10-709: Read The Web

Tom Mitchell

January 2006

Learn to Read / Read to Learn

Thesis: We can achieve a breakthrough in NLP by building a continuously learning, continuously reading system, targeted toward understanding and extracting 80% of the factual content on the internet

Why now?

1. Recent progress in NLP
2. Recent progress in statistical machine learning
   - Especially bootstrapping methods that leverage *redundancy*
3. The web provides huge corpus of highly *redundant* text

The Idea

- Build on existing components
  - Named entity extractors, question answerers, parsers, coreference resolvers, ...
  - Self-supervised learning algorithms
  - Knowledge representations, ontologies, KBs, ...
- Create agent that formulates and pursues an infinite stream of learning/reading/fact acquisition subgoals
- Learn to read / Read to learn
- Primarily unsupervised (self-supervised)

Design goals for ReadTheWeb system

- Nonstop 24x7 operation, pursuing two goals:
  - Learning to read
  - Reading the web

- Begin with state-of-the-art methods (NLP, learning, representation)

- Architecture for improving continuously
  - A growing knowledge base (with pointers back to text sources)
  - A growing ability to understand complex text (and non-text)

- <1 day barrier to entry for researchers

Design of the course

- Become experts in state of the art of semi-supervised learning for NLP

- Design, implement, experiment with, and write up a first ReadTheWeb system

- First 4 weeks: each team implements working semi-supervised learner, for some aspect of NLP
- Next 8 weeks: we design and implement integrated system
- All 13 weeks: cover state-of-art research papers

What we'll build on

- State of the art semi-supervised learning and NLP algorithms
- Existing software
  - Knowledge repository (SCONE)
  - Text learning package (Minor Third)
  - Text annotation framework (UIMA)
  - Web crawl / web query engine
- Your expertise, creativity and hard work

Course Logistics/Details

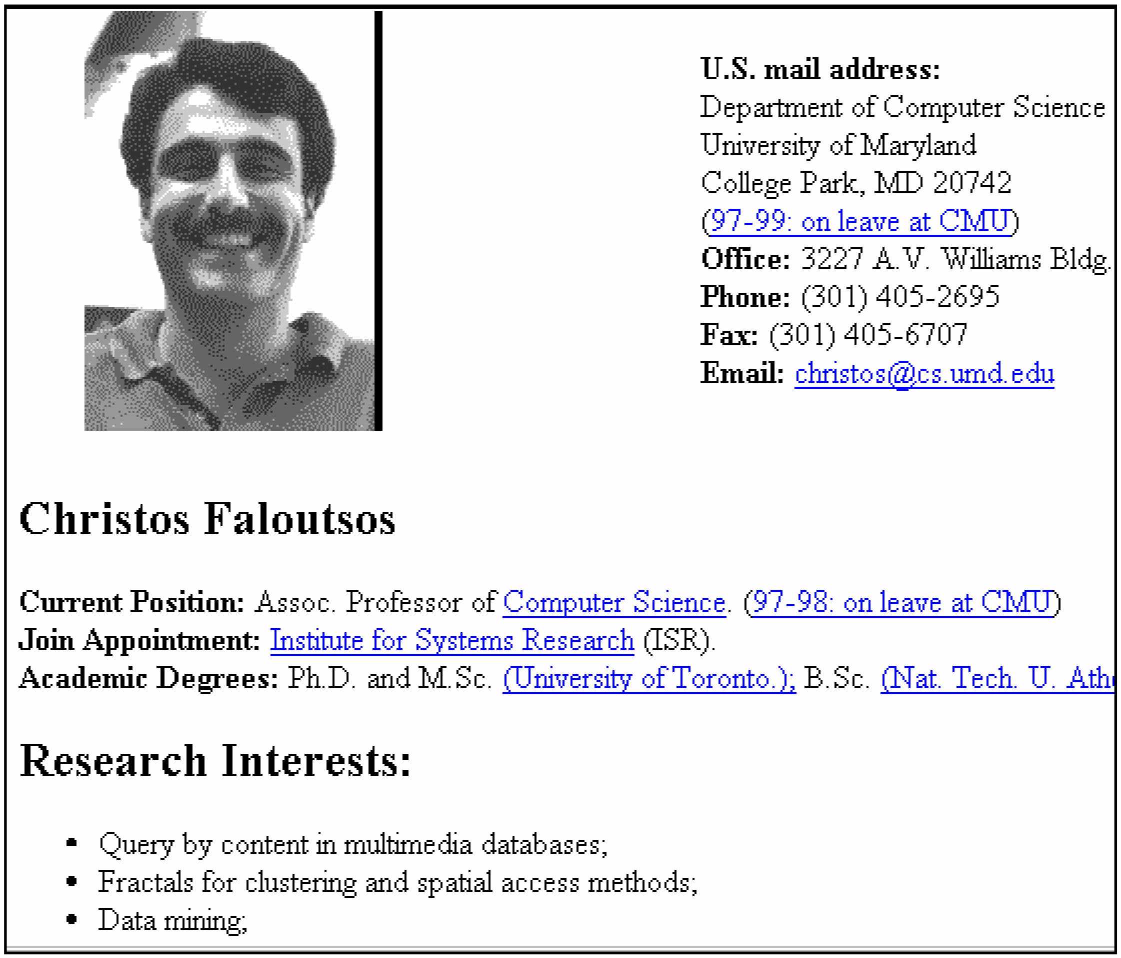
- This is a research project disguised as a course

- This will be hard work, and fun
- Some guest lectures (e.g., Oren Etzioni, Feb 9)
- No exams
- Grading based on projects and course participation
- Course web site will appear by tomorrow, off http://www.cs.cmu.edu/~tom

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

Research Interests:

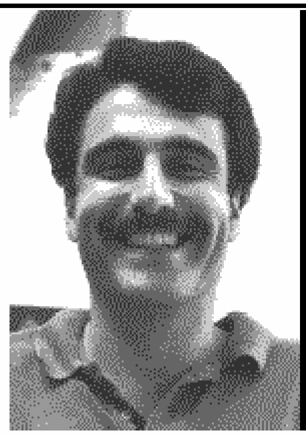
- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features

Professor Faloutsos

my advisor

Redundantly Sufficient Features



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

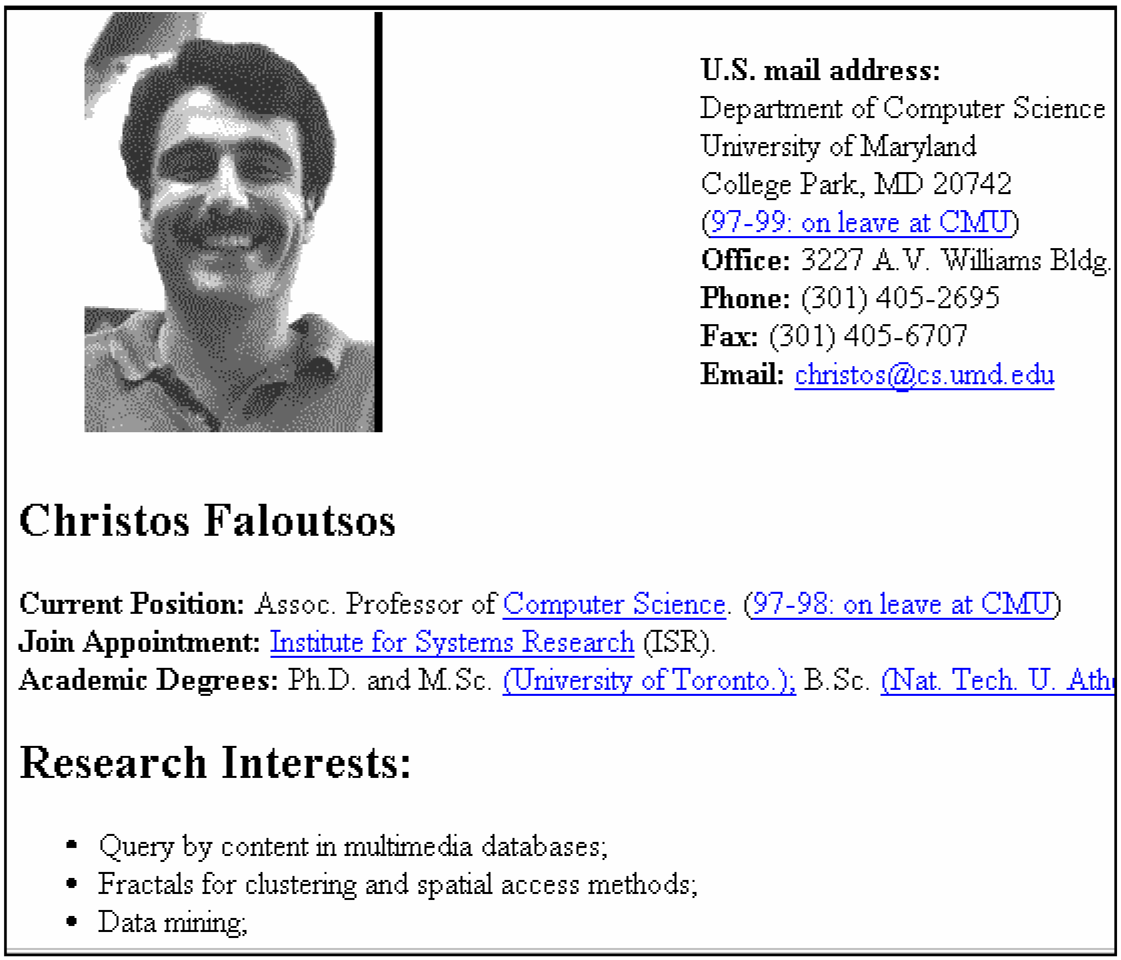
Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

