6.864 (Fall 2007)

Word-Sense Disambiguation, and Semi-Supervised Learning

Overview

- A supervised method for word-sense disambiguation: decision lists
- A semi-supervised method for word-sense disambiguation
- A semi-supervised method for named-entity classification

Words in Context

Sense	Examples (keyword in context)
1	. . . used to strain microscopic plant life from the . . .
1	. . . too rapid growth of aquatic plant life in water . . .
2	. . . automated manufacturing plant in Fremont . . .
2	. . . discovered at a St. Louis plant manufacturing . . .

- **The task:** given a word in context, decide on its word sense

Examples

Examples of words used in [Yarowsky, 1995]:

Word	Senses
plant	living/factory
tank	vehicle/container
poach	steal/boil
palm	tree/hand
axes	grind/tools
sake	benefit/drink
bass	fish/music
space	volume/outer
motion	legal/phsyical
crane	bird/machine

Features Used in the Model

- Word found in $+/-k$ word window
- Word immediately to the right (+1 W)
- Word immediately to the left (-1 W)
- Pair of words at offsets -2 and -1
- Pair of words at offsets -1 and +1
- Pair of words at offsets +1 and +2

Features Used in the Model

- Also maps words to parts of speech, and general classes (e.g., WEEKDAY, MONTH etc.)
- Local features including word classes are added:
  - Pair of tags at offsets -2 and -1
  - Tag at position -2, word at position -1
  - etc.

An Example

The ocean reflects the color of the sky, but even on cloudless days the color of the ocean is not a consistent blue. Phytoplankton, microscopic plant life that floats freely in the lighted surface waters, may alter the color of the water. When a great number of organisms are concentrated in an area, the plankton changes the color of the ocean surface. This is called a 'bloom.'

$\Downarrow$

w_{-1} = Phytoplankton	t_{-1} = JJ
w_{+1} = life	t_{+1} = NN
w_{-2}, w_{-1} = (Phytoplankton,microscopic)	t_{-2}, t_{-1} = (NN,JJ)
w_{-1}, w_{+1} = (microscopic,life)	...
w_{+1}, w_{+2} = (life,that)	
word-within-k = ocean	
word-within-k = reflects	
word-within-k = color	
...	
word-within-k = bloom	

A Machine-Learning Method: Decision Lists

- For each feature, we can get an estimate of conditional probability of sense 1 and sense 2
- For example, take the feature $w_{+1} = \texttt{life}$
- We might have

$$\begin{aligned} Count(\text{sense 1 of plant}, w_{+1} = \texttt{life}) &= 100 \\ Count(\text{sense 2 of plant}, w_{+1} = \texttt{life}) &= 1 \end{aligned}$$

- Maximum-likelihood estimate

$$P(\text{sense 1 of plant} \mid w_{+1} = \texttt{life}) = \frac{100}{101}$$

Smoothed Estimates

- Usual problem: some counts are sparse

- We might have

$$Count(\text{sense 1 of plant}, w_{-1} = \texttt{Phytoplankton}) = 2$$
$$Count(\text{sense 2 of plant}, w_{-1} = \texttt{Phytoplankton}) = 0$$

- α smoothing (empirically, $\alpha \approx 0.1$ works well):

$$P(\text{sense 1 of plant} \mid w_{-1} = \texttt{Phytoplankton}) = \frac{2+\alpha}{2+2\alpha}$$
$$P(\text{sense 1 of plant} \mid w_{+1} = \texttt{life}) = \frac{100+\alpha}{101+2\alpha}$$

with $\alpha = 0.1$, gives values of 0.95 and 0.99 (unsmoothed gives values of 1 and 0.99)

Creating a Decision List

- For each feature, find

$$sense(feature) = \text{argmax}_{sense} P(sense \mid feature)$$

e.g., $sense(w_{+1} = \texttt{life}) = sense1$

- Create a rule `feature` $\rightarrow sense(feature)$ with weight $P(sense(feature) \mid feature)$. e.g.,

Rule			Weight
$w_{+1} =$`life`	$\rightarrow$	sense 1	0.99
$w_{-1} =$`Phytoplankton`	$\rightarrow$	sense 1	0.95
...			

