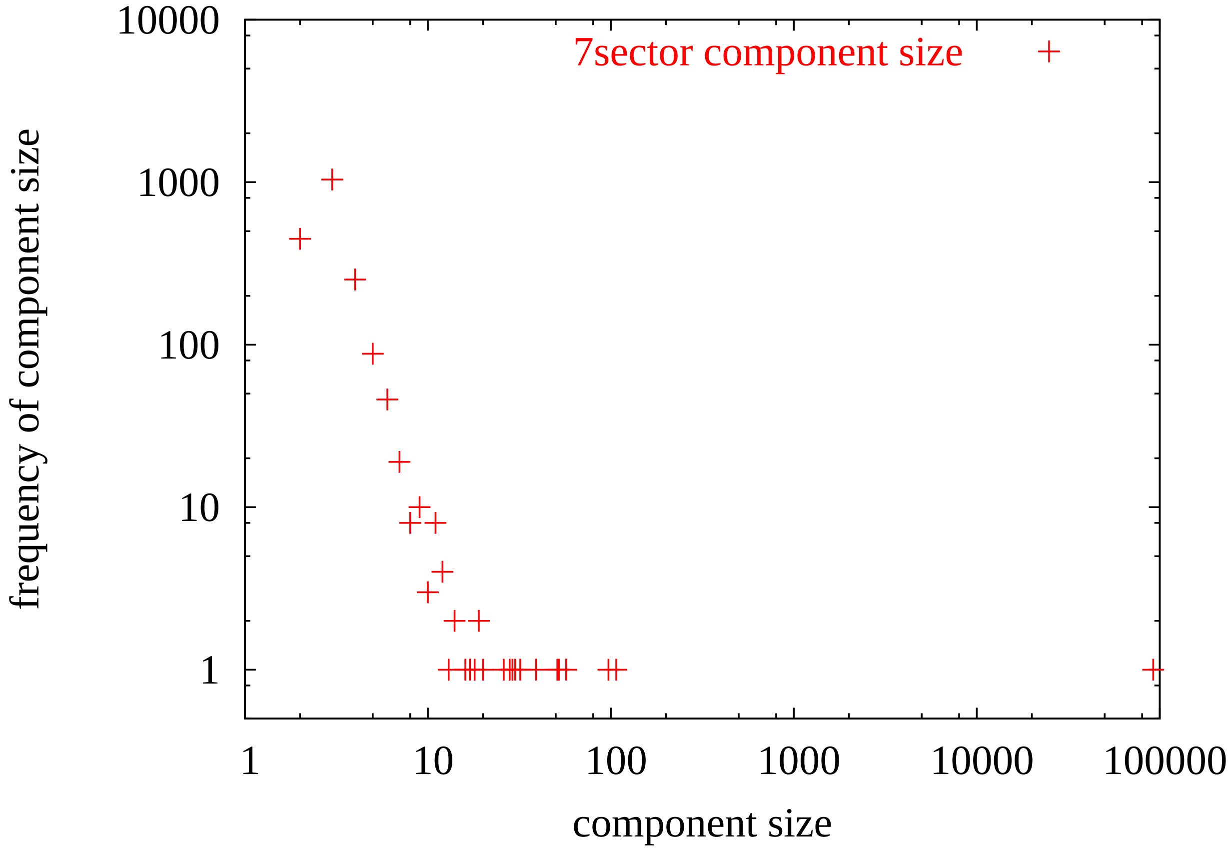
Some nodes are more important than others



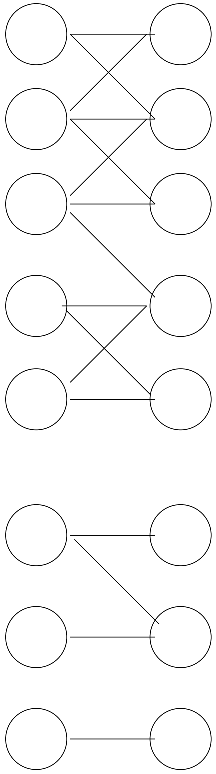
Noun-phrase	Outdegree
you	1656
we	1479
it	1173
company	1043
this	635
all	520
they	500
information	448
us	367
any	339
products	332
i	319
site	314
one	311
1996	282
he	269
customers	269
these	263
them	263
time	234

Context	Outdegree
<x> including	683
including <x>	612
<x> provides	565
provides <x>	565
provide <x>	390
<x> include	389
include <x>	375
<x> provide	364
one of <x>	354
<x> made	345
<x> offers	338
offers <x>	320
<x> said	287
<x> used	283
includes <x>	279
to provide <x>	266
use <x>	263
like <x>	260
variety of <x>	252
<x> includes	250

Component Size is Power-Law Distributed



Some Components Are More Important Than Others



Graph is Small-World

A small-world graph has:

- Characteristic path length similar to a random graph
- Clustering coefficient much higher than a random graph

	$\|V\|$	$\bar{k}$	L_{rand}	L	C	C_{rand}
noun-phrases	71,090	62	2.7	2.7	0.86	0.0018
contexts	21,039	265	1.78	2.54	0.74	0.025
bipartite	92,129	1.86	18	5.4	-	-

Short characteristic path length

$\Rightarrow$ Average shortest path between a pair of nodes is less than 6

High clustering coefficient

$\Rightarrow$ A node's neighbors are likely to be each other's neighbors

Why Should Graph Properties Affect Learning Performance?

- Small-world → Short path-lengths

  → All nodes in component reachable in few steps

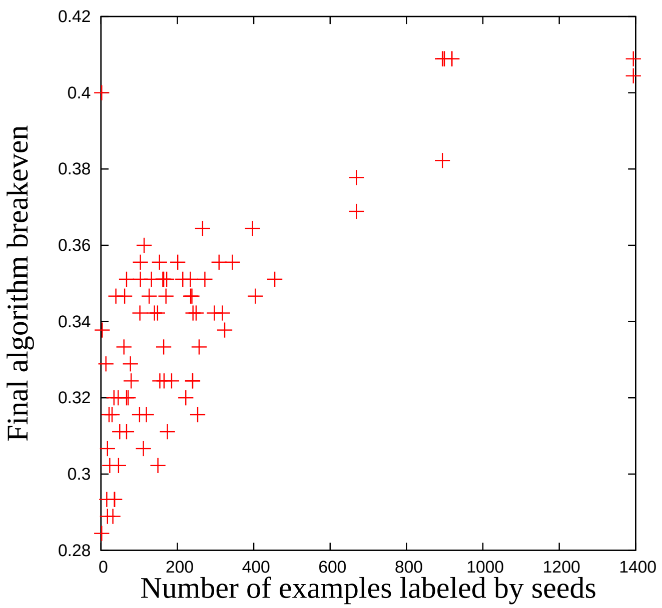
- Power-law → One large component, many small components

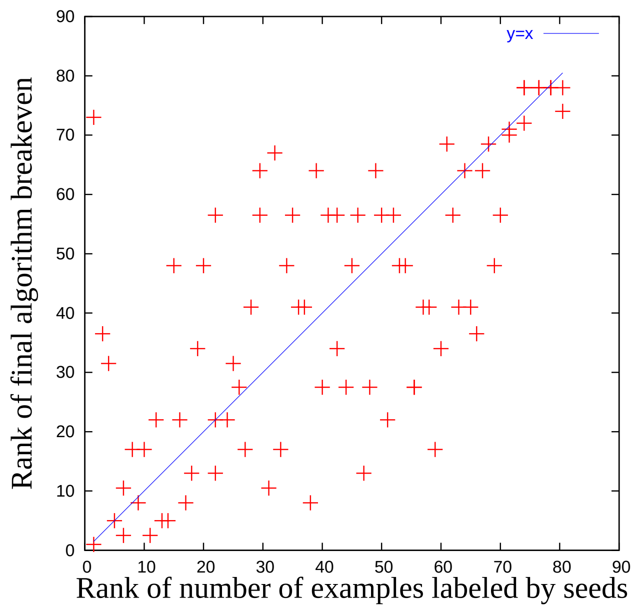
  → Distribution of seeds over components affects learning