CoTraining Algorithm #1

[Blum&Mitchell, 1998]

Given: labeled data L,

unlabeled data U

Loop:

Train g1 (hyperlink classifier) using L

Train g2 (page classifier) using L

Allow g1 to label *p* positive, *n* negative examps from U

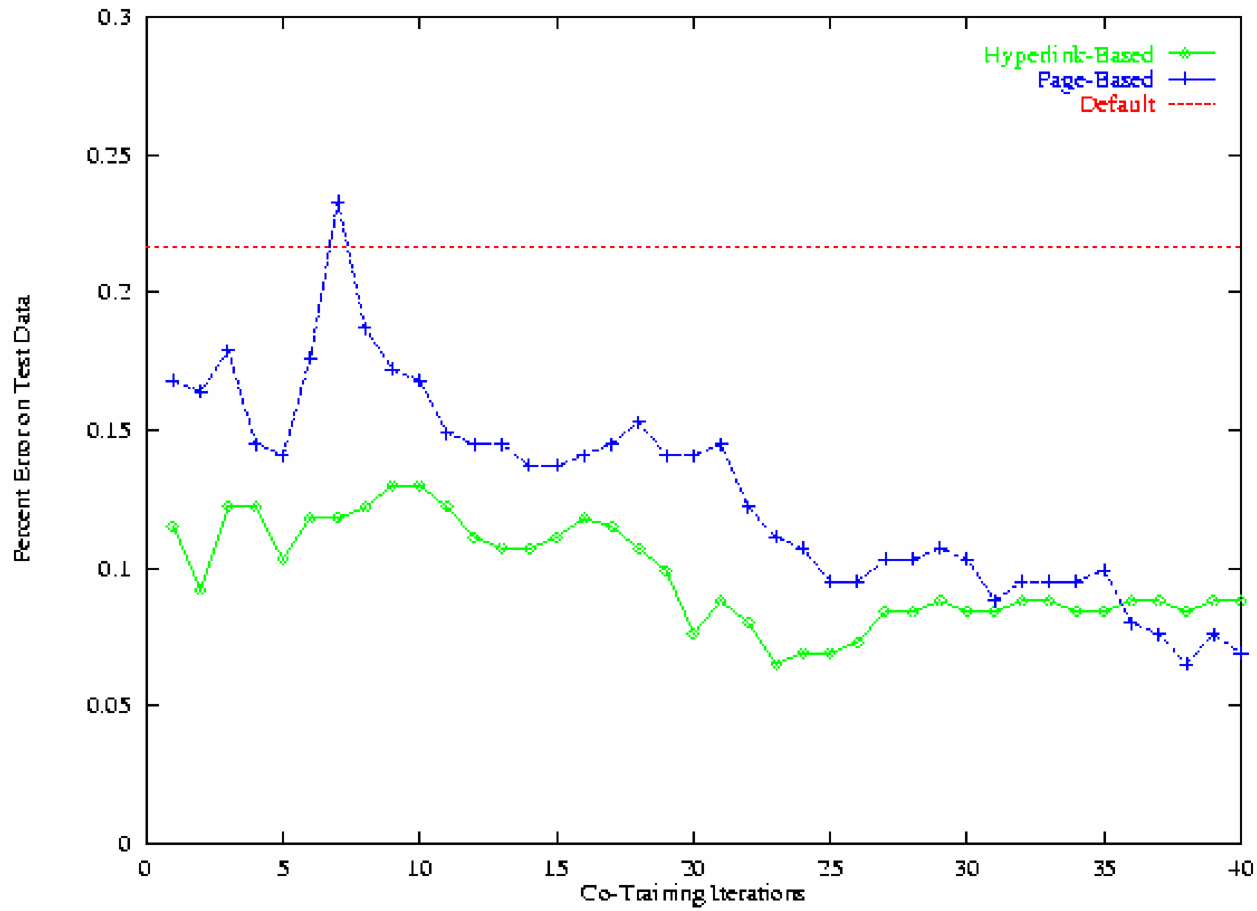
Allow g2 to label *p* positive, *n* negative examps from U

Add these self-labeled examples to L

CoTraining: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: cotraining 5.0%

Typical run:



CoTraining setting:

- wish to learn f: X → Y, given L and U drawn from P(X,Y)
- features describing X can be partitioned (X = X1 x X2)

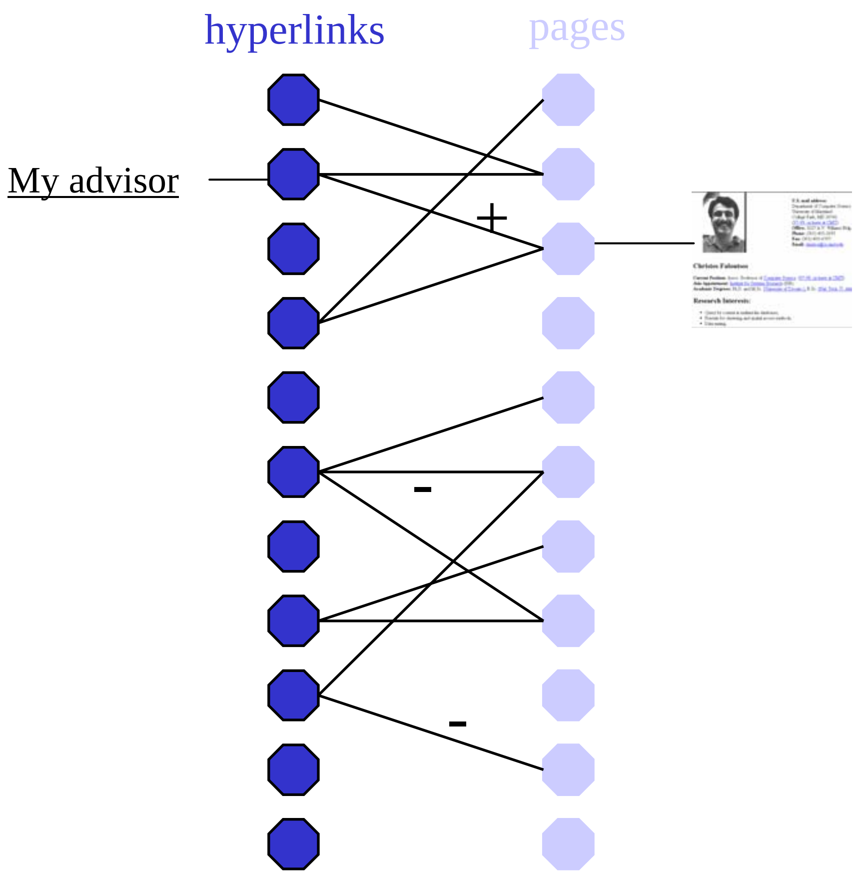
  such that f can be computed from either X1 or X2