Creating a Decision List

- Create a list of rules sorted by strength

Rule			Weight
$w_{+1} =$`life`	$\rightarrow$	sense 1	0.99
$w_{-1} =$`manufacturing`	$\rightarrow$	sense 2	0.985
`word-within-k=life`	$\rightarrow$	sense 1	0.98
`word-within-k=manufacturing`	$\rightarrow$	sense 2	0.979
`word-within-k=animal`	$\rightarrow$	sense 1	0.975
`word-within-k=equipment`	$\rightarrow$	sense 2	0.97
`word-within-k=employee`	$\rightarrow$	sense 2	0.968
$w_{-1} =$`assembly`	$\rightarrow$	sense 2	0.965
...			

- To apply the decision list: take the first (strongest) rule in the list which applies to an example

The ocean reflects the color of the sky, but even on cloudless days the color of the ocean is not a consistent blue. Phytoplankton, microscopic plant life that floats freely in the lighted surface waters, may alter the color of the water. When a great number of organisms are concentrated in an area, the plankton changes the color of the ocean surface. This is called a 'bloom.'

Feature	Sense	Strength
w_{-1} = Phytoplankton	1	0.95
w_{+1} = life	1	0.99
w_{-2}, w_{-1} = (Phytoplankton,microscopic)	N/A	
w_{-1}, w_{+1} = (microscopic,life)	N/A	
w_{+1}, w_{+2} = (life,that)	1	0.96
word-within-k = ocean	1	0.93
word-within-k = reflects	N/A	
word-within-k = color	2	0.65
t_{-1} = JJ	2	0.56
t_{-2}, t_{-1} = (NN,JJ)	2	0.7
t_{+1} = NN	1	0.64
...		

- N/A $\Rightarrow$ feature has not been seen in training data
- w_{+1} = life $\rightarrow$ `Sense 1` is chosen

Experiments

- [Yarowsky, 1994] applies the method to accent restoration in French, Spanish

De-accented form	Accented form	Percentage
cesse	cesse	53%
	cessé	47%
coute	coûte	53%
	coûté	47%
cote	côté	69%
	côte	28%
	cote	3%
	coté	$<$ 1%

- Task is to recover accents on words
  - Very easy to collect training/test data
  - Very similar task to word-sense disambiguation
  - Useful for restoring accents in de-accented text, or in automatic generation of accents while typing

Overview

- A supervised method for word-sense disambiguation: decision lists
- A semi-supervised method for word-sense disambiguation
- A semi-supervised method for named-entity classification

A Partially Supervised Method

- Collecting labeled data can be **expensive**
- We'll now describe an approach that uses a small amount of labeled data, and a large amount of unlabeled data

A Key Property: Redundancy

The ocean reflects the color of the sky, but even on cloudless days the color of the ocean is not a consistent blue. Phytoplankton, microscopic plant life that floats freely in the lighted surface waters, may alter the color of the water. When a great number of organisms are concentrated in an area, the plankton changes the color of the ocean surface. This is called a 'bloom.'

$\Downarrow$

w_{-1} = Phytoplankton	word-within-k = ocean
w_{+1} = life	word-within-k = reflects
w_{-2}, w_{-1} = (Phytoplankton,microscopic)	word-within-k = bloom
w_{-1}, w_{+1} = (microscopic,life)	word-within-k = color
w_{+1}, w_{+2} = (life,that)	...

There are often many features which indicate the sense of the word

Another Useful Property: “One Sense per Discourse”

- Yarowsky observes that if the same word appears more than once in a document, then it is very likely to have the same sense every time

Step 1 of the Method: Collecting Seed Examples

- Goal: start with a small subset of the training data being labeled

- Various methods for achieving this:
  - Label a number of training examples by hand
  - Pick a single feature for each class by hand
    e.g., `word-within-k=bird` and
    `word-within-k=machinery` for *crane*
  - Look through frequently occurring features, and label a few of them
  - Using words in dictionary definitions
    e.g., Pick words in the two definitions for “plant”

    A vegetable organism, or part of one, ready for planting or lately planted.

    equipment, machinery, apparatus, for an industrial activity

An example: for the “plant” sense distinction, initial seeds are `word-within-k=life` and `word-within-k=manufacturing`

