Weakly-Supervised Learning with Cost-Augmented Contrastive Estimation

Kevin Gimpel Mohit Bansal

TOYOTA TECHNOLOGICAL INSTITUTE CHICAGO

- New objective for weakly-supervised NLP, generalizes contrastive estimation (Smith & Eisner, 2005)
- Adds two cost functions: inputs and outputs
- Improved system combination for POS tagging

	many-to-1 accuracy	1-to-1 accuracy
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation	**64.3**	**51.7**

avg. across 5 languages,
PASCAL 2012 POS shared task

- New objective for weakly-supervised NLP, generalizes contrastive estimation (Smith & Eisner, 2005)
- Adds two cost functions: inputs and outputs
- Improved system combination for POS tagging

	many-to-1 accuracy	1-to-1 accuracy
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation	**64.3**	**51.7**
Posterior Regularization (Graça et al., 2011)	60.9	50.1

avg. across 5 languages,
PASCAL 2012 POS shared task

EM and Contrastive Estimation

Modification 1: Input Cost

Modification 2: Output Cost

Generative Log-Linear Models

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\right\}}{\sum_{\boldsymbol{x}'} \sum_{\boldsymbol{y}'} \exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{y}')\right\}}$$

Generative Log-Linear Models

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\right\}}{\sum_{\boldsymbol{x}'} \sum_{\boldsymbol{y}'} \exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{y}')\right\}}$$

word sequence

part-of-speech tag sequence

Generative Log-Linear Models

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\right\}}{\sum_{\boldsymbol{x}'} \sum_{\boldsymbol{y}'} \exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{y}')\right\}}$$

word sequence

part-of-speech tag sequence

parameters

feature vector

Generative Log-Linear Models

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\exp\left\{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\right\}}{\sum_{\boldsymbol{x}'} \sum_{\boldsymbol{y}'} \exp\left\{\underbrace{\boldsymbol{\theta}^\top \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{y}')}_{\text{score}(\boldsymbol{x}', \boldsymbol{y}')}\right\}}$$

Unsupervised Learning for Log-Linear Models

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_i \operatorname{gain}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta})$$

EM

$$\mathrm{gain}_{\mathrm{EM}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y})$$

EM

$$\mathrm{gain}_{\mathrm{EM}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) =$$

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \right\} \; - \; \log \sum_{\boldsymbol{x}' \in \mathcal{X}} \sum_{\boldsymbol{y}'} \exp \{ \mathrm{score}(\boldsymbol{x}', \boldsymbol{y}') \}$$

EM

$$\text{gain}_{\text{EM}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) =$$

$$\underbrace{\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\}}_{\text{reward all y's for observed x}} - \underbrace{\log \sum_{\boldsymbol{x}' \in \mathcal{X}} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\right\}}_{\text{penalize all y's for ALL x's}}$$

Contrastive Estimation (CE)

(Smith & Eisner, 2005)

$$\mathrm{gain}_{\mathrm{CE}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y} \mid \mathcal{N}(\boldsymbol{x}^{(i)}))$$

"corruption neighborhood"

Contrastive Estimation (CE)

(Smith & Eisner, 2005)

$$\text{gain}_{\text{CE}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y} \mid \mathcal{N}(\boldsymbol{x}^{(i)})) =$$

$$\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\} \; - \; \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\right\}$$

Contrastive Estimation (CE)

(Smith & Eisner, 2005)

$$\text{gain}_{\text{CE}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y} \mid \mathcal{N}(\boldsymbol{x}^{(i)})) =$$

$$\underbrace{\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\}}_{\substack{\text{reward all y's for observed x} \\ \text{(same as EM)}}} \; - \; \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\}$$

Contrastive Estimation (CE)

(Smith & Eisner, 2005)

$$\text{gain}_{\text{CE}}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}, \boldsymbol{y} \mid \mathcal{N}(\boldsymbol{x}^{(i)})) =$$

$$\underbrace{\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\}}_{\text{reward all y's for observed x (same as EM)}} \; - \; \underbrace{\log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\right\}}_{\text{penalize all y's for x's in corruption neighborhood}}$$