$$(\exists g_1, g_2)(\forall x \in X) \quad g_1(x_1) = f(x) = g_2(x_2)$$

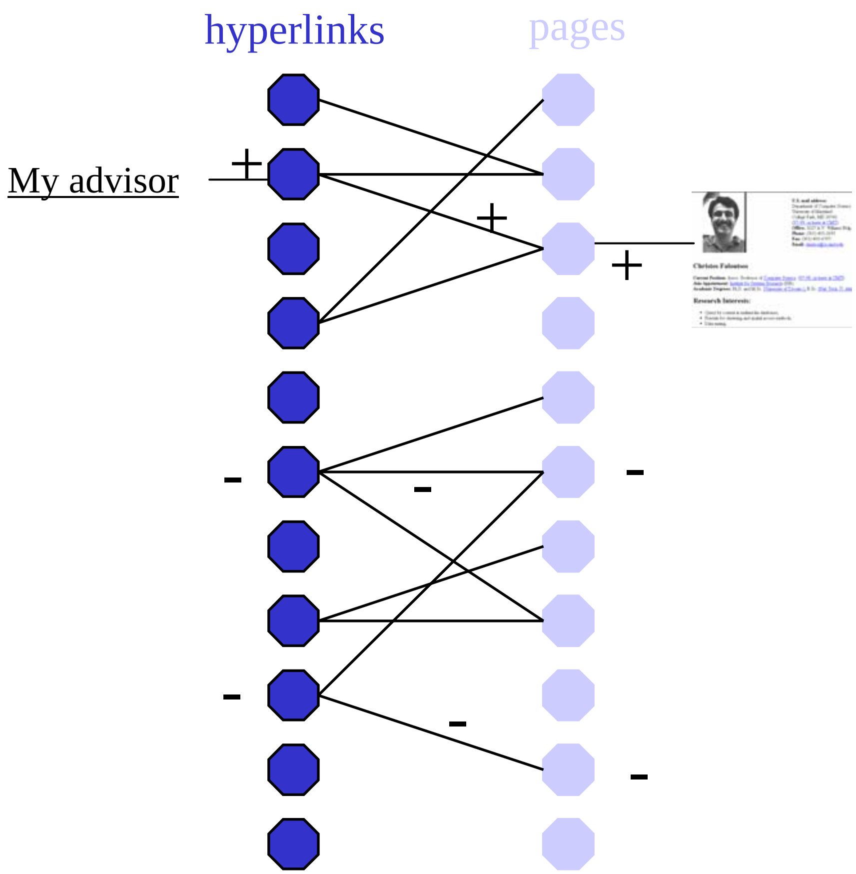
One result [Blum&Mitchell 1998]:

- If
  - X1 and X2 are conditionally independent given Y
  - f is PAC learnable from noisy *labeled* data
- Then
  - f is PAC learnable from weak initial classifier plus *unlabeled* data

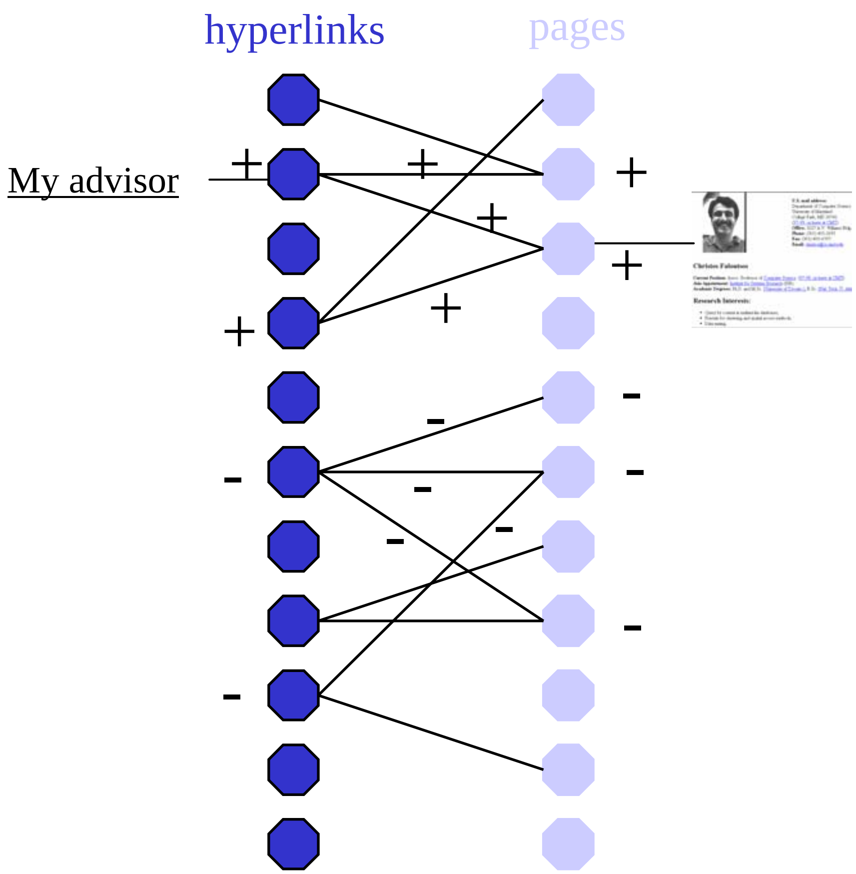
Co-Training Rote Learner



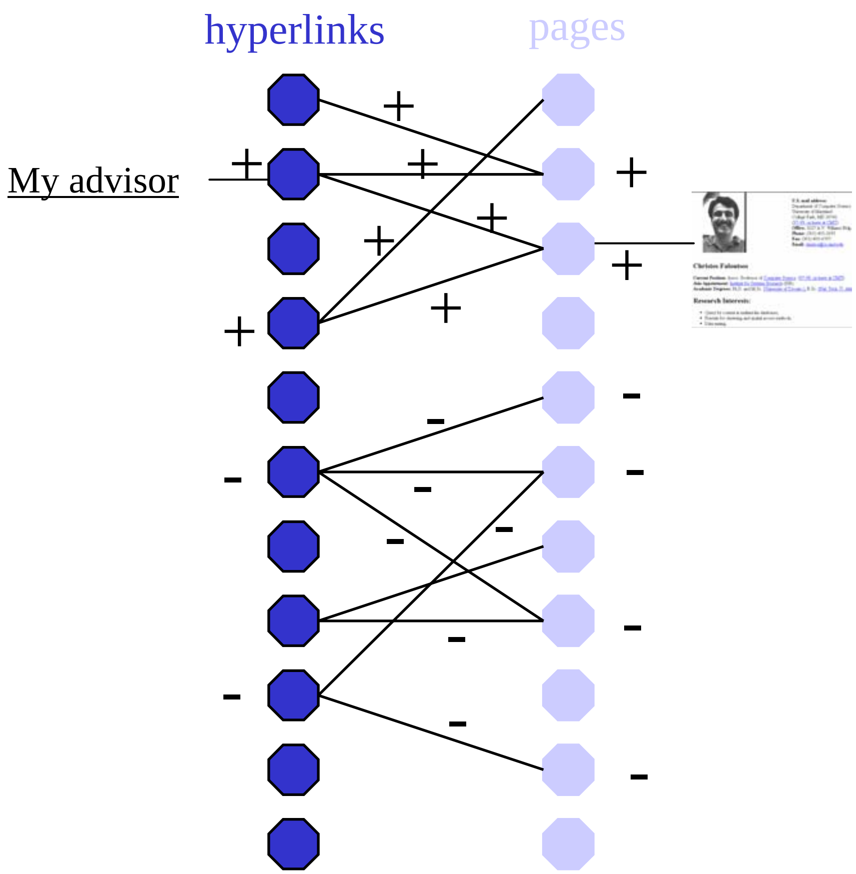
Co-Training Rote Learner



Co-Training Rote Learner



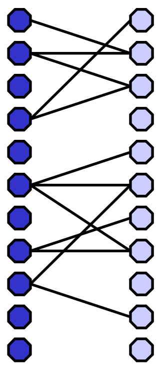
Co-Training Rote Learner



Expected Rote CoTraining error given *m* labeled examples, rote learning, perfectly redundantly sufficient

$$CoTraining \quad setting:$$
$$learn \quad f: X \rightarrow Y$$
$$where \quad X = X_1 \times X_2$$
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$
$$and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)$$

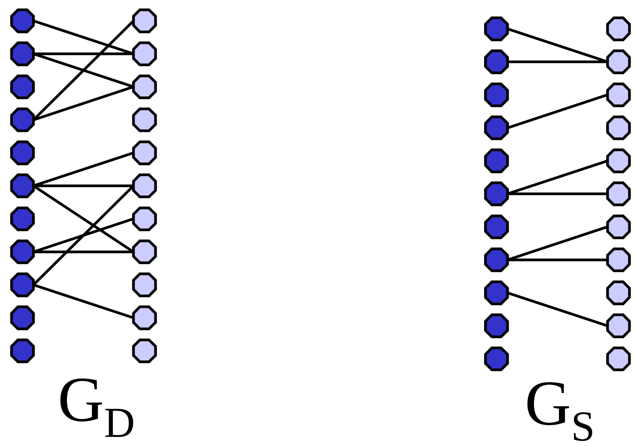
$$E[error] = \sum_j P(x \in g_j)(1 - P(x \in g_j))^m$$



Where g_j is the jth connected component of graph of L+U, m is number of labeled examples

How many *unlabeled* examples suffice?