- Power-law → Skewed distribution of node degrees

  → Node degree of labeled examples affects learning

Number of Examples Labeled By Seeds Correlates with Rank of Algorithm Breakeven





$$r_s = \frac{\sum_i (R_i - \overline{R_i})(S_i - \overline{S_i})}{\sqrt{\sum_i (R_i - \overline{R_i})^2}\sqrt{\sum_i (S_i - \overline{S_i})^2}} \qquad r_s = 0.678$$

Graph Features Explain Algorithm Performance

Feature	r_s
Num. unique seeds head-matching some NP in graph	**0.295**
Num. unique seeds exact-matching some NP in the graph	**0.302**
Num. unique seeds head-matching NPs in the largest component	**0.295**
Num. unique examples labeled (sum node degree)	**0.670**
Num. components containing at least one seed	**0.541**
Num. unique seed-examples in the largest component	**0.669**
Num. unique contexts covered by seeds	**0.657**
Total examples labeled	**0.678**
Num. unique contexts covered by more than one seed	**0.716**

Contexts Selected by Location Seeds

Context	Num Seeds Selected By
operations:in <X>	10
locations:in <X>	9
<X> comments	8
<X> updated	7
offices:in <X>	6
operates:in <X>	6
headquartered:in <X>	6
facilities:in <X>	5
customers:in <X>	5
owned:in	1
originated:in	1
grown:in <X>	1
found:in <X>	1
filed:in <X>	1
due:in <X>	1
targeting $< X >$	1
covering <X>	1

Graph Features in Combination Explain Algorithm Performance

Num. unique seeds head-matching NPs in largest component
Total examples labeled
Num. unique seed-labeled-examples in largest component
Num. unique contexts covered by more than one seed

Correlation of 0.78 with algorithm performance

Statistically significantly higher correlation than best single feature correlation (0.72)

Contributions to Understanding Graph Properties and Bootstrapping

- Number of seeds (examples) is not the biggest factor
- Overlap of those seeds' contexts (disambiguation, generalization)
- Distribution of seeds over graph components
- Combination of these factors affects performance

Talk Outline

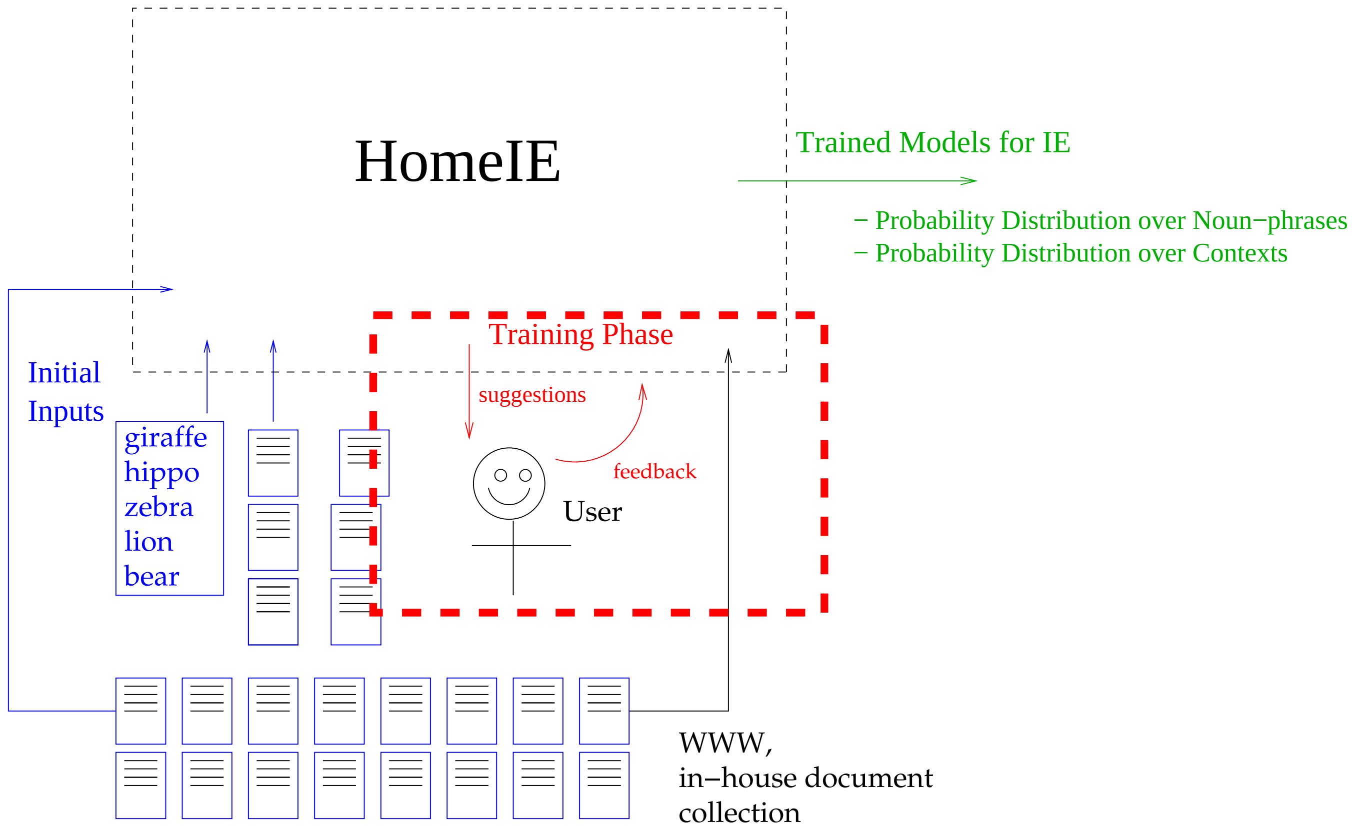
- Information Extraction
- Bootstrapping algorithm: coEM
- Understanding the Data: Graph Properties
- Active learning: Effective Use of User Time

Active Learning Question

- How can we improve results by asking the user some questions?

- Is there a way to be most efficient with user time?

Active Learning



Active Learning Methods I

- Uniform Random Selection

- Density-based selection

$$Score(np, context) = freq(np, context)$$

Active Learning Methods II

- NP-Context Disagreement (novel)
  Kullback Leibler divergence to the mean, weighted by example density

$$KL(\hat{P}_{f_1}(+|e), \hat{P}_{f_2}(+|e)) = \sum_i \hat{P}_{f_i}(+|e) \frac{log \hat{P}_{f_i}(+|e)}{log(\hat{P}_{mean}(+|e))}$$

NP	score	context	score	freq	freq * KL
mexico	1	gulf of <X>	0.66	27	19.83
united states	1	trademark in <X>	0.44	12	6.65
united states	1	regions of <X>	0.66	4	3.12

Active Learning Methods III

- Context-disagreement (novel)

$$score(NP) = freq(NP) * KL(context_1..context_n)$$

NP	contexts	score	freq	freq * KL
de benelux	offices:in <X>	0.10	23	2.63542
	consulting:in <X>	0.16		
	office:in <X>	0.036		
	support:in <X>	0.05		
	seminars:in <X>	0.22		
	distributors:in <X>	0.18		
italy	centers:in <X>	0.05	14	1.22012
	operations:in <X>	0.24		
	<X> updated	0.10		
	<X> updated:1997	0.28		
	<X> comments	0.03		
	introduced:in <X>	0.11		
	partners:in	0.02		
	offices:in	0.19		

Which Properties are Correlated With Rank of Active Learning Performance?