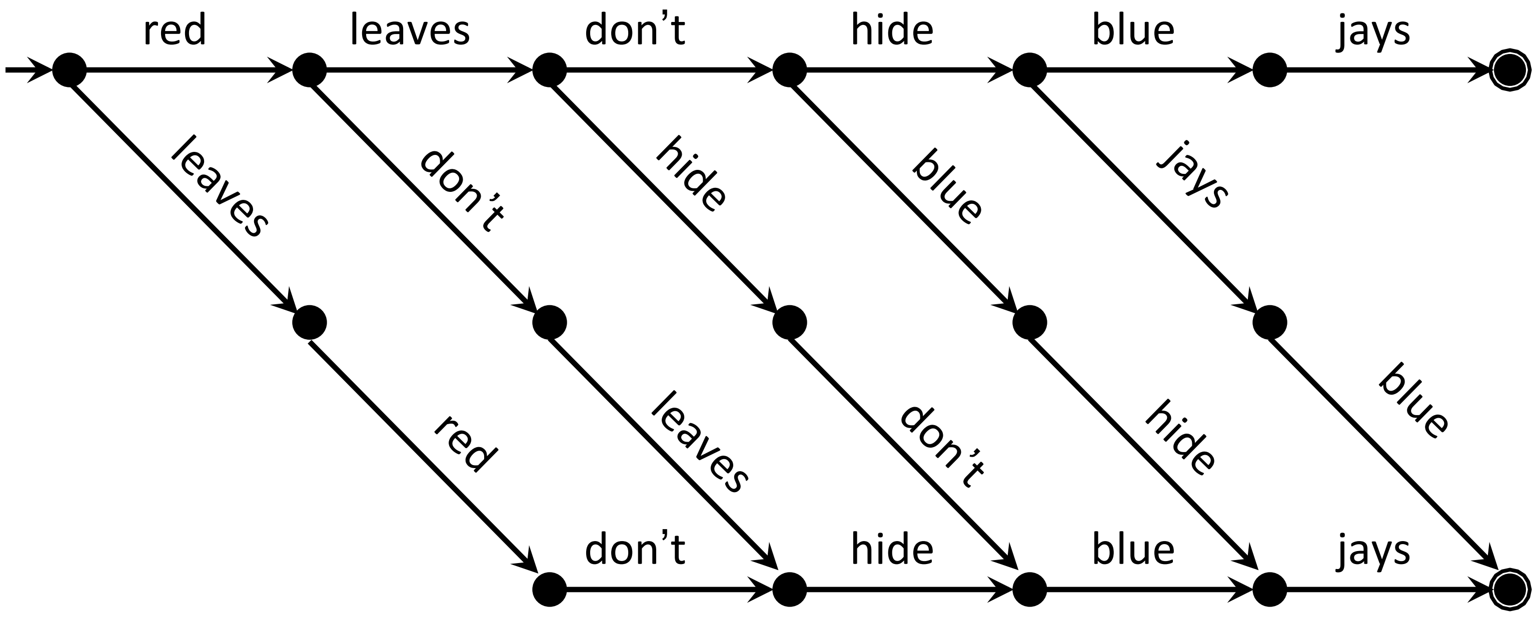
With well-designed neighborhood, CE shown effective for:

- part-of-speech tagging (Smith & Eisner, 2005a)
- dependency parsing (Smith & Eisner, 2005b)
- morphological segmentation (Poon et al., 2009)
- bilingual part-of-speech induction (Chen et al., 2011)
- machine translation (Xiao et al., 2011)

“Transpose1” Neighborhood

Sentence: red leaves don’t hide blue jays

Neighborhood:



Smith & Eisner (2005)