Partitions the unlabeled data into three sets:

- 82 examples labelled with “life” sense
- 106 examples labelled with “manufacturing” sense
- 7350 unlabeled examples

Training New Rules

1. From the seed data, learn a decision list of all rules with weight above some threshold (e.g., all rules with weight > 0.97)

2. Using the new rules, relabel the data
   (usually we will now end up with more data being labeled)

3. Induce a new set of rules with weight above the threshold from the labeled data

4. If some examples are still not labeled, return to step 2

Experiments

- Yarowsky describes several experiments:
  - A baseline score for just picking the most frequent sense for each word
  - Score for a fully supervised method
  - Partially supervised method with "two words" as a seed
  - Partially supervised method with dictionary defn. as a seed
  - Partially supervised method with hand-chosen rules as a seed
  - Dictionary defn. method combined with one-sense-per-discourse constraint

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
			%		Seed Training Options			(7) + OSPD		
Word	Senses	Samp. Size	Major Sense	Supvsd Algrtm	Two Words	Dict. Defn.	Top Colls.	End only	Each Iter.	Schütze Algrthm
plant	living/factory	7538	53.1	97.7	97.1	97.3	97.6	98.3	98.6	92
space	volume/outer	5745	50.7	93.9	89.1	92.3	93.5	93.3	93.6	90
tank	vehicle/container	11420	58.2	97.1	94.2	94.6	95.8	96.1	96.5	95
motion	legal/physical	11968	57.5	98.0	93.5	97.4	97.4	97.8	97.9	92
bass	fish/music	1859	56.1	97.8	96.6	97.2	97.7	98.5	98.8	–
palm	tree/hand	1572	74.9	96.5	93.9	94.7	95.8	95.5	95.9	–
poach	steal/boil	585	84.6	97.1	96.6	97.2	97.7	98.4	98.5	–
axes	grid/tools	1344	71.8	95.5	94.0	94.3	94.7	96.8	97.0	–
duty	tax/obligation	1280	50.0	93.7	90.4	92.1	93.2	93.9	94.1	–
drug	medicine/narcotic	1380	50.0	93.0	90.4	91.4	92.6	93.3	93.9	–
sake	benefit/drink	407	82.8	96.3	59.6	95.8	96.1	96.1	97.5	–
crane	bird/machine	2145	78.0	96.6	92.3	93.6	94.2	95.4	95.5	–
AVG		3936	63.9	96.1	90.6	94.8	95.5	96.1	96.5	92.2

4 after the algorithm has converged, or in Step 3c after each iteration.

At the end of Step 4, this property is used for error correction. When a polysemous word such as *plant* occurs multiple times in a discourse, tokens that were tagged by the algorithm with low confidence using local collocation information may be overridden by the dominant tag for the discourse.

however, as such isolated tokens tend to strongly favor a particular sense (the less "bursty" one). We have yet to use this additional information.

8 Evaluation

The words used in this evaluation were randomly selected from those previously studied in the litera-

Some Comments

- Very impressive results using relatively little supervision
- How well would this perform on words with "weaker" sense distinctions? (e.g., *interest*)
- Can we give formal guarantees for when this method will/won't work?
  (how to give a formal characterization of redundancy, and show that this implies guarantees concerning the utility of unlabeled data?)
- There are several "tweakable" parameters of the method
  (e.g., the weight threshold used to filter the rules)
- Another issue: the method as described may not ever label all examples

Overview

- A supervised method for word-sense disambiguation: decision lists
- A semi-supervised method for word-sense disambiguation
- A semi-supervised method for named-entity classification

Supervised Learning

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
- Task is to learn a function $F : \mathcal{X} \rightarrow \mathcal{Y}$

Statistical Assumptions

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
- Task is to learn a function $F : \mathcal{X} \rightarrow \mathcal{Y}$
- Typical assumption is that there is some distribution $P(x, y)$ from which examples are drawn
- Aim is to find a function F with a low value for

$$Er(F) = \sum_{x,y} P(x, y)[[F(x) \neq y]]$$

i.e., minimize probability of error on new examples

Partially Supervised Learning

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$ (n is typically small)
- We have **unlabeled** examples (x_i) for $i = (n + 1) \ldots (n + m)$
- Task is to learn a function $F : \mathcal{X} \rightarrow \mathcal{Y}$
- New questions:
  - Under what assumptions is unlabeled data “useful”?
  - Can we fi nd NLP problems where these assumptions hold?
  - Which algorithms are suggested by the theory?