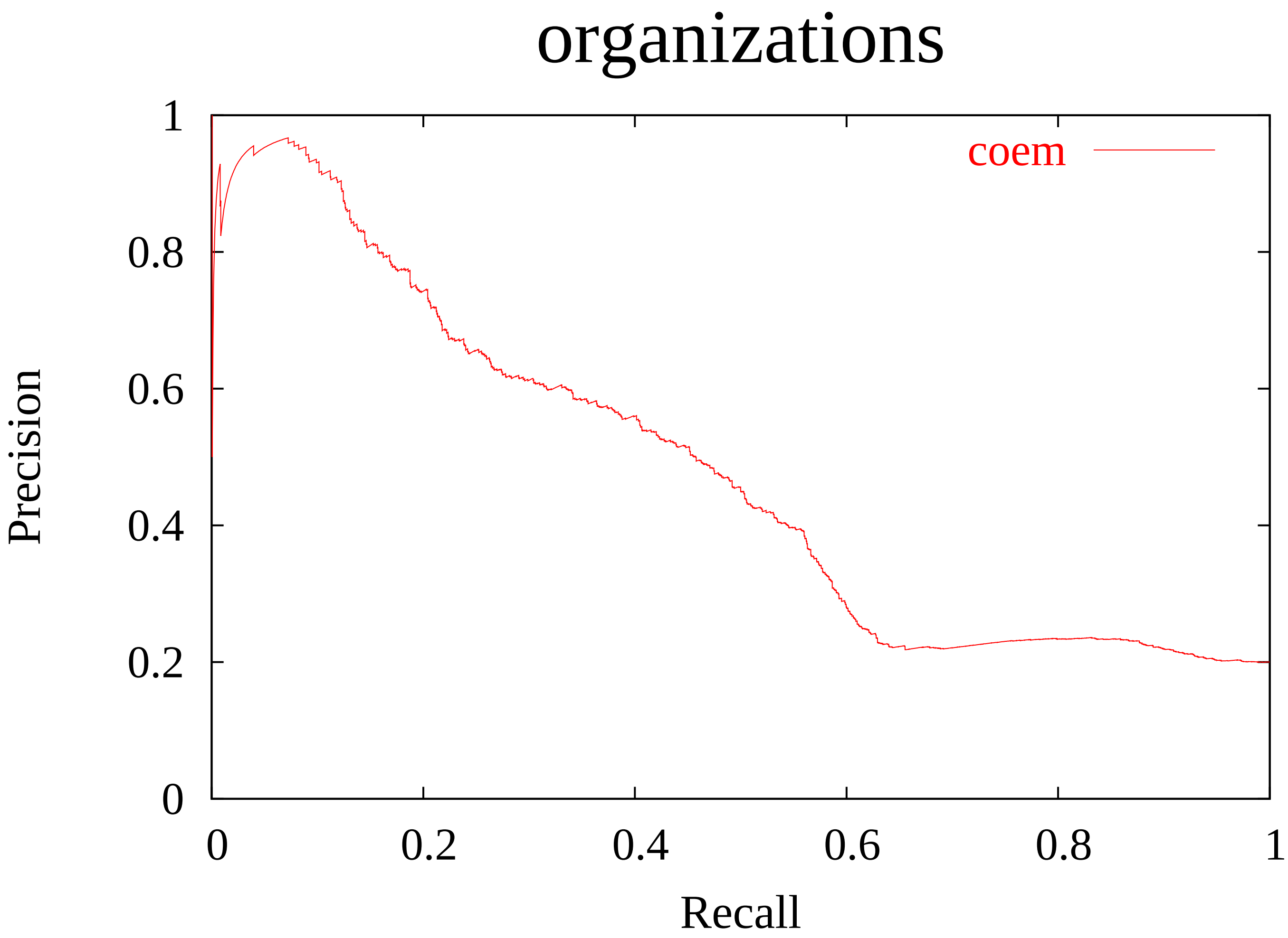
Feature	$r_{s_{act.}}$	$r_{s_{base}}$
Num. unique seeds head-matching	**0.282**	0.295
Num. unique seeds exact-matching	**0.285**	0.302
Num. unique seeds head-matching in largest component	**0.282**	0.295
% positive examples labeled during active learning	**0.167**	
% nonseed examples labeled positive during active learning	**0.167**	
Num. examples labeled during active learning	**0.434**	
Num. positive examples labeled during active learning	**0.460**	
Num. nonseed examples labeled during active learning	**0.434**	
Num. nonseed examples labeled positive during active learning	**0.460**	
Num. unique examples labeled (sum node degree)	**0.630**	0.670
Num. components containing at least one example	**0.501**	0.541
Num. components containing at least one seed or positive example	**0.529**	0.541
Num. unique seed or positive examples in largest component	**0.624**	0.669
Num. unique contexts covered by seeds	**0.551**	0.657
Num. unique contexts covered by more than one seed	**0.581**	0.716
Total examples labeled	**0.628**	0.678

Graph Features in Combination Explain Active Learning Performance

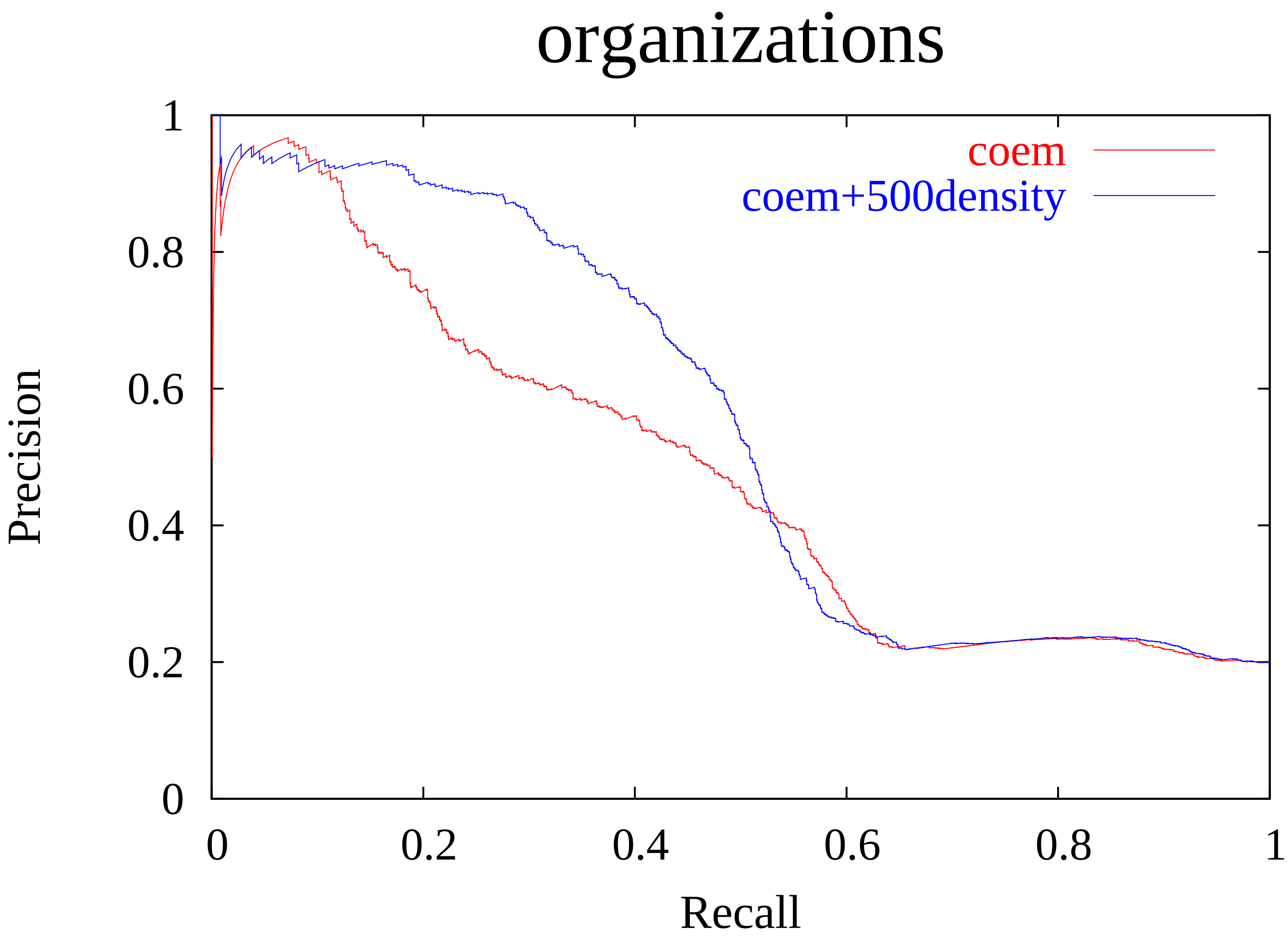
Features
Num. unique seeds head-matching NPs in the largest component
Num. unique examples labeled
Total examples labeled
Num. unique contexts covered by seeds
Num. unique contexts covered by more than one seed
Num. positive examples labeled during active learning