EM and Contrastive Estimation

Modification 1: Input Cost

Modification 2: Output Cost

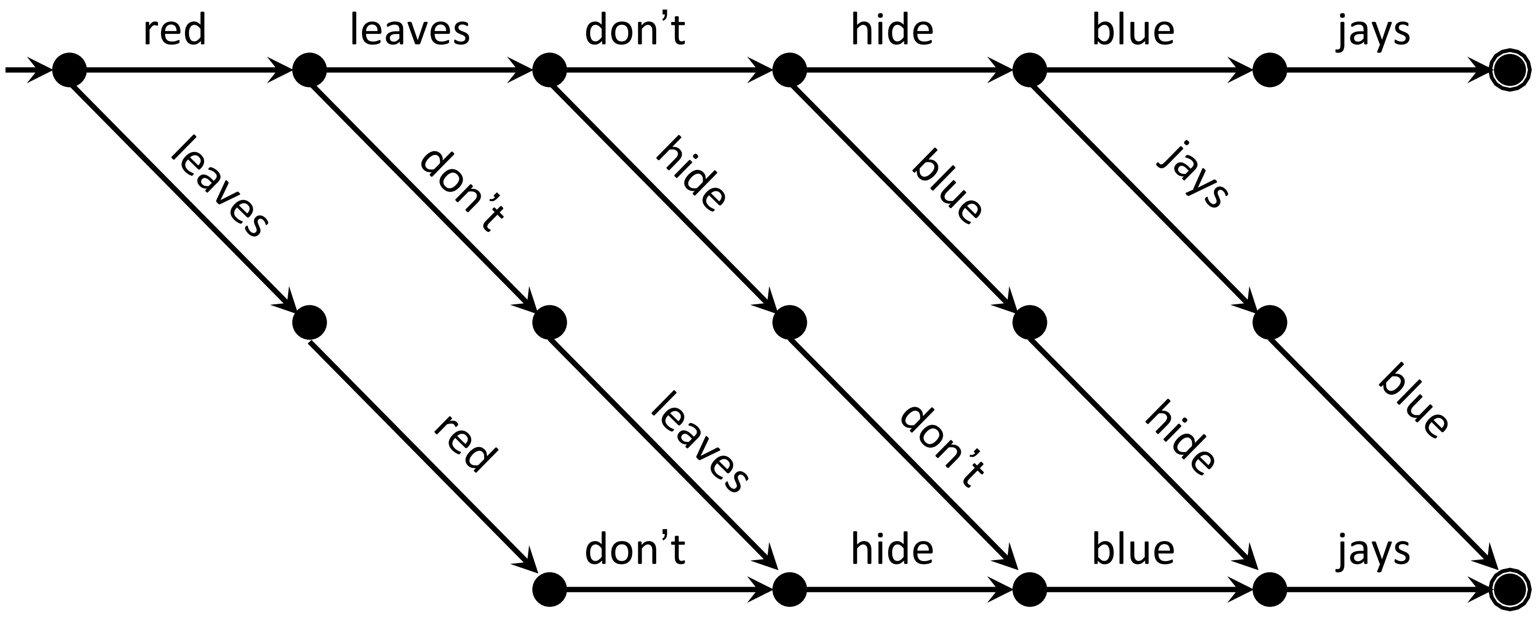
Contrastive Estimation:

$$\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\} \; - \; \underbrace{\log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\right\}}_{\text{all x's in corruption neighborhood treated equally!}}$$

Transpose1 Neighborhood

Sentence: red leaves don’t hide blue jays

Neighborhood:



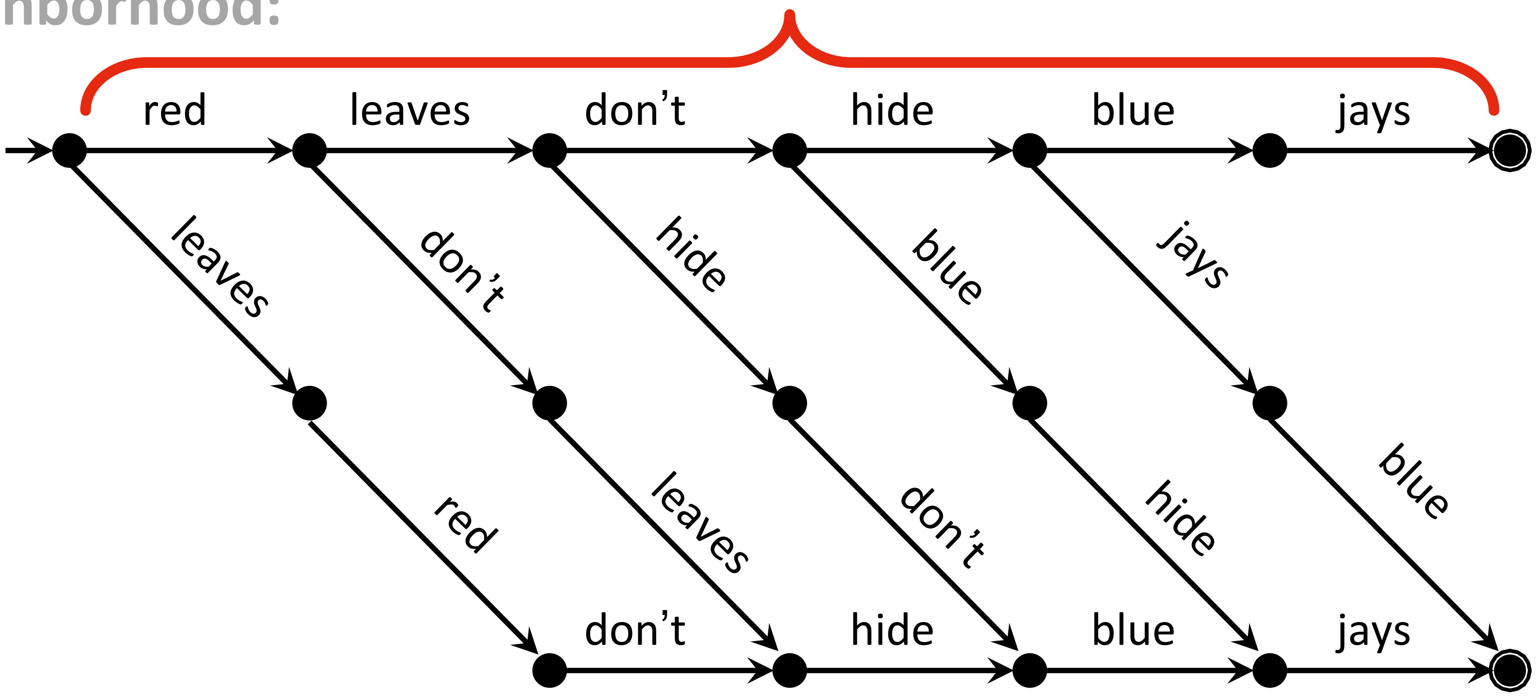
Smith & Eisner (2005)

Transpose1 Neighborhood

Sentence: red leaves don't hide blue jays

neighborhood always contains original sentence

Neighborhood:

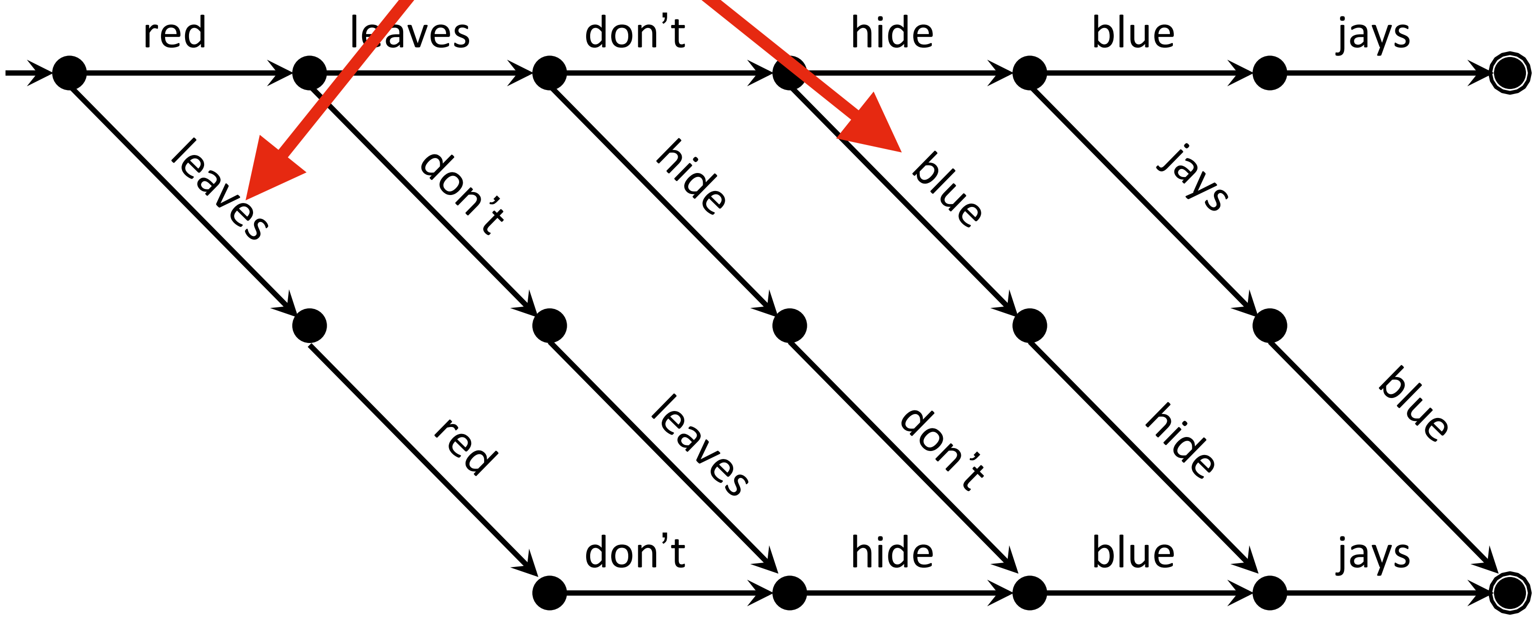


Smith & Eisner (2005)

Transpose1 Neighborhood

some corruptions not as bad as others

Neighborhood:



Smith & Eisner (2005)

First modification:

add **input cost function** $\Delta(x, x')$

First modification: add **input cost function** $\Delta(\boldsymbol{x}, \boldsymbol{x}')$

$$\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\} - \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}') + \underbrace{\alpha \Delta(\boldsymbol{x}^{(i)}, \boldsymbol{x}')}\right\}$$

measures difference between observed and corrupted sentences, α is weight

Inspiration: Structured Large-Margin Learning

margin-rescaled structured hinge (Taskar et al., 2003):

$$\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \max_{\boldsymbol{y}} \left(\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})\right)$$

softmax-margin (Povey et al., 2008; Gimpel & Smith, 2010) :

$$\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \log \sum_{\boldsymbol{y}} \exp \left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})\right\}$$

Inspiration: Structured Large-Margin Learning

margin-rescaled structured hinge (Taskar et al., 2003):

$$\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \max_{\boldsymbol{y}} \left(\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) + \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) \right)$$

softmax-margin (Povey et al., 2008; Gimpel & Smith, 2010) :

$$\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) - \log \sum_{\boldsymbol{y}} \exp \left\{ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) + \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) \right\}$$

(soft)max-margin: cost compares two outputs

this talk: cost compares two **inputs**

Input Cost Functions $\Delta(\boldsymbol{x}^{(i)}, \boldsymbol{x})$

Match:

count unmatched bigrams in corrupted sentence

$$\sum_{j=1}^{|\boldsymbol{x}|+1} \mathbb{I}\left[x_{j-1}x_j \notin \mathrm{bigrams}(\boldsymbol{x}^{(i)})\right]$$

Match LM:

weight by language model (negative) log-probability

$$\sum_{j=1}^{|\boldsymbol{x}|+1} -\log \Pr(x_j | x_{j-1})\ \mathbb{I}\left[x_{j-1}x_j \notin \mathrm{bigrams}(\boldsymbol{x}^{(i)})\right]$$

Experiments

Unsupervised part-of-speech tagging, 12 tags, no tag dictionaries

Evaluation: many-to-1 & 1-to-1 accuracy

5 languages from PASCAL 2012 shared task (Gelling et al., 2012):
Danish, Dutch, Portuguese, Slovene, Swedish

Neighborhoods

Transpose1 (Smith & Eisner, 2005)

Shuffle10:

original sentence + 10 random permutations

Setup

Features:

- tag-tag transitions
- tag-word emissions
- spelling features (Smith & Eisner, 2005)
- tag-cluster emissions (from Brown clustering with {12,40} clusters)

LBFGS for 100 iterations, random initialization

L2 regularization with (untuned) coefficient 0.0001

	input cost	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	None (CE baseline)	51.3	39.7
Transpose1	None (CE baseline)	61.8	47.2

avg. across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	input cost	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	None (CE baseline)	51.3	39.7
	Match	53.3 (+2.0)	40.5 (+0.8)
Transpose1	None (CE baseline)	61.8	47.2
	Match	63.1 (+1.3)	47.6 (+0.4)

avg. across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	input cost	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	None (CE baseline)	51.3	39.7
	Match	53.3 (+2.0)	40.5 (+0.8)
	Match LM	53.9 (+2.6)	41.6 (+1.9)
Transpose1	None (CE baseline)	61.8	47.2
	Match	63.1 (+1.3)	47.6 (+0.4)
	Match LM	62.8 (+1.0)	49.9 (+2.7)

avg. across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	input cost	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	None (CE baseline)	51.3	39.7
	Match	53.3 (+2.0)	40.5 (+0.8)
	Match LM	53.9 (+2.6)	41.6 (+1.9)
Transpose1	None (CE baseline)	61.8	47.2
	Match	63.1 (+1.3)	47.6 (+0.4)
	Match LM	62.8 (+1.0)	49.9 (+2.7)