Named Entity Classification

- Classify entities as organizations, people or locations

Steptoe & Johnson	= Organization
Mrs. Frank	= Person
Honduras	= Location

- Need to learn (weighted) rules such as

contains(Mrs.)	$\Rightarrow$	Person
full-string=Honduras	$\Rightarrow$	Location
context=company	$\Rightarrow$	Organization

An Approach Using Minimal Supervision

- Assume a small set of "seed" rules

contains(Incorporated)	$\Rightarrow$	Organization
full-string=Microsoft	$\Rightarrow$	Organization
full-string=I.B.M.	$\Rightarrow$	Organization
contains(Mr.)	$\Rightarrow$	Person
full-string=New_York	$\Rightarrow$	Location
full-string=California	$\Rightarrow$	Location
full-string=U.S.	$\Rightarrow$	Location

- Assume a large amount of unlabeled data

  .., says Mr. Cooper, a vice president of ...

- Methods gain leverage from redundancy:

  Either Spelling or Context alone is often sufficient to determine an entity's type

Cotraining (Blum and Mitchell, 1998)

- We have domains $\mathcal{X}$, $\mathcal{Y}$
- We have **labeled** examples (x_i, y_i) for $i = 1 \ldots n$
- We have **unlabeled** examples (x_i) for $i = (n+1) \ldots (n+m)$
- We assume each example x_i splits into two views, x_{1i} and x_{2i}
- e.g., if x_i is a feature vector in $\mathbb{R}^{2d}$, then x_{1i} and x_{2i} are representations in $\mathbb{R}^d$.

The Data

- Approx 90,000 spelling/context pairs collected
- Two types of contexts identified by a parser

1. Appositives

   .., says Mr. Cooper, a vice president of ...

2. Prepositional Phrases

   Robert Haft , president of the Dart Group Corporation ...

Features: Two Views of Each Example

.., says Mr. Cooper, a vice president of ...

$\Downarrow$

Spelling Features	Contextual Features
Full-String = Mr. Cooper	appositive = president
Contains(Mr.)	
Contains(Cooper)	

Two Assumptions Behind Cotraining

Assumption 1: Either view is sufficient for learning

There are functions F_1 and F_2 such that

$$F(x) = F_1(x_1) = F_2(x_2) = y$$

for all (x, y) pairs

Examples of Problems with Two Natural Views

- Named entity classification (spelling vs. context)
- Web page classification [Blum and Mitchell, 1998]
  One view = words on the page, other view is pages linking to a page
- Word sense disambiguation: a random split of the text?

A Key Property: Redundancy

The ocean reflects the color of the sky, but even on cloudless days the color of the ocean is not a consistent blue. Phytoplankton, microscopic plant life that floats freely in the lighted surface waters, may alter the color of the water. When a great number of organisms are concentrated in an area, the plankton changes the color of the ocean surface. This is called a 'bloom.'

$\Downarrow$

w_{-1} = Phytoplankton	word-within-k = ocean
w_{+1} = life	word-within-k = reflects
w_{-2}, w_{-1} = (Phytoplankton,microscopic)	word-within-k = bloom
w_{-1}, w_{+1} = (microscopic,life)	word-within-k = color
w_{+1}, w_{+2} = (life,that)	...

There are often many features which indicate the sense of the word

Two Assumptions Behind Cotraining

Assumption 2:
Some notion of independence between the two views

e.g., The **Conditional-independence-given-label** assumption:
If $P(x_1, x_2, y)$ is the distribution over examples, then

$$P(x_1, x_2, y) = P_0(y)P_1(x_1 \mid y)P_2(x_2 \mid y)$$

for some distributions P_0, P_1 and P_2

Why are these Assumptions Useful?