Want to assure that connected components in the underlying distribution, G_D, are connected components in the observed sample, G_S

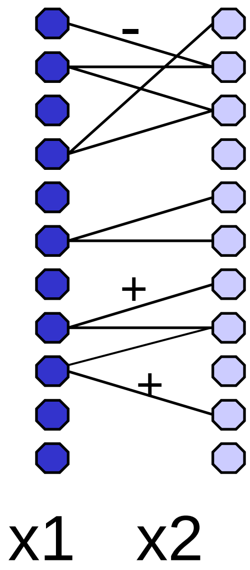


O(log(N)/α) examples assure that with high probability, G_S has same connected components as G_D [Karger, 94]

N is size of G_D, α is min cut over all connected components of G_D

Co Training

- What's the best-case graph? (most benefit from unlabeled data)
- What the worst case?
- What does conditional-independence imply about graph?



PAC Generalization Bounds on CoTraining

[Dasgupta et al., NIPS 2001]

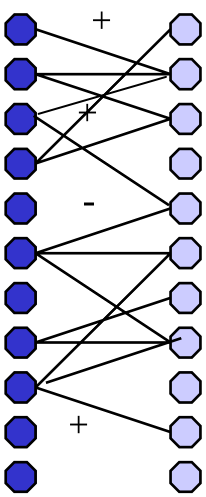
This theorem assumes X1 and X2 are conditionally independent given Y

Theorem 1 *With probability at least* $1 - \delta$ *over the choice of the sample* S, *we have that for all* h_1 *and* h_2, *if* $\gamma_i(h_1, h_2, \delta) > 0$ *for* $1 \leq i \leq k$ *then (a)* f *is a permutation and (b) for all* $1 \leq i \leq k$,

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp) \leq \frac{\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) + \epsilon_i(h_1, h_2, \delta)}{\gamma_i(h_1, h_2, \delta)}.$$

The theorem states, in essence, that if the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp)$ is a good estimate of the error rate $P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp)$.

What if CoTraining Assumption Not Perfectly Satisfied?



- Idea: Want classifiers that produce a *maximally consistent* labeling of the data
- If learning is an optimization problem, what function should we optimize?

What Objective Function?

$$E = E1 + E2 + c_3 E3 + c_4 E4$$

Error on labeled examples

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

Disagreement over unlabeled

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Misfit to estimated class priors

$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y>\in L} y \right) - \left(\frac{1}{|L| + |U|} \sum_{x\in L\cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

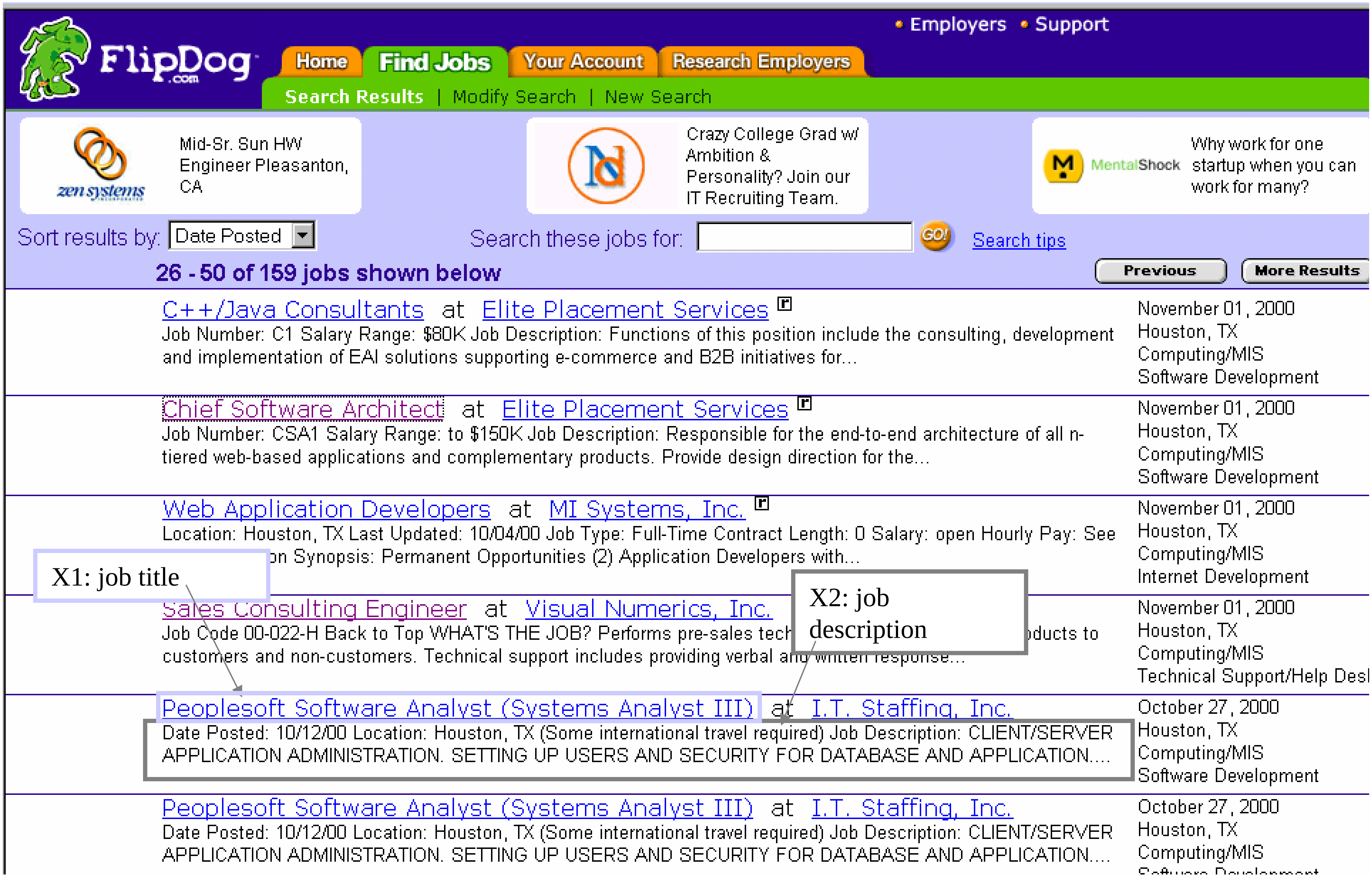
What Function Approximators?