Using language model probabilities helps

EM and Contrastive Estimation

Modification 1: Input Cost

Modification 2: Output Cost

Contrastive Estimation:

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \right\} \; - \; \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp \left\{ \text{score}(\boldsymbol{x}', \boldsymbol{y}') \right\}$$

Contrastive Estimation:

$$\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y})\right\} \;-\; \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\text{score}(\boldsymbol{x}', \boldsymbol{y}')\right\}$$

we sum over all y's for each x (observed or corrupted)

how can we encode intuitions about y?

Second modification: adding an output cost function $\pi(\boldsymbol{y})$

$$\log \sum_{\boldsymbol{y}} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \beta\pi(\boldsymbol{y})\right\} - \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\{\text{score}(\boldsymbol{x}', \boldsymbol{y}') + \beta\pi(\boldsymbol{y}')\}$$

expresses preferences on outputs, regardless of input

Second modification: adding an **output cost function** $\pi(\boldsymbol{y})$

$$\log \sum_{\boldsymbol{y}} \exp\left\{\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \beta\pi(\boldsymbol{y})\right\} - \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp\left\{\mathrm{score}(\boldsymbol{x}', \boldsymbol{y}') + \beta\pi(\boldsymbol{y}')\right\}$$

expresses preferences on outputs, regardless of input

similar to ``structural bias'' (Smith & Eisner, 2006), posterior regularization (Graça et al., 2010), and universal dependency rules (Naseem et al., 2010)

Inspiration

Some objectives for *supervised* learning never need to score the true output:

ramp (Do et al., 2008):

$$\max_{\boldsymbol{y}} \ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) \ - \ \max_{\boldsymbol{y}'} \left(\mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}') + \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}')\right)$$

supervision
used only in
cost function

“Soft” ramp gain (Gimpel, 2012):

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) \right\} - \log \sum_{\boldsymbol{y}'} \exp \left\{ \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}') + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}') \right\}$$

CE with output cost function (this talk):

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \beta\pi(\boldsymbol{y}) \right\} - \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp \left\{ \text{score}(\boldsymbol{x}', \boldsymbol{y}') + \beta\pi(\boldsymbol{y}') \right\}$$

"Soft" ramp gain (Gimpel, 2012):

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) \right\} - \log \sum_{\boldsymbol{y}'} \exp \left\{ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}') + \mathrm{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}') \right\}$$

CE with output cost function (this talk):

$$\log \sum_{\boldsymbol{y}} \exp \left\{ \mathrm{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}) - \beta\pi(\boldsymbol{y}) \right\} - \log \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x}^{(i)})} \sum_{\boldsymbol{y}'} \exp \{ \mathrm{score}(\boldsymbol{x}', \boldsymbol{y}') + \beta\pi(\boldsymbol{y}') \}$$

true y dropped from cost function

contrastive neighborhood used for denominator

Universal Tag Priors

We counted tags in 11 treebanks (for languages not used in our experiments)

$$\mathrm{cost}(y) = \log\left(\frac{\max_{y'} \mathrm{count}(y')}{\mathrm{count}(y)}\right)$$

Universal Tag Priors

We counted tags in 11 treebanks (for languages not used in our experiments)

$$\text{cost}(y) = \log\left(\frac{\max_{y'} \text{count}(y')}{\text{count}(y)}\right)$$

tag	**count**	**cost**
noun	2.3M	0
punctuation	1M	0.81
verb	1M	0.83
adposition	900K	0.95
adjective	700K	1.21
determiner	600K	1.33
pronoun	500K	1.62
conjunction	400K	1.68
adverb	300K	1.96
verb particle	179K	2.57
numeral	175K	2.59
X (“other”)	50K	3.83

tag bigram	count	cost
noun punctuation	500K	0
determiner noun	450K	1.04
noun noun	410K	2.09
…		
numeral adverb	1587	57.63
determiner conjunction	518	68.82
determiner particle	109	84.41

$$\mathrm{cost}(\langle y_1, y_2 \rangle) = 10 \times \log \left(\frac{\max_{\langle y'_1, y'_2 \rangle} \mathrm{count}(\langle y'_1, y'_2 \rangle)}{\mathrm{count}(\langle y_1, y_2 \rangle)} \right)$$

