Discriminative Modeling Topics

April 2, 2013

Menu du Jour

- MaxEnt phrase reordering
  (Xiong et al., 2006)
- Discriminative lexicon models
  (Mauser et al., 2009)
- Translation as CRFs
  (Blunsom et al., 2008)

SCFGs: A problem

A → bruja | witch	0.2
A → verde | green	0.2
A → A A | 1 2	0.6
A → A A | 2 1	0.2

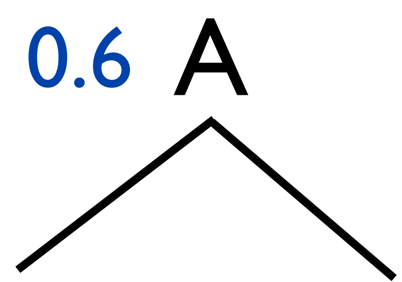
SCFGs: A problem

A → bruja | witch   0.2

A → verde | green   0.2

A → A A | 1 2   0.6

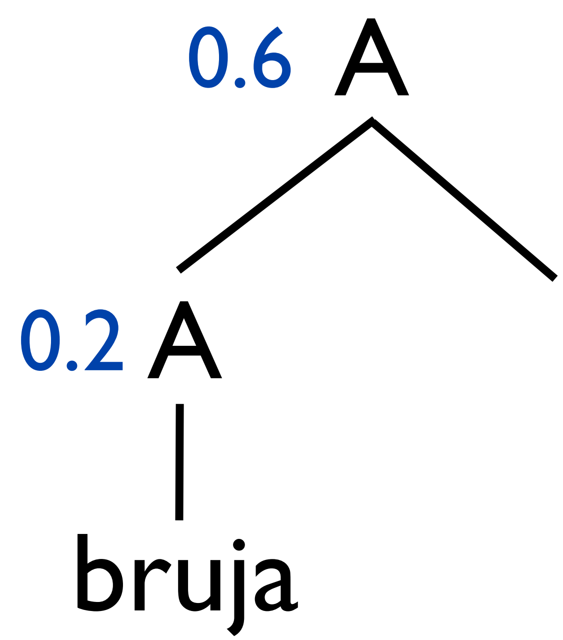
A → A A | 2 1   0.2



0.6

SCFGs: A problem

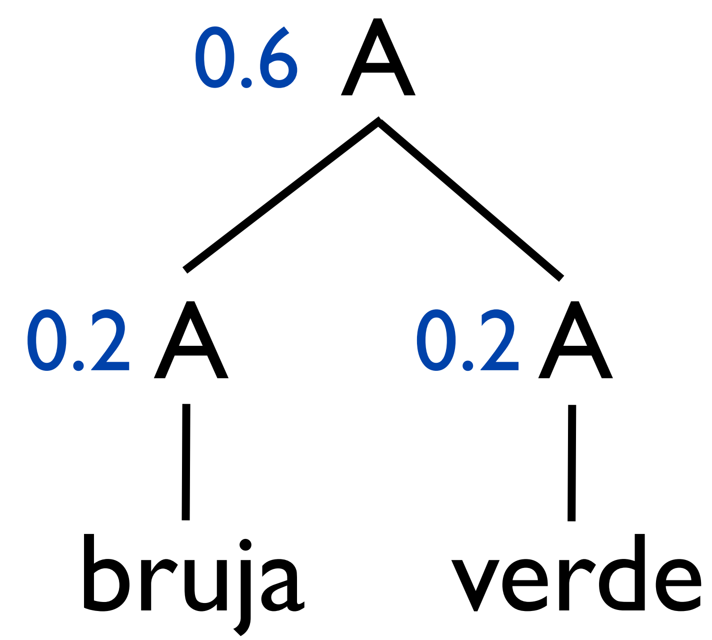
A → bruja | witch	0.2
A → verde | green	0.2
A → A A | 1 2	0.6
A → A A | 2 1	0.2