- Two examples/scenarios:
  - Rote learning, and a graph interpretation
  - Constraints on hypothesis spaces

Rote Learning, and a Graph Interpretation

- In a rote learner, functions F_1 and F_2 are look-up tables

Spelling	Category
Robert-Jordan	PERSON
Washington	LOCATION
Washington	LOCATION
Jamie-Gorelick	PERSON
Jerry-Jasinowski	PERSON
Pacifi Corp	COMPANY
...	...

Context	Category
partner	PERSON
partner-at	COMPANY
law-in	LOCATION
fi rm-in	LOCATION
partner	PERSON
partner-of	COMPANY
...	...

- Note: this can be a very inefficient learning method
  (no chance to learn generalizations such as "any name containing *Mr.* is a person")

Rote Learning, and a Graph Interpretation

- Each node in the graph is a spelling or context
  A node for *Robert Jordan*, *Washington*, *law-in*, *partner* etc.

- Each (x_{1i}, x_{2i}) pair is an edge in the graph
  e.g., (Robert Jordan, partner)

- An edge between two nodes mean they have **the same label**
  (relies on assumption 1: each view is sufficient for classification)

- As quantity of unlabeled data increases, graph becomes more connected
  (relies on assumption 2: some independence between the two views)

Constraints on Hypothesis Spaces

- $n + m$ training examples $x_i = (x_{1i}, x_{2i})$
- First n examples have labels y_i
- Learn functions F_1 and F_2 such that

$$F_1(x_{1i}) = F_2(x_{2i}) = y_i \qquad i = 1 \ldots n$$

$$F_1(x_{1i}) = F_2(x_{2i}) \qquad i = n+1 \ldots n+m$$

- The second set of constraints is new, and may significantly restrict the set of possible functions F_1 and F_2. This may significantly reduce the number of labeled examples, n, that are required for accurate learning.

A Linear Model

- How to build a classifier from spelling features alone?
  A linear model:
  - $\mathbf{GEN}(x_1)$ is possible labels $\{person, location, organization\}$
  - $\mathbf{f}(x_1, y)$ is a set of features on spelling/label pairs, e.g.,

$$f_{100}(x_1, y) = \begin{cases} 1 & \text{if } x_1 \text{ contains } \textit{Mr.}\text{, and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{101}(x_1, y) = \begin{cases} 1 & \text{if } x_1 \text{ is } \textit{IBM}\text{, and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

  - $\mathbf{w}$ is parameter vector, as usual choose

$$F_1(x_1, \mathbf{w}) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{f}(x_1, y) \cdot \mathbf{w}$$

  - $\Rightarrow$ each parameter in $\mathbf{w}$ gives a weight for a feature/label pair.
    e.g., $\mathbf{w}_{100} = 2.5$, $\mathbf{w}_{101} = -1.3$

A Boosting Approach to Supervised Learning

- Greedily minimize

$$L(\mathbf{w}) = \sum_i \sum_{y \neq y_i} e^{-\mathbf{m}(y_i, y, \mathbf{w})}$$

where

$$\mathbf{m}(y_i, y, \mathbf{w}) = \mathbf{f}(x_i, y_i) \cdot \mathbf{w} - \mathbf{f}(x_i, y) \cdot \mathbf{w}$$

- $L(\mathbf{w})$ is an upper bound on the number of ranking errors,

$$L(\mathbf{w}) \geq \sum_i \sum_{y \neq y_i} [[\mathbf{m}(y_i, y, \mathbf{w}) \leq 0]]$$

(Note: we define $[[\pi]]$ to be 1 if the statement π is true, 0 otherwise)

An Extension to the Cotraining Scenario

- Now build **two** linear models in parallel
  - $\mathbf{GEN}(x_1) = \mathbf{GEN}(x_2)$ is set of possible labels $\{person, location, organization\}$
  - $\mathbf{f}^1(x_1, y)$ is a set of features on spelling/label pairs
  - $\mathbf{f}^2(x_2, y)$ is a set of features on context/label pairs, e.g.,

$$f^2{}_{100}(x_2, y) = \begin{cases} 1 & \text{if } x_2 \text{ is } \textit{president} \text{ and } y = \textit{person} \\ 0 & \text{otherwise} \end{cases}$$