Results

	many-to-1 accuracy	**1-to-1 accuracy**
HMM, EM	50.9	34.2

accuracies averaged across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	many-to-1 accuracy	1-to-1 accuracy
HMM, EM	50.9	34.2
HMM, stepwise EM (Liang et al., 2009)	57.7	41.1

accuracies averaged across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	many-to-1 accuracy	1-to-1 accuracy
HMM, EM	50.9	34.2
HMM, stepwise EM (Liang et al., 2009)	57.7	41.1
Brown Clustering	57.6	45.5
`mkcls` (Och, 1995)	58.4	45.8

accuracies averaged across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	many-to-1 accuracy	1-to-1 accuracy
HMM, EM	50.9	34.2
HMM, stepwise EM (Liang et al., 2009)	57.7	41.1
Brown Clustering	57.6	45.5
`mkcls` (Och, 1995)	58.4	45.8
Posterior Regularization (Graça et al., 2011)	60.9	50.1

accuracies averaged across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	many-to-1 accuracy	1-to-1 accuracy
HMM, EM	50.9	34.2
HMM, stepwise EM (Liang et al., 2009)	57.7	41.1
Brown Clustering	57.6	45.5
`mkcls` (Och, 1995)	58.4	45.8
Posterior Regularization (Graça et al., 2011)	60.9	**50.1**
Contrastive Estimation	**61.8**	47.2

accuracies averaged across 5 languages:
Danish, Dutch, Portuguese, Slovene, Swedish

	many-to-1 accuracy	1-to-1 accuracy
Posterior Regularization	60.9	50.1
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation:		
Match LM	62.8	49.9

	many-to-1 accuracy	1-to-1 accuracy
Posterior Regularization	60.9	50.1
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation:		
Match LM	62.8	49.9
Universal	61.7	51.3

	many-to-1 accuracy	1-to-1 accuracy
Posterior Regularization	60.9	50.1
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation:		
Match LM	62.8	49.9
Universal	61.7	51.3
Match LM + Universal	**64.3**	**51.7**

Conclusions

- New learning criterion for weakly-supervised learning, generalizes contrastive estimation
- Cost functions allow modeler to direct learning in new ways
- Improves over strong POS tagging baselines

Thanks!

Unsupervised Model Selection

1. Maximize CE objective on held-out data
2. Maximize log-likelihood of held-out data
   - using efficient estimator of Bengio et al. (2013)
3. Voting:
   a. **naïve**: after making predictions with each model, return tags with most votes
   b. **align**: solve weighted bipartite matching problems to align tag identifiers across runs, then do voting

Comparing Model Selection Criteria

	cost	model selection	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	Match LM	contrastive estimation	53.2 (+1.9)	40.2 (+0.5)
		log-likelihood	53.9 (+2.6)	41.6 (+1.9)
Transpose1	Match LM	contrastive estimation	62.2 (+0.4)	47.5 (+0.3)
		log-likelihood	62.8 (+1.0)	49.9 (+2.7)

Comparing Model Selection Criteria

	cost	model selection	many-to-1 accuracy	1-to-1 accuracy
Shuffle10	Match LM	contrastive estimation	53.2 (+1.9)	40.2 (+0.5)
		log-likelihood	53.9 (+2.6)	41.6 (+1.9)
Transpose1	Match LM	contrastive estimation	62.2 (+0.4)	47.5 (+0.3)
		log-likelihood	62.8 (+1.0)	49.9 (+2.7)

Log-likelihood works better than CE

	many-to-1 accuracy	1-to-1 accuracy
Posterior Regularization	60.9	50.1
Contrastive Estimation	61.8	47.2
Cost-Augmented Contrastive Estimation:		
Match LM	62.8	49.9
Universal	61.7	51.3
Match LM + Universal (“naïve”)	60.6	51.4
Match LM + Universal (“align”)	**64.3**	**51.7**

Aligned voting works better than naïve voting