0.6x0.2

SCFGs: A problem

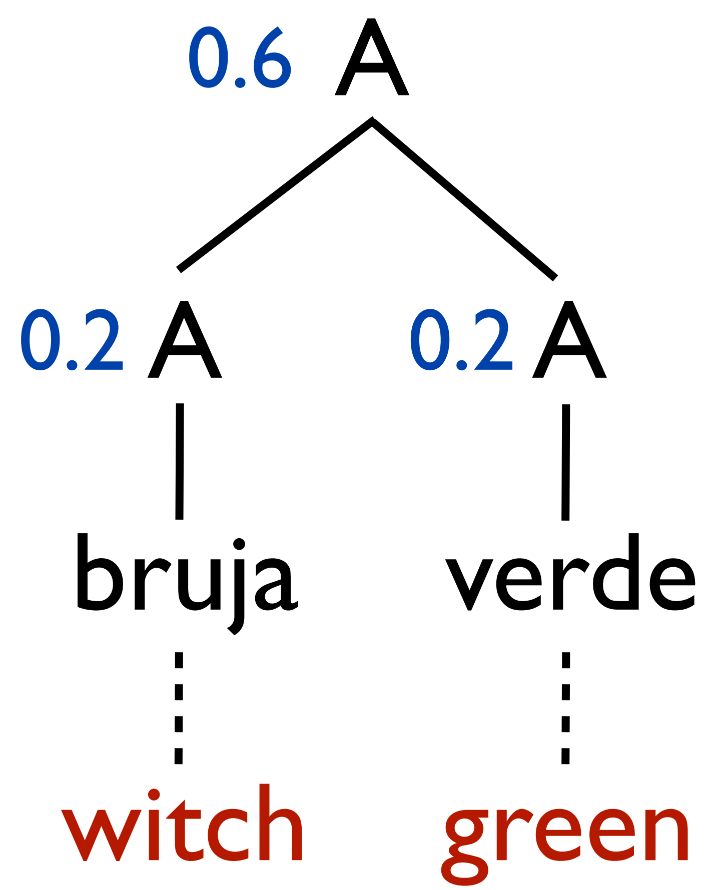
A → bruja | witch	0.2
A → verde | green	0.2
A → A A | 1 2	0.6
A → A A | 2 1	0.2