  - $\mathbf{w}^1$ and $\mathbf{w}^2$ are the two parameter vectors

$$F_1(x_1, \mathbf{w}^1) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{f}^1(x_1, y) \cdot \mathbf{w}^1$$

$$F_2(x_2, \mathbf{w}^2) = \arg \max_{y \in \mathbf{GEN}(x_2)} \mathbf{f}^2(x_2, y) \cdot \mathbf{w}^2$$

An Extension to the Cotraining Scenario

- $n + m$ training examples $x_i = (x_{1i}, x_{2i})$
- First n examples have labels y_i
- Linear models defi ne F_1 and F_2 as

$$F_1(x_1, \mathbf{w}^1) = \arg \max_{y \in \mathbf{GEN}(x_1)} \mathbf{f}^1(x_1, y) \cdot \mathbf{w}^1$$

$$F_2(x_2, \mathbf{w}^2) = \arg \max_{y \in \mathbf{GEN}(x_2)} \mathbf{f}^2(x_2, y) \cdot \mathbf{w}^2$$

- Three types of errors:

$$E_1 = \sum_{i=1}^{n} [[F_1(x_{1i}, \mathbf{w}^1) \neq y_i]]$$

$$E_2 = \sum_{i=1}^{n} [[F_2(x_{2i}, \mathbf{w}^2) \neq y_i]]$$

$$E_3 = \sum_{i=n+1}^{m+1} [[F_1(x_{1i}, \mathbf{w}^1) \neq F_2(x_{2i}, \mathbf{w}^2)]]$$

Objective Functions for Cotraining

- Define "pseudo labels"

$$z_{1i}(\mathbf{w}^1) = F_1(x_{1i}, \mathbf{w}^1) \qquad i = (n+1)\ldots(n+m)$$
$$z_{2i}(\mathbf{w}^2) = F_2(x_{2i}, \mathbf{w}^2) \qquad i = (n+1)\ldots(n+m)$$

e.g., z_{1i} is output of first classifier on the i'th example

$$\begin{aligned} L(\mathbf{w}^1, \mathbf{w}^2) \;=\; & \sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{f}^1(x_{1i},y)\cdot\mathbf{w}^1 - \mathbf{f}^1(x_{1i},y_i)\cdot\mathbf{w}^1} \\ & + \sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{f}^2(x_{2i},y)\cdot\mathbf{w}^2 - \mathbf{f}^2(x_{2i},y_i)\cdot\mathbf{w}^2} \\ & + \sum_{i=n+1}^{n+m} \sum_{y \neq z_{2i}} e^{\mathbf{f}^1(x_{1i},y)\cdot\mathbf{w}^1 - \mathbf{f}^1(x_{1i},z_{2i})\cdot\mathbf{w}^1} \\ & + \sum_{i=n+1}^{n+m} \sum_{y \neq z_{1i}} e^{\mathbf{f}^2(x_{2i},y)\cdot\mathbf{w}^2 - \mathbf{f}^2(x_{2i},z_{2i})\cdot\mathbf{w}^2} \end{aligned}$$

More Intuition

- Need to minimize $L(\mathbf{w}^1, \mathbf{w}^2)$, do this by greedily minimizing w.r.t. first $\mathbf{w}^1$, then $\mathbf{w}^2$
- Algorithm boils down to:

1. Start with labeled data alone
2. Induce a contextual feature for each class (person/location/organization) from the current set of labelled data
3. Label unlabeled examples using contextual rules
4. Induce a spelling feature for each class (person/location/organization) from the current set of labelled data
5. Label unlabeled examples using spelling rules
6. Return to step 2

Optimization Method

1. Set pseudo labels z_{2i}
2. Update $\mathbf{w}^1$ to minimize

$$\sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{f}^1(x_{1i},y)\cdot\mathbf{w}^1 - \mathbf{f}^1(x_{1i},y_i)\cdot\mathbf{w}^1}$$
$$+ \sum_{i=n+1}^{n+m} \sum_{y \neq z_{2i}} e^{\mathbf{f}^1(x_{1i},y)\cdot\mathbf{w}^1 - \mathbf{f}^1(x_{1i},z_{2i})\cdot\mathbf{w}^1}$$

(for each class choose a spelling feature, weight)