The correlation of this model with algorithm performance is 0.73, greater than the correlation of any individual feature in isolation (0.63)

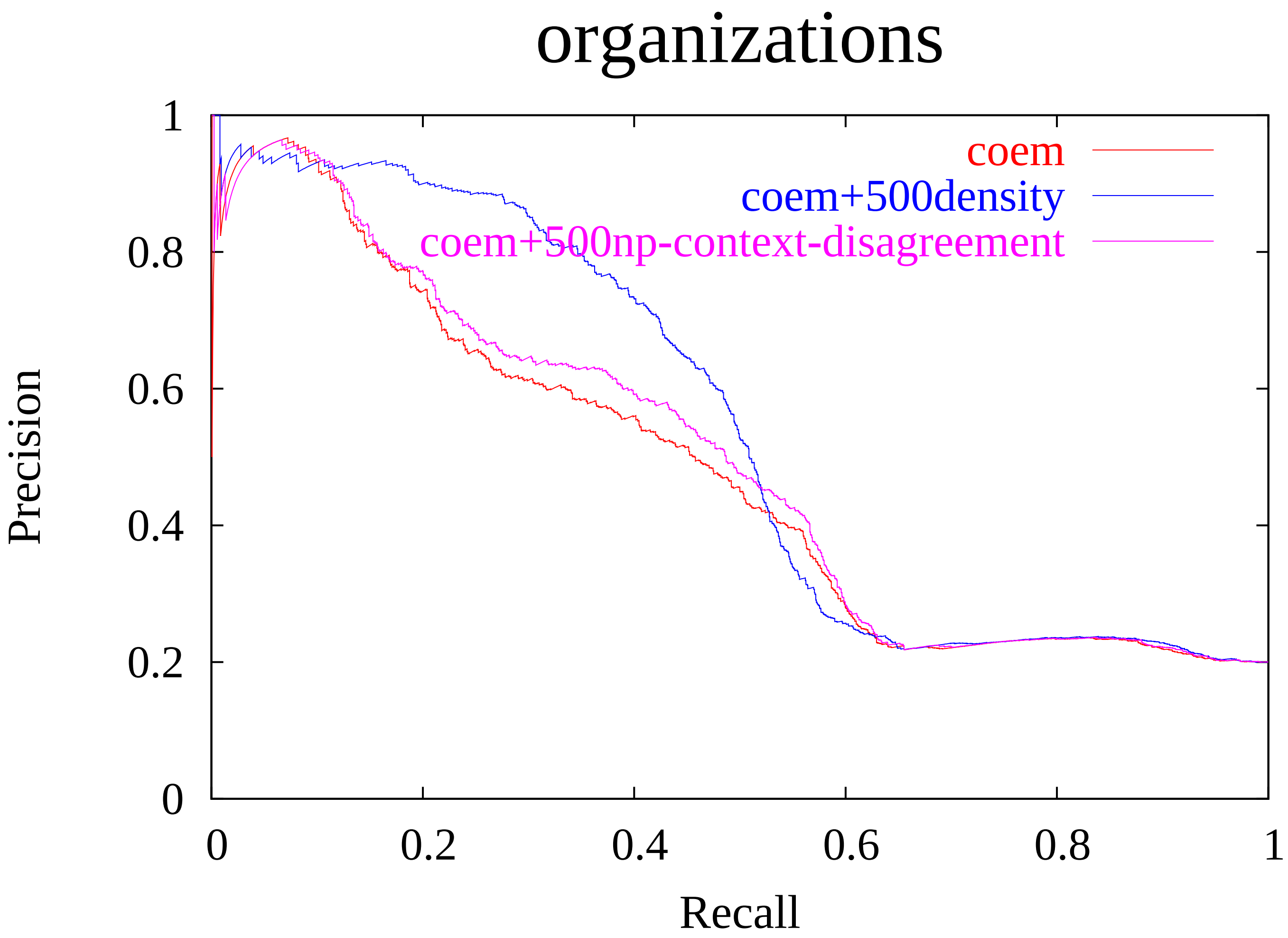
Active Learning Results



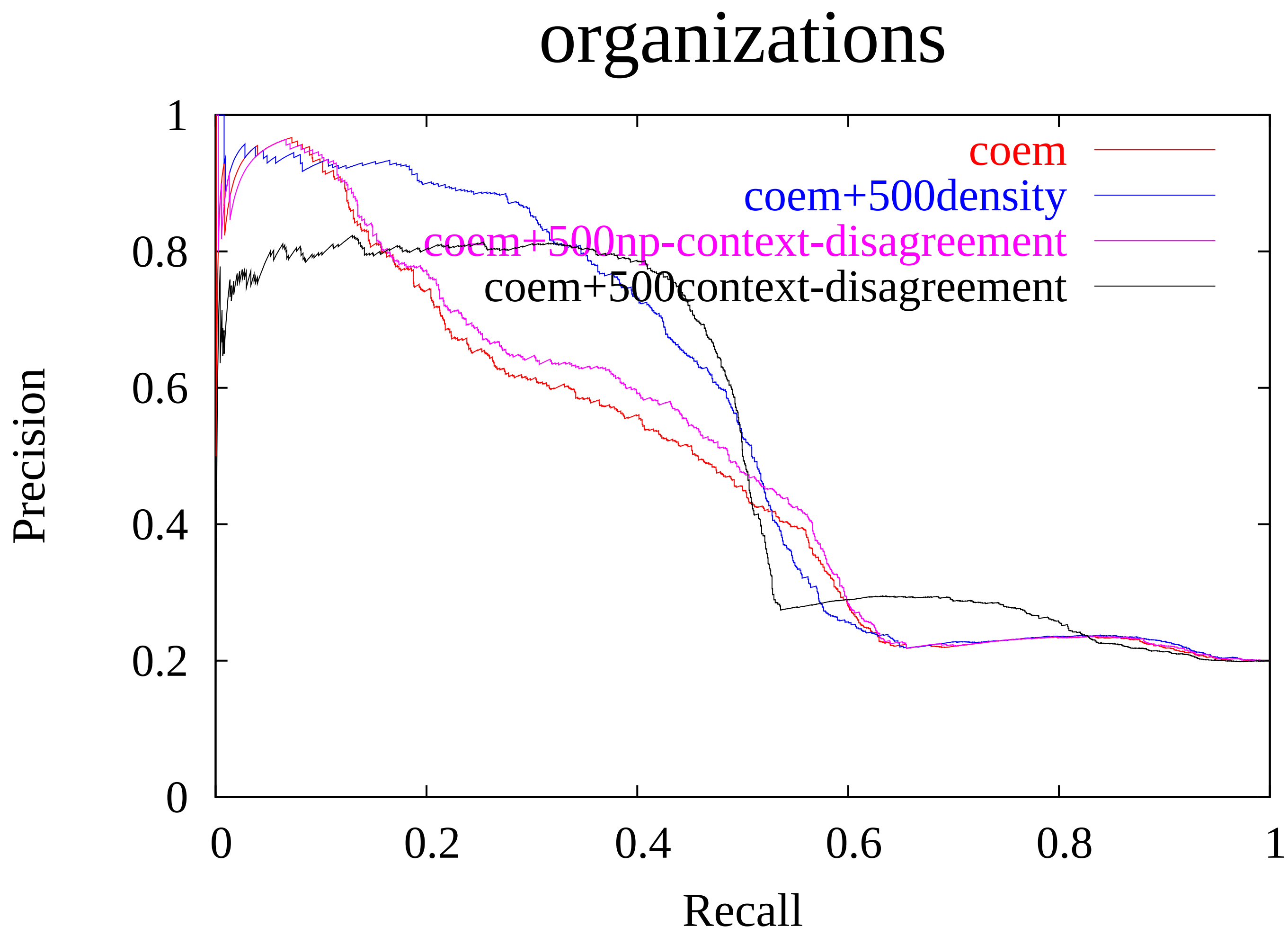
Active Learning Results



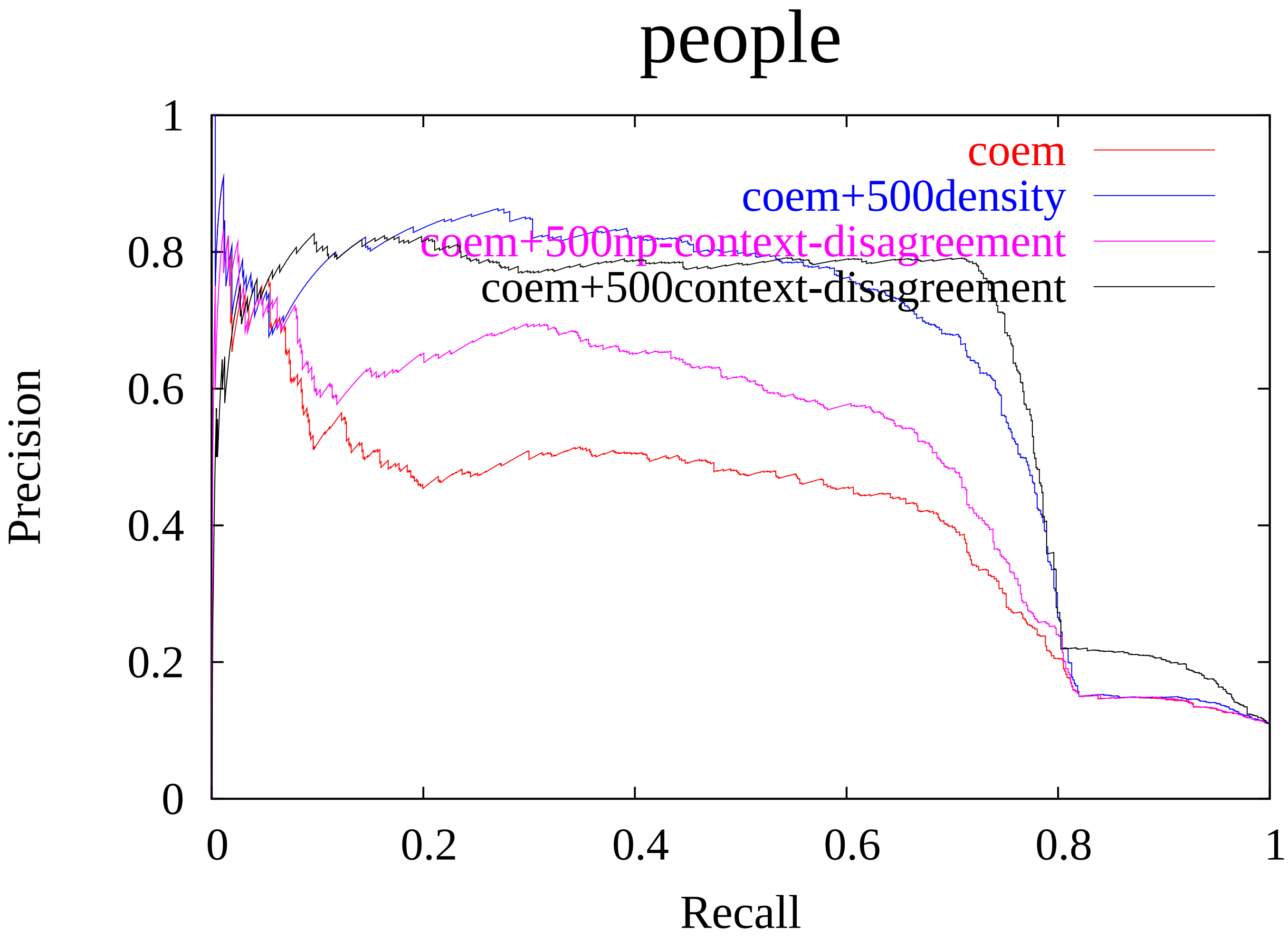
Active Learning Results



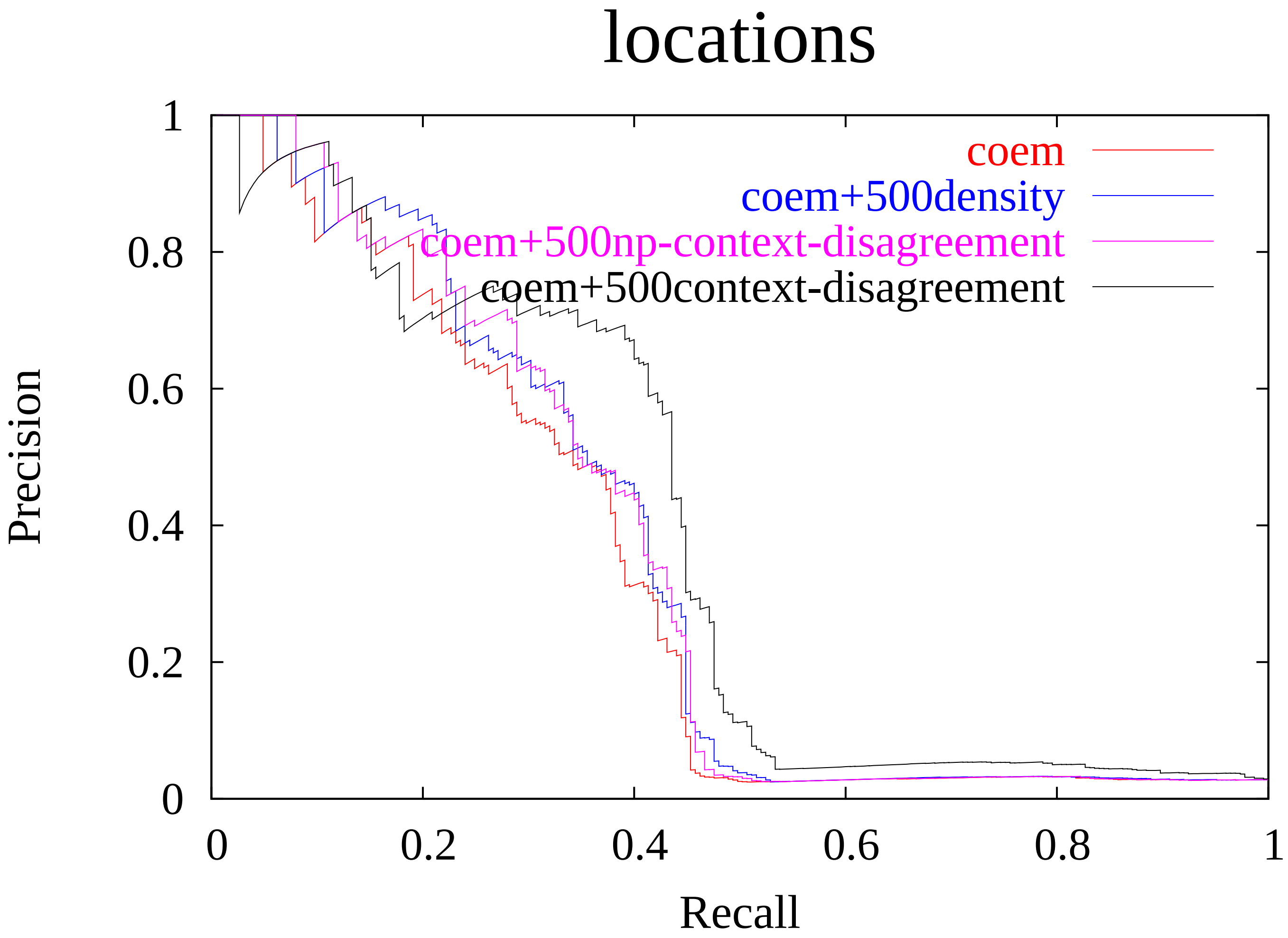
Active Learning Results



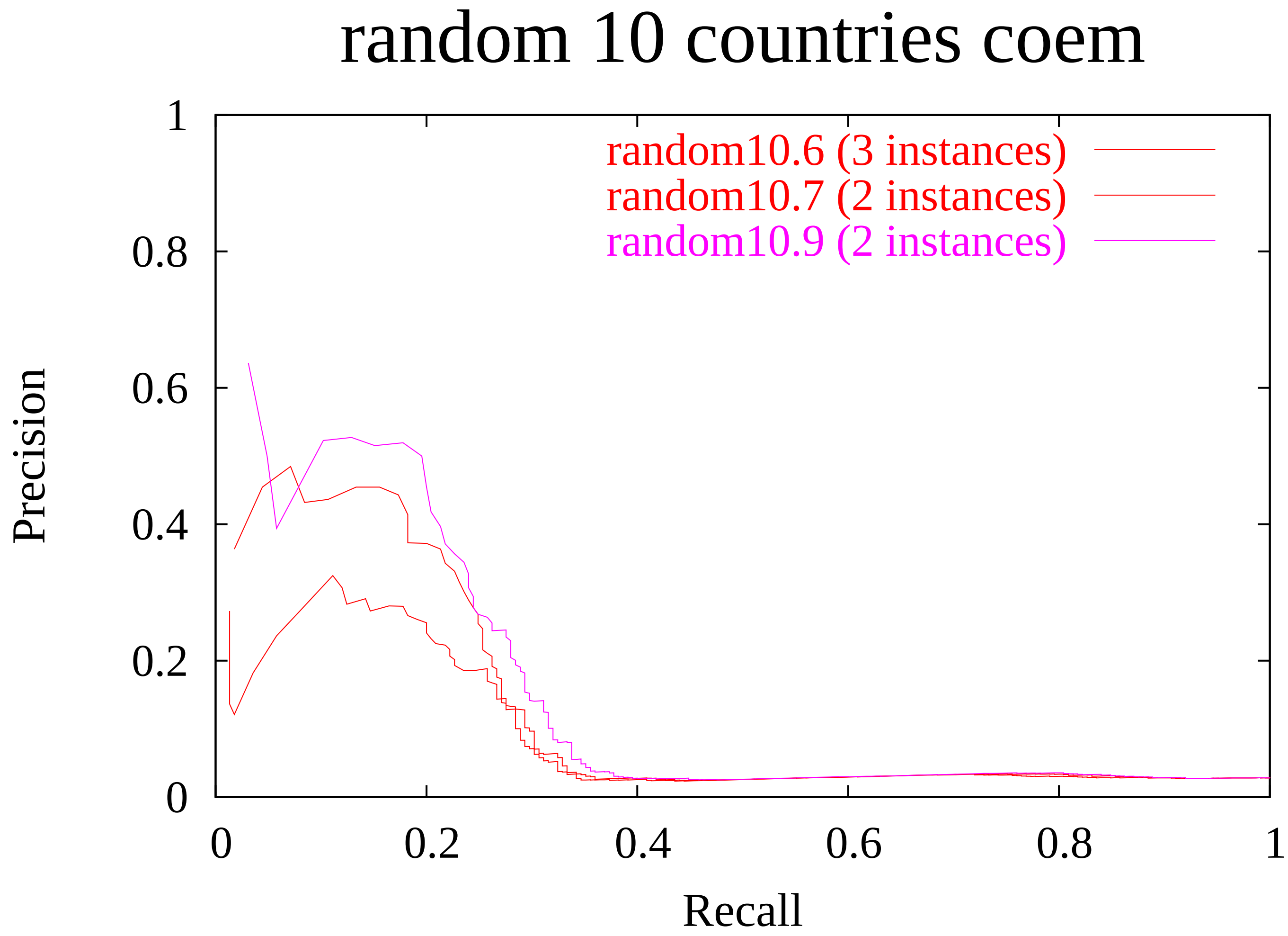
Active Learning Results



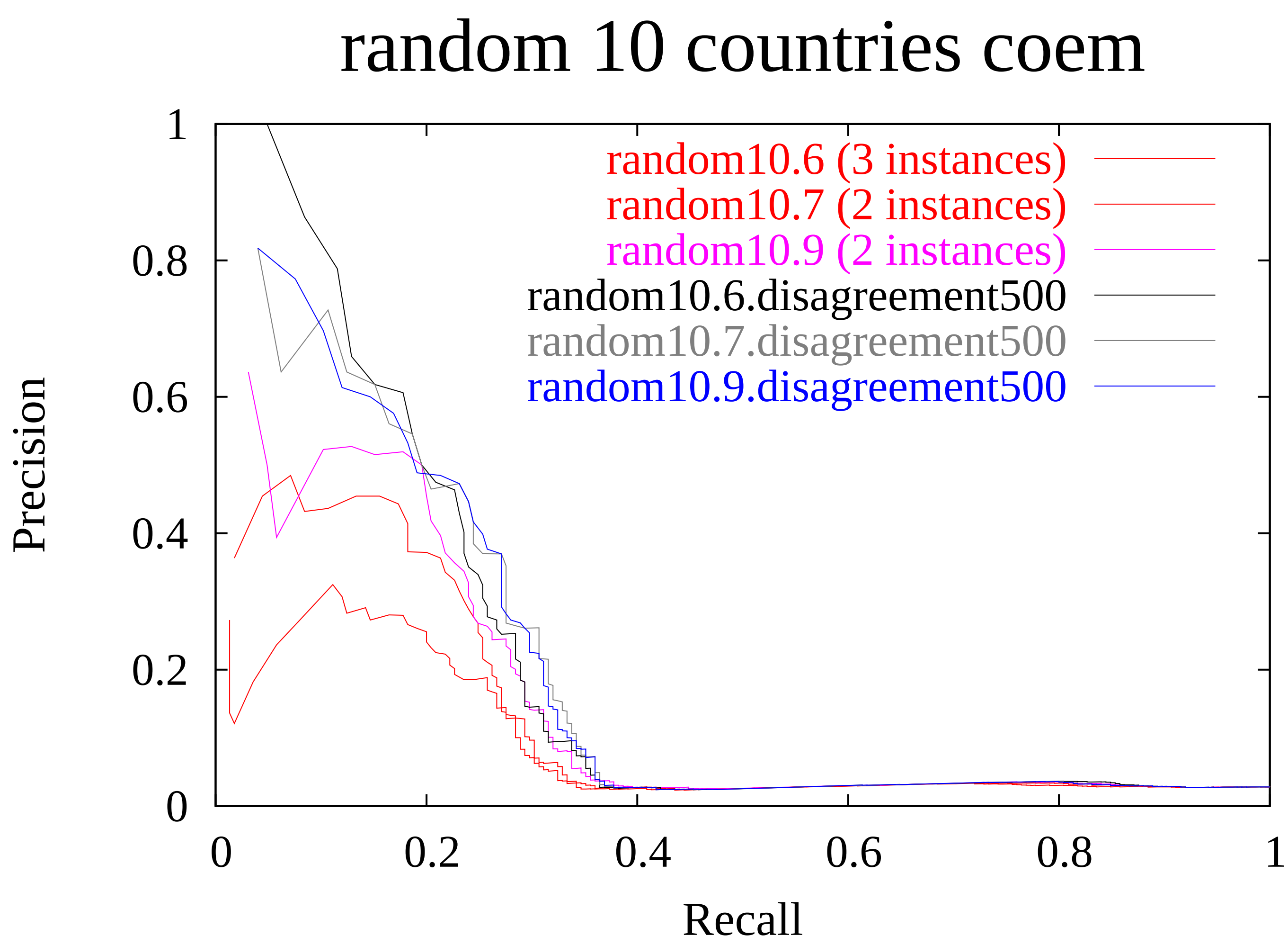
Active Learning Results



Active Learning Compensates for Infrequent Seeds



Active Learning Compensates for Infrequent Seeds



Contributions Summary

- In-depth experiments with bootstrapping algorithms across multiple semantic classes.
- Adapted existing semi-supervised learning algorithms for the task of information extraction.
- Novel active learning algorithms that take into account the feature set split into two sets.
- Analysis of the noun-phrase context co-occurrence graph to show that it exhibits small-world and power-law structure.