$0.6 \times 0.2 \times 0.2 = 0.024$

SCFGs: A problem

A → bruja | witch	0.2
A → verde | green	0.2
A → A A | 1 2	0.6
A → A A | 2 1	0.2



0.6x0.2x0.2 = 0.024

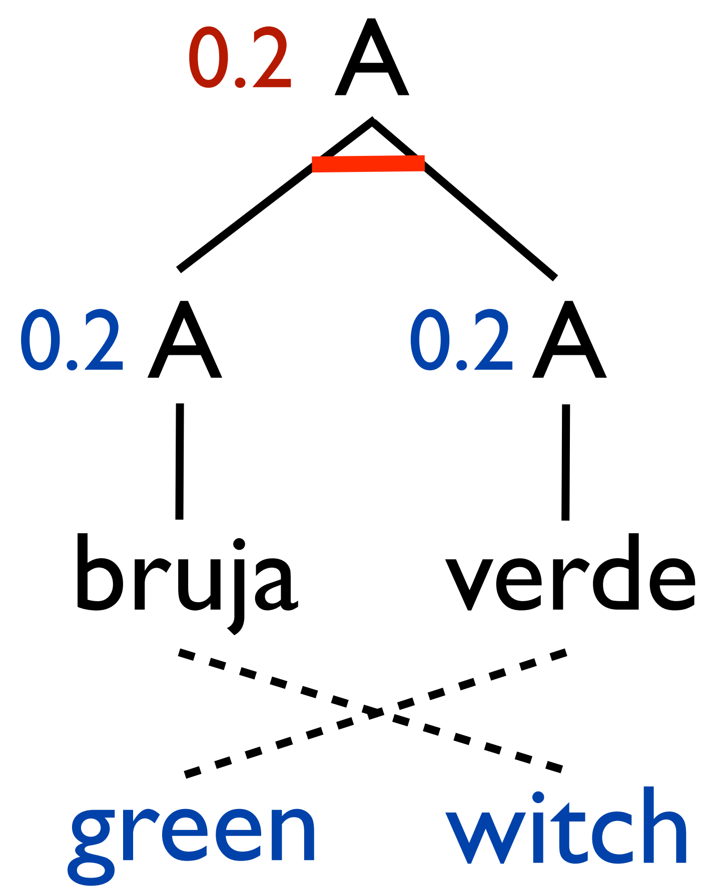
SCFGs: A problem

$A \rightarrow$ bruja | witch   0.2

$A \rightarrow$ verde | green   0.2

$A \rightarrow$ A A | 1 2   0.6

$A \rightarrow$ A A | 2 1   0.2



0.2x0.2x0.2 = 0.008

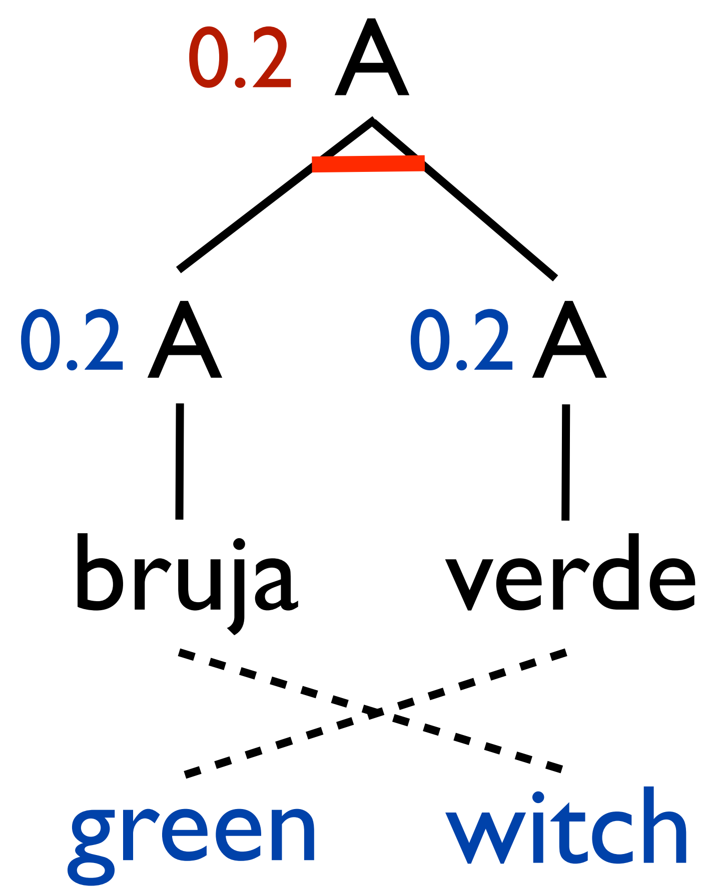
SCFGs: A problem

$A \rightarrow$ bruja | witch 0.2

$A \rightarrow$ verde | green 0.2

$A \rightarrow$ A A | 1 2 0.6

$A \rightarrow$ A A | 2 1 0.2



0.2x0.2x0.2 = 0.008

Context-free rules apply independent of context.

Some solutions

- More sophisticated grammars

  $A \rightarrow A\,A \mid 2\ 1$

Some solutions

- More sophisticated grammars

  ~~A → A A | 2 1~~

  NP → NN JJ | 2 1

Some solutions

- More sophisticated grammars

  ~~A → A A | 2 1~~

  NP → NN JJ | 2 1

  What are the problems with this?

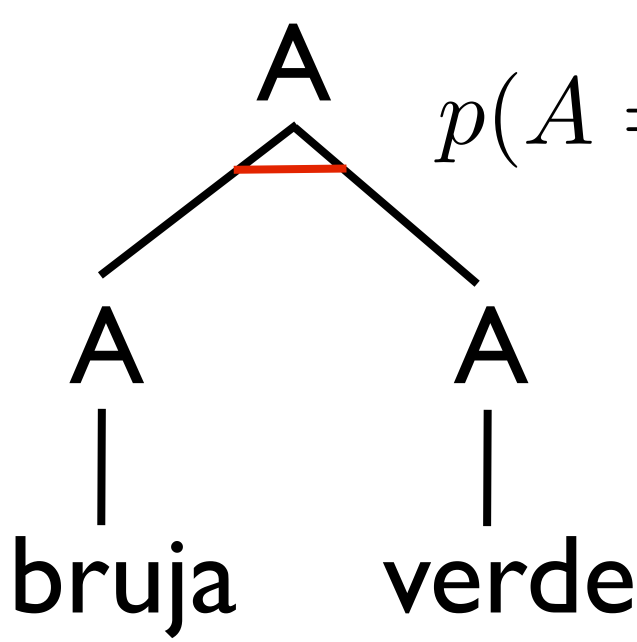
Some solutions

- More sophisticated grammars

  ~~A → A A | 2 1~~

  NP → NN JJ | 2 1

- Discriminative "parsing"



$$p(A = \{\leftarrow, \rightarrow\} \mid \mathbf{f}, A_{\mathrm{left}}, A_{\mathrm{right}})$$

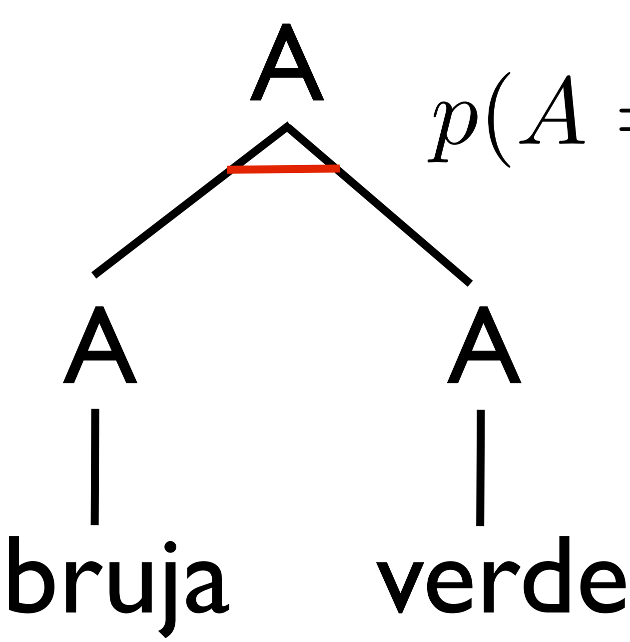
Some solutions

- More sophisticated grammars

  ~~A → A A | 2 1~~

  NP → NN JJ | 2 1

- Discriminative "parsing"



$$p(A = \{\leftarrow, \rightarrow\} \mid \mathbf{f}, A_{\text{left}}, A_{\text{right}})$$