3. Set pseudo labels z_{1i}
4. Update $\mathbf{w}^2$ to minimize

$$\sum_{i=1}^{n} \sum_{y \neq y_i} e^{\mathbf{f}^2(x_{2i},y)\cdot\mathbf{w}^2 - \mathbf{f}^2(x_{2i},y_i)\cdot\mathbf{w}^2}$$
$$+ \sum_{i=n+1}^{n+m} \sum_{y \neq z_{1i}} e^{\mathbf{f}^2(x_{2i},y)\cdot\mathbf{w}^2 - \mathbf{f}^2(x_{2i},z_{2i})\cdot\mathbf{w}^2}$$

(for each class choose a contextual feature, weight)

5. Return to step 1

An Example Trace

1. Use seeds to label 8593 examples
   (4160 companies, 2788 people, 1645 locations)
2. Pick a contextual feature for each class:

COMPANY:	preposition=unit of	2.386	274/2
PERSON:	appositive=president	1.593	120/6
LOCATION:	preposition=Company of	1.673	46/1

3. Set pseudo labels using seeds + contextual features
   (5319 companies, 6811 people, 1961 locations)
4. Pick a spelling feature for each class

COMPANY:	Contains(Corporation)	2.475	495/10
PERSON:	Contains(.)	2.482	4229/106
LOCATION:	fullstring=America	2.311	91/0

5. Set pseudo labels using seeds + spelling features
   (7180 companies, 8161 people, 1911 locations)
6. Continue ...

Evaluation

- 88,962 $(spelling, context)$ pairs extracted as training data
- 7 seed rules used

contains(Incorporated)	$\Rightarrow$	Organization
full-string=Microsoft	$\Rightarrow$	Organization
full-string=I.B.M.	$\Rightarrow$	Organization
contains(Mr.)	$\Rightarrow$	Person
full-string=New_York	$\Rightarrow$	Location
full-string=California	$\Rightarrow$	Location
full-string=U.S.	$\Rightarrow$	Location

- 1,000 examples picked at random, and labelled by hand to give a test set.

- Around 9% of examples were "noise", not falling into any of the three categories
- Two measures given: one excluding all noise items, the other counting noise items as errors

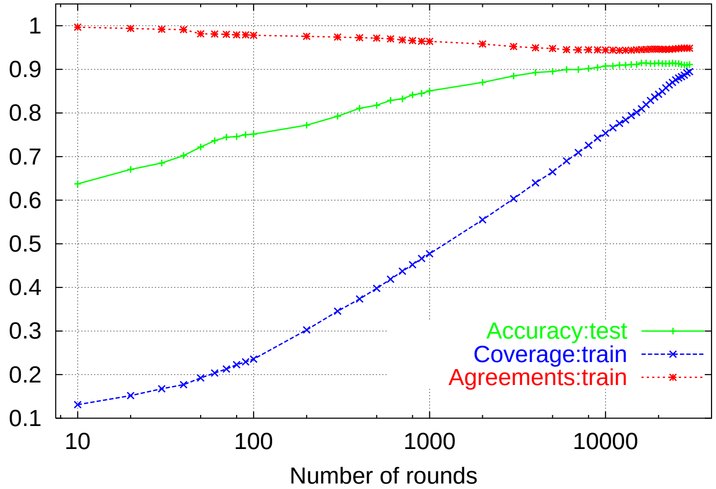
Other Methods

- EM approach
- Decision list (Yarowsky 95)
- Decision list 2 (modification of Yarowsky 95)
- DL-Cotrain:
  decision list alternating between two feature types

Results

Learning Algorithm	Accuracy (Clean)	Accuracy (Noise)
Baseline	45.8%	41.8%
EM	83.1%	75.8%
Decision List	81.3%	74.1%
Decision List 2	91.2%	83.2%
DL-CoTrain	91.3%	83.3%
CoBoost	91.1%	83.1%

Learning Curves for Coboosting



Summary

- Appears to be a complex task: many features/rules required
- With unlabeled data, supervision is reduced to 7 "seed" rules
- Key is **redundancy** in the data
- Cotraining suggests training two classifiers that "**agree**" as much as possible on unlabeled examples
- **CoBoost** algorithm builds two additive models in parallel, with an objective function that bounds the rate of agreement