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}}$$

$$\hat{g}_2(x) = \frac{1}{1 + e^{\sum_j w_{j,2} x_j}}$$

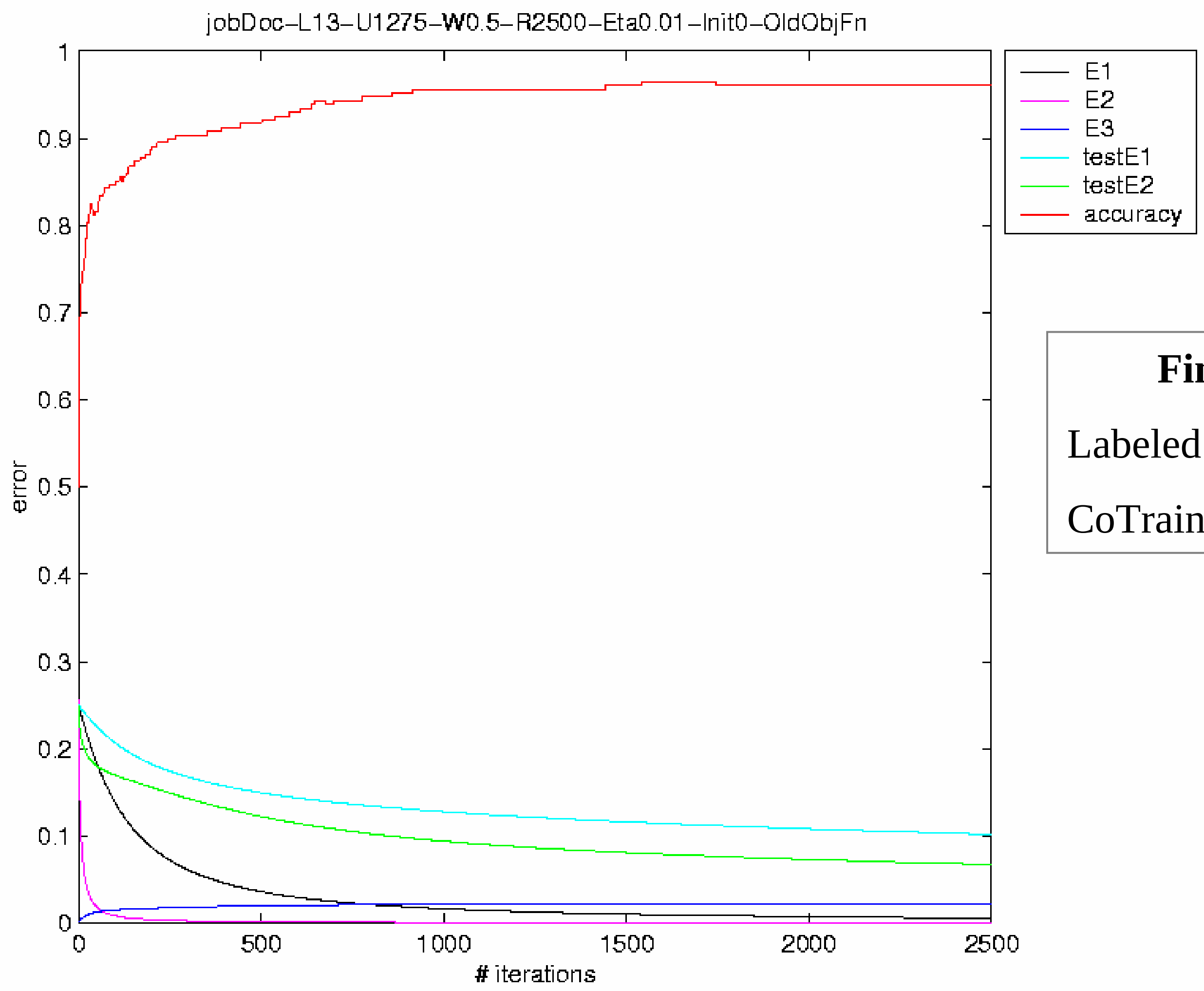
- Same functional form as logistic regression
- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4
- No word independence assumption, use both labeled and unlabeled data

Classifying Jobs for FlipDog



Gradient CoTraining

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoTraining: 96%

Gradient CoTraining

Classifying Capitalized sequences as Person Names

Eg., "Company president <u>Mary Smith</u> said today…"

x1 x2 x1

Error Rates

	25 labeled 5000 unlabeled	***2300 labeled 5000 unlabeled***
Using labeled data only	.24	.13
Cotraining	.15 *	.11 *
Cotraining without fitting class priors (E4)	.27 *	

* Quite sensitive to weights of error terms E3 and E4

Co-EM [Nigam & Ghani, 2000]

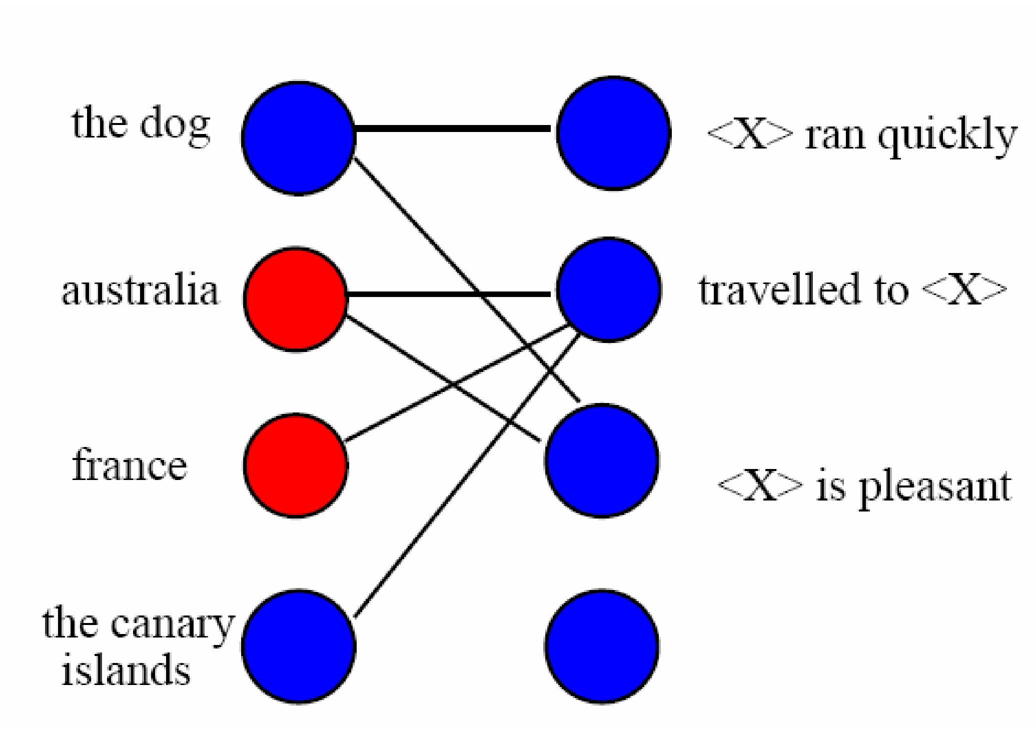
Idea:

- Like co-training, use one set of features to label the other
- Like EM, iterate
  - Assigning probabilistic values to unobserved class labels
  - Updating model parameters (= labels of other feature set)

$$
\begin{aligned}
P(class|context_i) &= \sum_j P(class|NP_j)P(NP_j|context_i) \\
P(class|NP_i) &= \sum_j P(class|context_j)P(context_j|NP_i)
\end{aligned}
$$

CoEM applied to Named Entity Recognition

[Rosie Jones, 2005], [Ghani & Nigam, 2000]



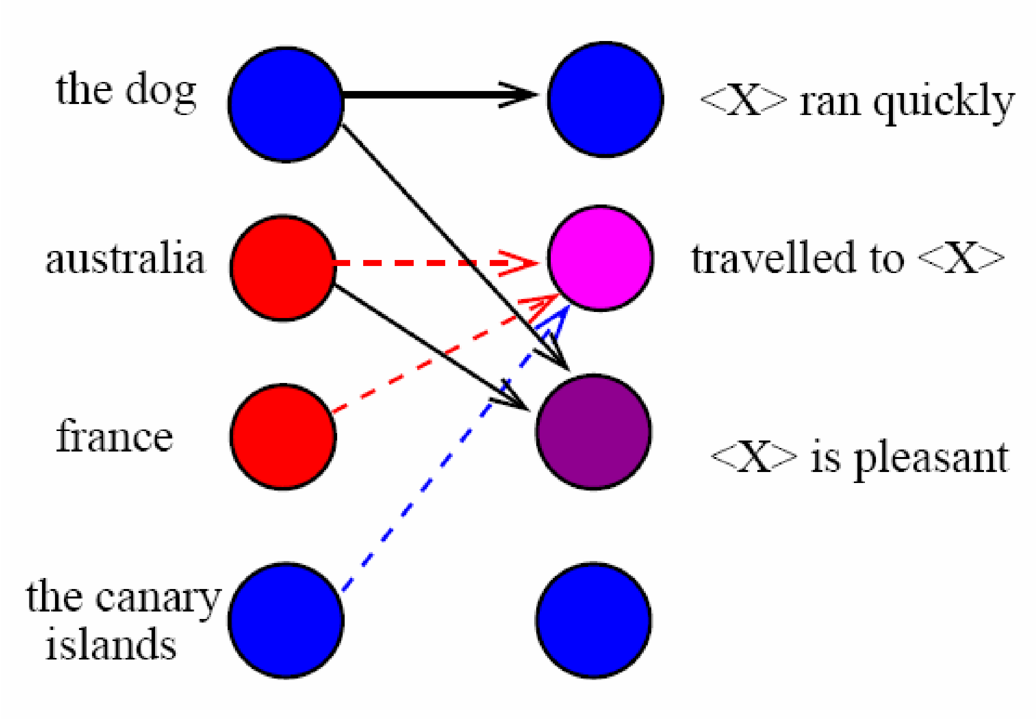
Update rules:

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

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

CoEM applied to Named Entity Recognition

[Rosie Jones, 2005], [Ghani & Nigam, 2000]



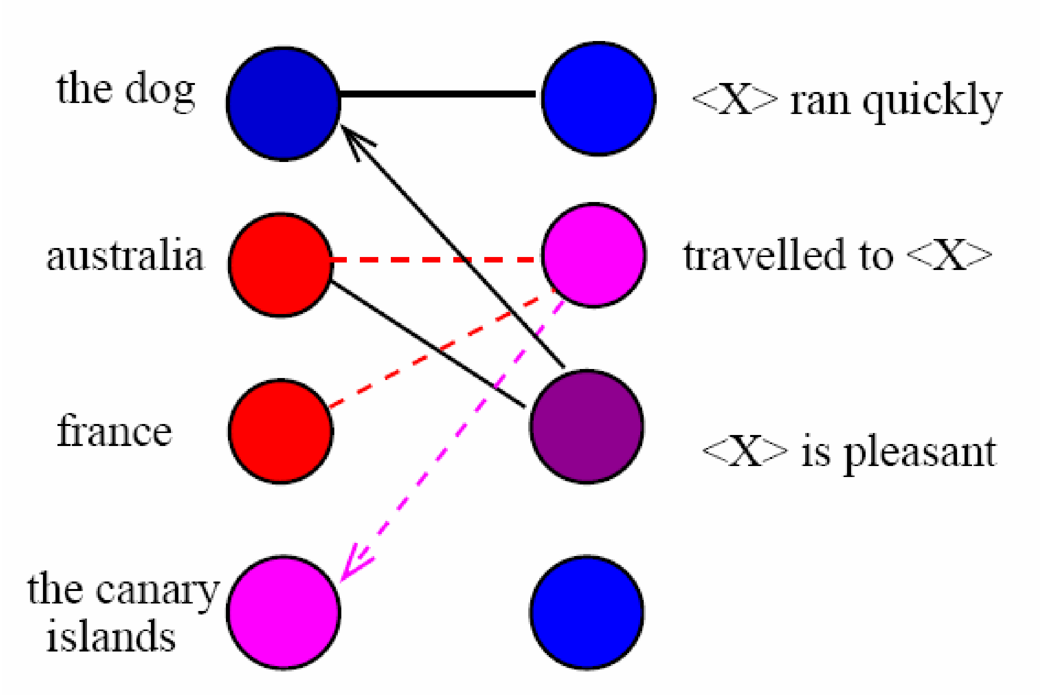
Update rules:

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

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

CoEM applied to Named Entity Recognition

[Rosie Jones, 2005], [Ghani & Nigam, 2000]

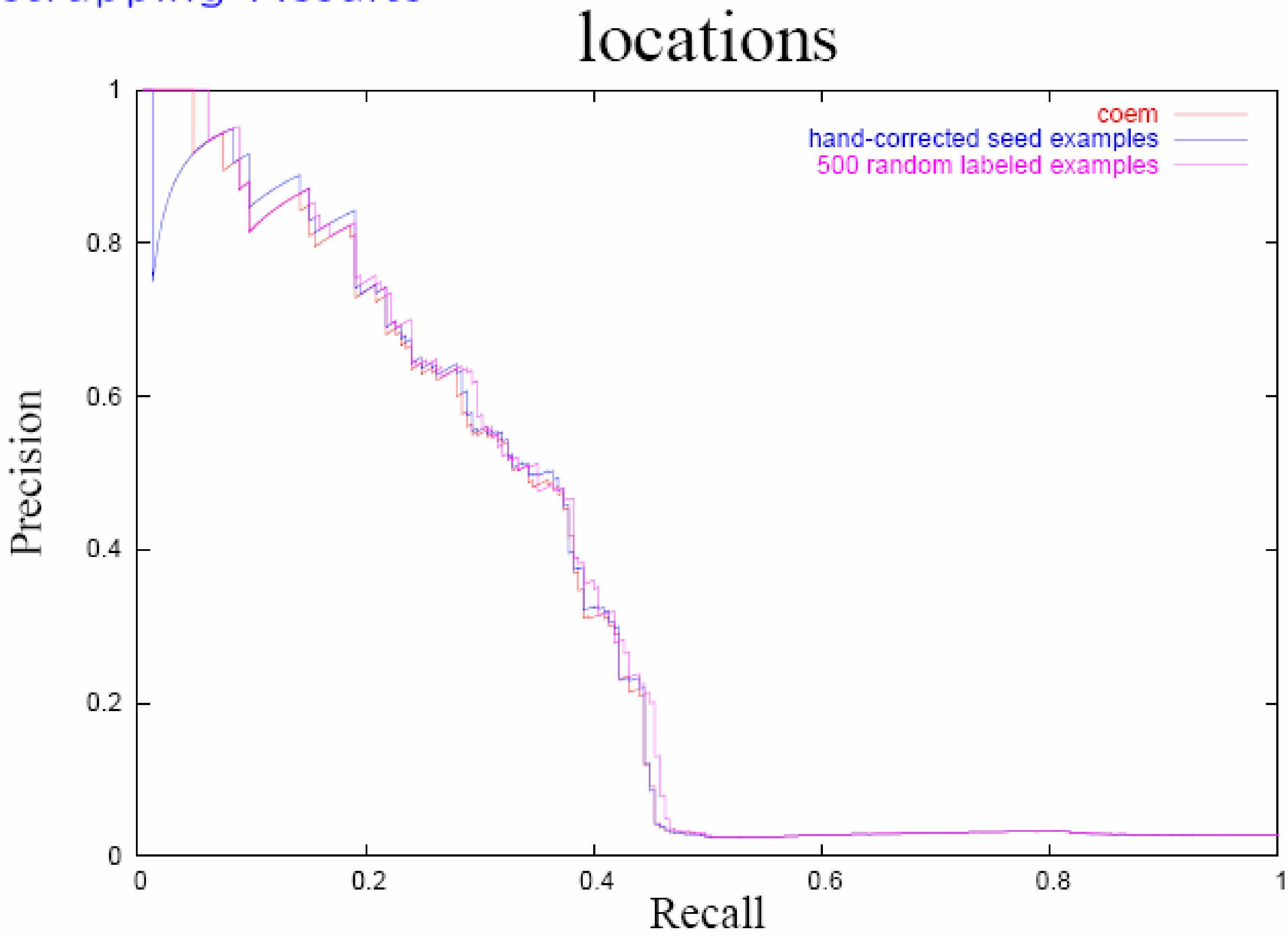


Update rules:

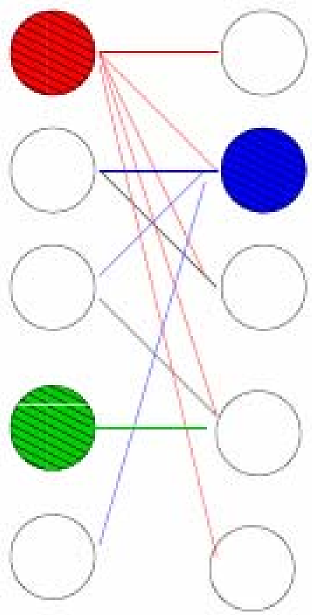
$$P(class|context_i) = \sum_j P(class|NP_j)P(NP_j|context_i)$$

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

Bootstrapping Results



Some nodes are more important than others [Jones, 2005]

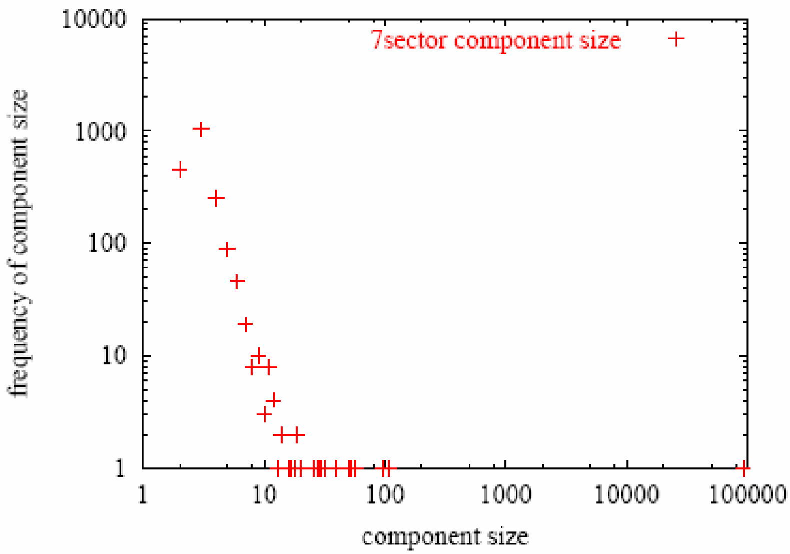


Can use this for active learning...