- Demonstration of the correlation between graph features and algorithm performance

Now we Know How to Select Seeds for Bootstrapping

- Identify the heads of noun-phrases
- Sort noun-phrases by their node degree
- Examine list till we have seen several seeds in the target class
- Examine list till we have seen at least one seed in the largest component

Now we Know If Our Target Class is Learnable with Bootstrapping

- We can find seeds in our corpus

- Overlap between the contexts of the seeds

- Active learning if few examples extracted by seeds

Now we Know How to Modify Active Learning for Bootstrapping

- Density-weighted example selection
- Prefer examples from largest component
- Select examples from unlabeled components
- Prefer likely positive examples for sparse class

Applying What We've Learned to a New Task

Traditional way: Asked three people for example seed-words for "products"

Labeler-set	Seeds	n
1-a	20GB iPod, Jetclean II, Tungsten T5, InFocus ScreenPlay 4805 DLP Projector, Sony PSP, Barbie Fairytopia, Crayola Construction Paper Crayons, Kodak Advantix 200 Speed Color Film, Timbuk2 Commute Messenger Bag, Sony MDR-V6 Stereo Headphones	0
1-b	mp3 player, Maytag dishwasher, Palm Pilot, home theater projector, PSP, Barbie, crayons, 35mm film, messenger bag, headphones	100
2-a*	Nestle, disposable razor, Toyota Prius, SUV, Armani Suit, Yemen Mocha Matari, 8" 2x4, cheddar cheese, HP Compaq nc6000, q-tips	5
2-b	Lipton Tea, 00 buckshot, Tomatoes, Loose-leaf paper, Nike shoes, Basil seeds, 2004 Toyota Camry SE, Laptop battery, Gummibears, M&Ms	83
3	Leather sofa, Electric violin, Chocolate cake, Mountain bike, Pair of glasses, K2 Rollerblades, Ipod, Dress shirt, Headphones, Webcam	20

Our Proposed New Method:
Selecting Seeds from 200 Most Frequent NPs

Seed-word	nps	examples	u. np-heads	u. Cont.	ex. Cont.
services	2711	7236	2427	4333	provides <x>, offers <x>, range of <x>
software	2679	7100	2159	4581	use of <x>, use <x>, <x> provides
products	2113	6281	2267	3952	information on <x>, range of <x>, line of <x>

20,311 unique examples labeled by these seed-words

Comparison

- Baseline: Seeds chosen by introspection + coEM
- Our new approach: Seeds chosen by inspecting frequent NPs + coEM + feature set disagreement active learning

Training corpus: large sample from TREC w10g

Test corpus: held out data

Evaluation Measures

- Precision for dictionary construction
  - Evaluate top-scoring 200 noun-phrases
  - Evaluate top-scoring 200 noun-phrases which do not match seeds
- Precision for extraction on held-out documents
  - Evaluate top-scoring extracted examples
  - Evaluate top-scoring extracted examples which do not match seeds

Results on New Task

	nps	nps (non-seed)	Examples	Examples (non-seed)
P@1	1	0	1	1
P@10	0.8	0.1	0.4	0.4
P@50	0.28	0.2	0.22	0.22
P@100	0.35	0.28	0.31	0.31
P@200	0.32	0.29	0.39	0.39

Seeds = Leather sofa, Electric violin, Chocolate cake, Mountain bike, Pair of glasses, K2 Rollerblades, Ipod, Dress shirt, Headphones, Webcam