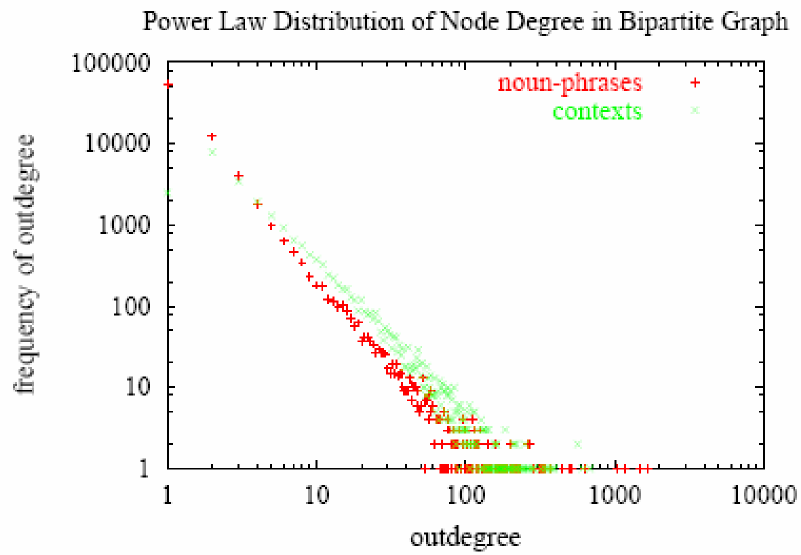
Noun-phrase	Outdegree
you	1656
we	1479
it	1173
company	1043
this	635
all	520
they	500
information	448
us	367
any	339
products	332
i	319
site	314
one	311
1996	282
he	269
customers	269
these	263
them	263
time	234

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



Node Degree is Power-Law Distributed



[Jones, 2005]

CoTraining Summary

- Unlabeled data improves supervised learning when example features are redundantly sufficient
  - Family of algorithms that train multiple classifiers

- Theoretical results

- Many real-world problems of this type
  - Semantic lexicon generation [Riloff, Jones 99], [Collins, Singer 99]
  - Web page classification [Blum, Mitchell 98]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]
  - Visual classification of cars [Levin, Viola, Freund 03]

Bootstrapping: Learning to extract named entities

location?

I arrived in Beijing on Saturday.

x_1: I arrived in ______ on Saturday.

x_2: Beijing

Example 3: Word sense disambiguation [Yarowsky]

- “bank” = river bank, or financial bank??

- Assumes a single word sense per document
  - X1: the document containing the word
  - X2: the immediate context of the word (‘swim near the __’)

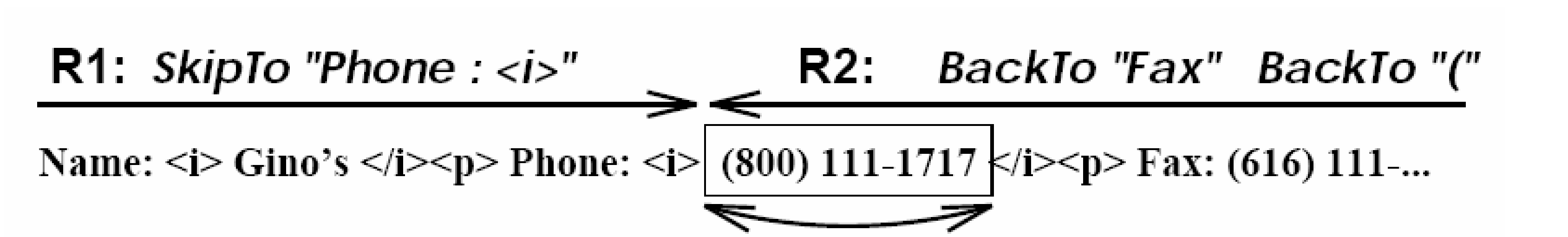
Successfully learns “context → word sense” rules when word occurs multiples times in document.

Example 4: Bootstrap learning for IE from HTML structure

[Muslea, et al. 2001]

X_1: HTML preceding the target

X_2: HTML following the target



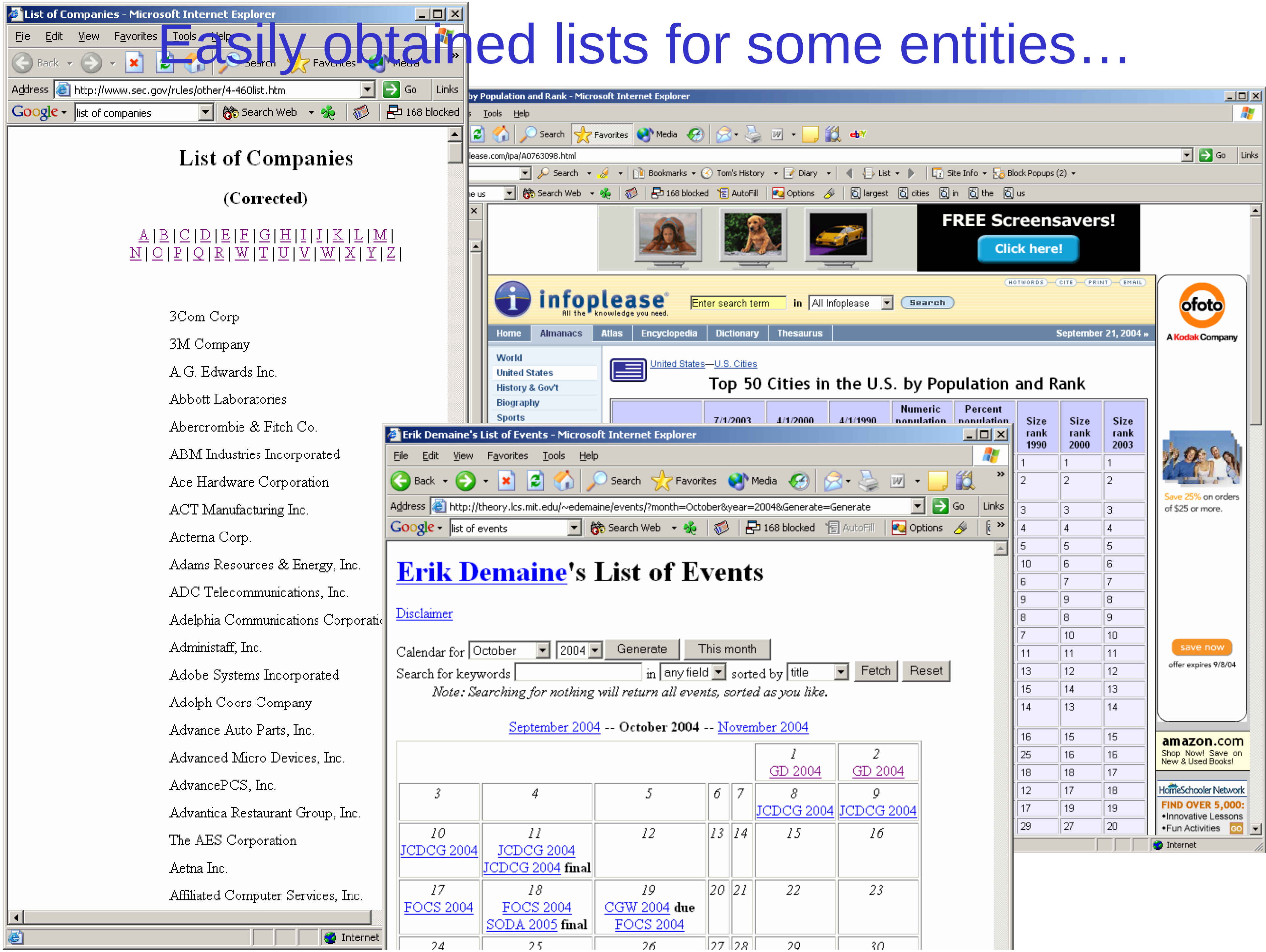
Example Bootstrap learning algorithms:

- Classifying web pages [Blum&Mitchell 98; Slattery 99]
- Classifying email [Kiritchenko&Matwin 01; Chan et al. 04]
- Named entity extraction [Collins&Singer 99; Jones&Riloff 99]
- Wrapper induction [Muslea et al., 01; Mohapatra et al. 04]
- Word sense disambiguation [Yarowsky 96]
- Discovering new word senses [Pantel&Lin 02]
- Synonym discovery [Lin et al., 03]
- Relation extraction [Brin et al.; Yangarber et al. 00]
- Statistical parsing [Sarkar 01]

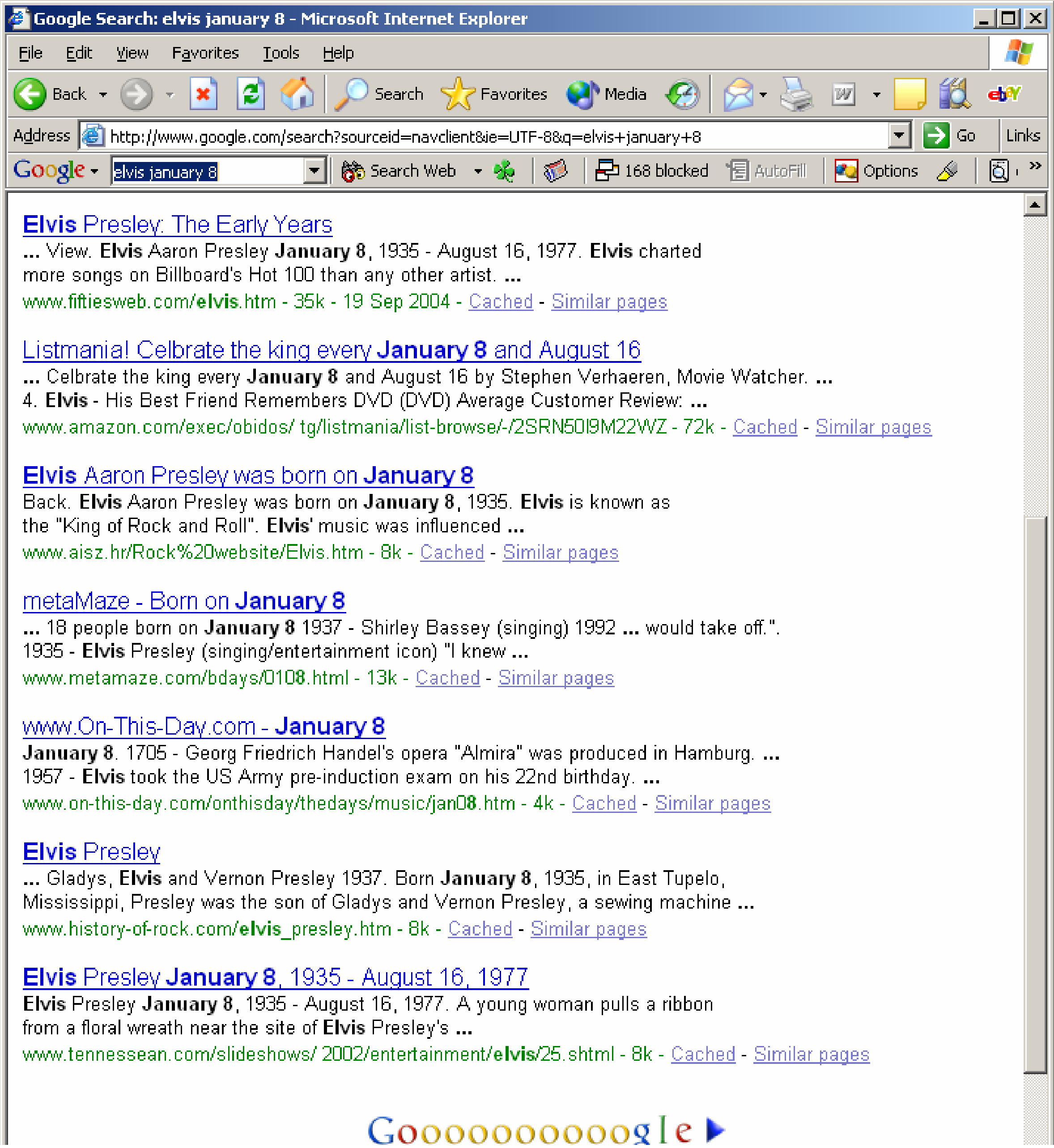
Many Exploitable Redundancies

- Hyperlink words, web page words
  - (page classification, hyperlink word sense)
- Email subject line, email body
  - (email classification)
- Statements of same fact on *many* different websites
  - EventDateIs(ElvisBirthday, January 28)
- Assertions in both text, and tables
  - Semi-structured HTML
  - Excel spreadsheets
- Directory names, directory contents
- Activity clusters from email text, or social network
- Calendar events, email before and after meeting
- Deductive inference, when knowledge available

Easily obtained lists for some entities…



What is relation between “Elvis” and “January 8”?


Google Search: elvis january 8 - Microsoft Internet Explorer

Address http://www.google.com/search?sourceid=navclient&ie=UTF-8&q=elvis+january+8

Elvis Presley: The Early Years
... View. **Elvis** Aaron Presley **January 8**, 1935 - August 16, 1977. **Elvis** charted more songs on Billboard's Hot 100 than any other artist. ...
www.fiftiesweb.com/**elvis**.htm - 35k - 19 Sep 2004 - Cached - Similar pages

Listmania! Celbrate the king every **January 8** and August 16
... Celbrate the king every **January 8** and August 16 by Stephen Verhaeren, Movie Watcher. ...
4. **Elvis** - His Best Friend Remembers DVD (DVD) Average Customer Review: ...
www.amazon.com/exec/obidos/ tg/listmania/list-browse/-/2SRN50I9M22WZ - 72k - Cached - Similar pages

Elvis Aaron Presley was born on **January 8**
Back. **Elvis** Aaron Presley was born on **January 8**, 1935. **Elvis** is known as the "King of Rock and Roll". **Elvis**' music was influenced ...
www.aisz.hr/Rock%20website/Elvis.htm - 8k - Cached - Similar pages

metaMaze - Born on **January 8**
... 18 people born on **January 8** 1937 - Shirley Bassey (singing) 1992 ... would take off.".
1935 - **Elvis** Presley (singing/entertainment icon) "I knew ...
www.metamaze.com/bdays/0108.html - 13k - Cached - Similar pages

www.On-This-Day.com - **January 8**
January 8. 1705 - Georg Friedrich Handel's opera "Almira" was produced in Hamburg. ...
1957 - **Elvis** took the US Army pre-induction exam on his 22nd birthday. ...
www.on-this-day.com/onthisday/thedays/music/jan08.htm - 4k - Cached - Similar pages

Elvis Presley
... Gladys, **Elvis** and Vernon Presley 1937. Born **January 8**, 1935, in East Tupelo, Mississippi, Presley was the son of Gladys and Vernon Presley, a sewing machine ...
www.history-of-rock.com/**elvis**_presley.htm - 8k - Cached - Similar pages

Elvis Presley **January 8**, 1935 - August 16, 1977
Elvis Presley **January 8**, 1935 - August 16, 1977. A young woman pulls a ribbon from a floral wreath near the site of **Elvis** Presley's ...
www.tennessean.com/slideshows/ 2002/entertainment/**elvis**/25.shtml - 8k - Cached - Similar pages

Gooooooooooogle

Some agent strategies for generating tasks

- Collect more data from web
  - To learn about specific entities (e.g., "Rolling Stones")
  - To learn meaning of particular language (e.g., "will attend")
  - To locate easy-to extract facts (e.g., web pages with lists)

- Learn regularities from the populated KB
  - "Most LTI office names are of the form "NSH dddd"

- Explore specializations of ontological categories
  - What distinguishes personal home pages that contain publications from those that don't? Can this be predicted from other (extractable) features of the home page?

- Explore specializations of language structures
  - Which 'location' entities share surrounding language? e.g., "the city of ?x," Do they share other properties?

Some Types of Knowledge to Learn

- Linguistic regularities
  - {“spoon”,”fork”,”chopsticks”} occur often in “eat with my ___”
- HTML layout regularities
  - HTML lists often contain items of the same type
- Web site regularities
  - University departments often have a page listing all faculty
- Regularities over extracted facts
  - ‘Professors typically have more publications than their advisees’
- Temporal stability
  - Birthdays don’t change. Stock prices do.