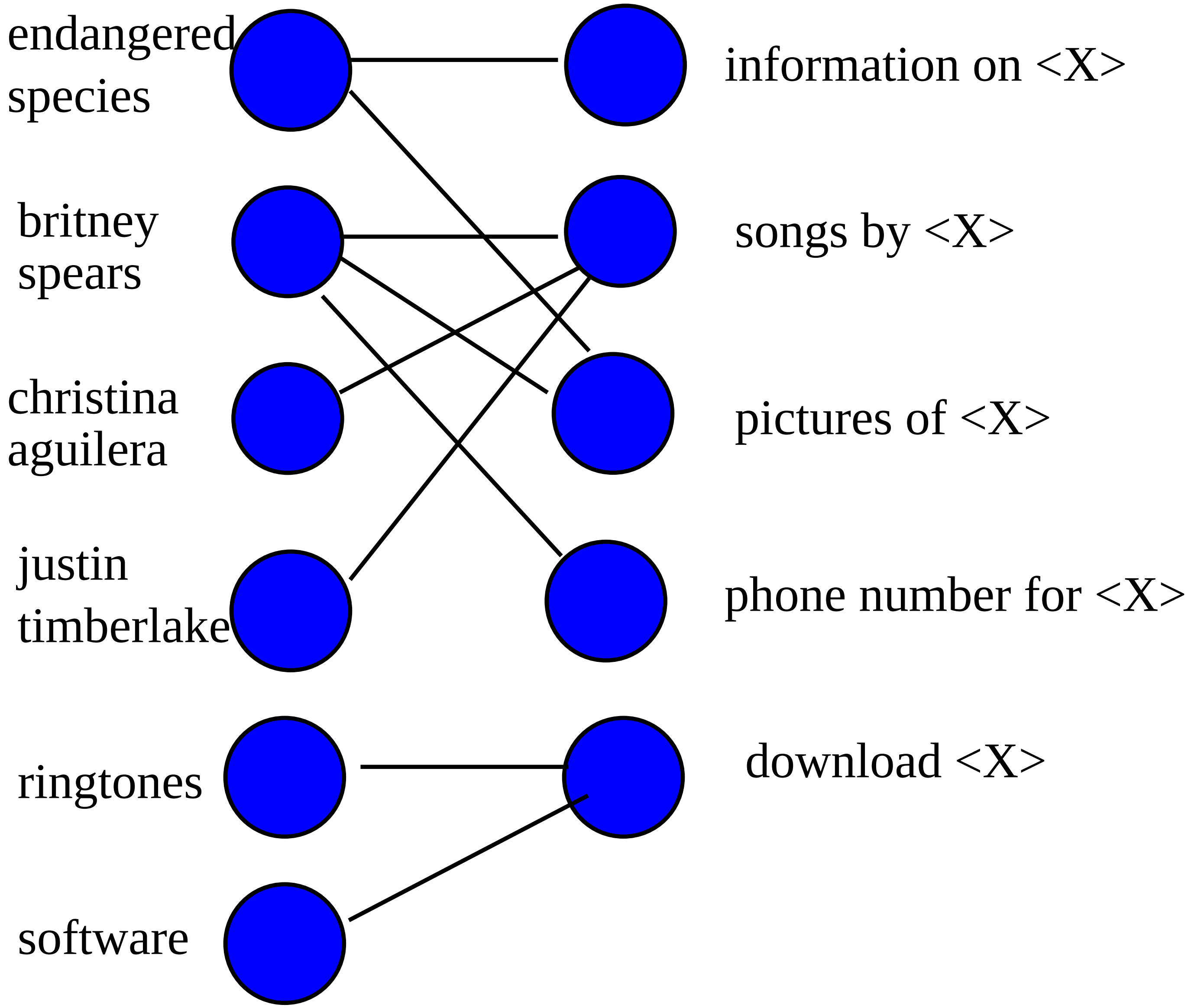
	nps	nps (non-seed)	Examples	Examples (non-seed)
P@1	1	1.	1	0
P@10	1	0.7	1	0.4
P@50	0.96	0.64	1	0.54
P@100	0.96	0.54	0.78	0.55
P@200	0.97	0.36	0.70	0.53

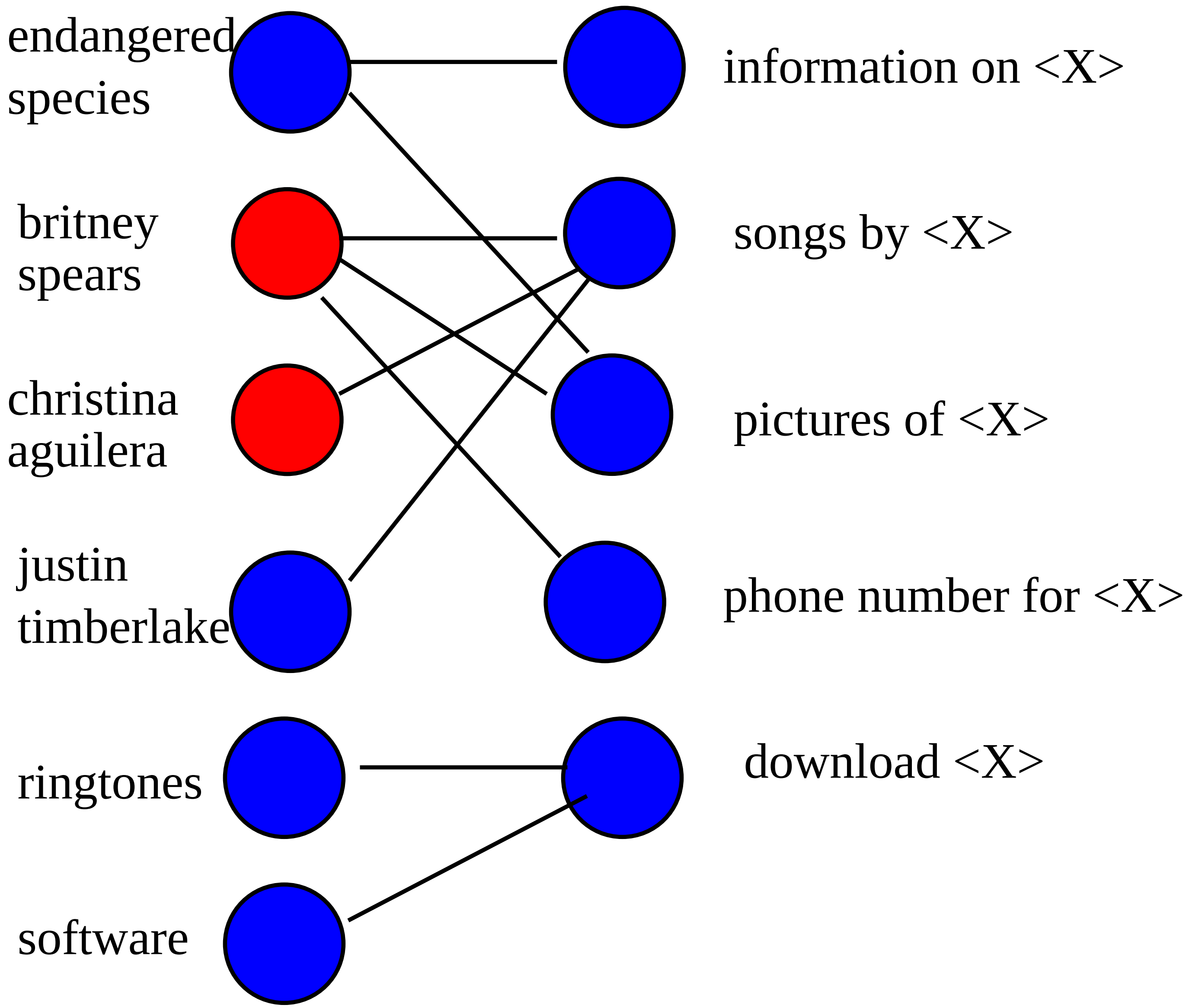
Seeds = services, software, products
Active learning = feature-set disagreement, 100 labeled

Other Potential Applications of this Work

Web search queries also exhibit regular grammatical structure

- verb + object
- np + pp


endangered
species
britney
spears
christina
aguilera
justin
timberlake
ringtones
software
information on <X>
songs by <X>
pictures of <X>
phone number for <X>
download <X>


endangered species
britney spears
christina aguilera
justin timberlake
ringtones
software
information on <X>
songs by <X>
pictures of <X>
phone number for <X>
download <X>

Contributions Summary

- In-depth experiments with bootstrapping algorithms across multiple semantic classes.
- Adapted existing semi-supervised learning algorithms for the task of information extraction.
- Novel active learning algorithms that take into account the feature set split into two sets.
- Analysis of the noun-phrase context co-occurrence graph to show that it exhibits small-world and power-law structure.
- Demonstration of the correlation between graph features and algorithm performance