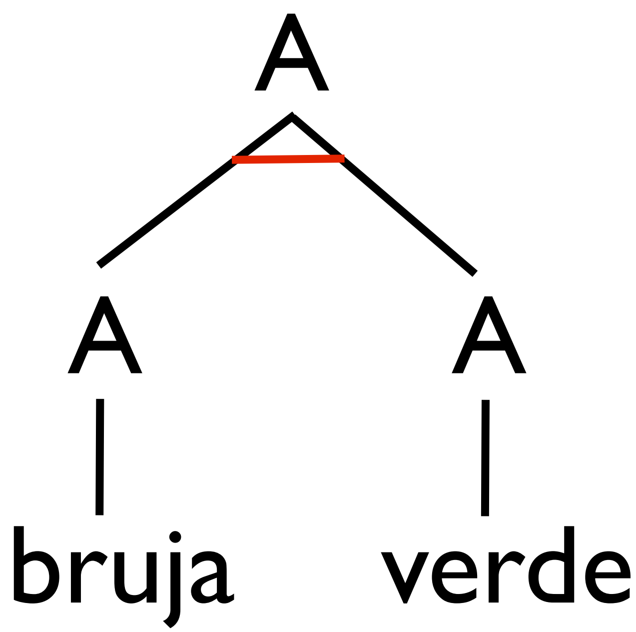
Source

Left child

Right child

Key Insight

- PCFGs are generative models of text (parallel text)
- In translation, the text is given: use discriminative models
- Xiong et al. propose a very simple approach:
  - Standard translation model (phrase based)
  - Standard (uniform) segmentation model
  - Standard *n*-gram language model
  - *Innovation*: every time you form a constituent, predict whether it should be **monotone** or **inverted**

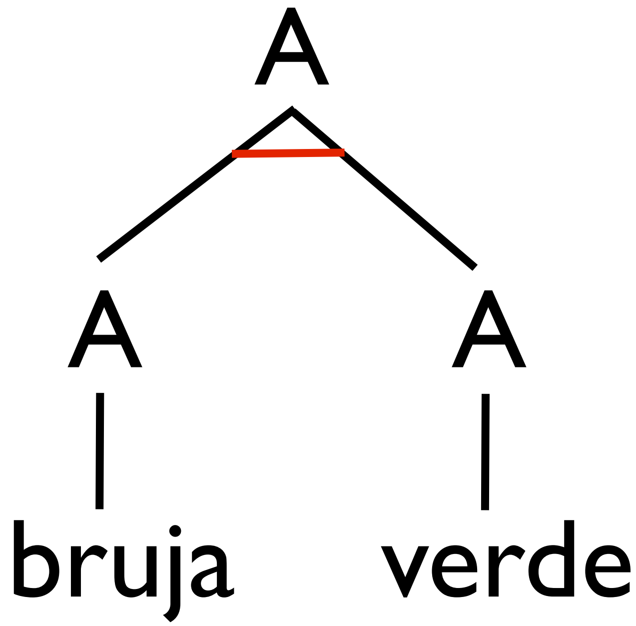


$$p(A = \leftarrow \mid \text{bruja verde}, (0,1), (1,2))$$

MaxEnt Model

$$p(A = \{\leftarrow, \rightarrow\} \mid \mathbf{f}, A_{\text{left}}, A_{\text{right}}) = \frac{1}{Z} \exp \boldsymbol{w}^\top \boldsymbol{\phi}(\mathbf{f}, A_{\text{left}}, A_{\text{right}})$$

Again, we reduce a major component of translation to **binary classification**.



$$p(A = \leftarrow \mid \text{bruja verde}, (0,1), (1,2))$$

This is a lot of conditioning

MaxEnt Model

$$p(A = \{\leftarrow, \rightarrow\} \mid \mathbf{f}, A_{\text{left}}, A_{\text{right}}) = \frac{1}{Z} \exp \boldsymbol{w}^\top \boldsymbol{\phi}(\mathbf{f}, A_{\text{left}}, A_{\text{right}})$$

Again, we reduce a major component of translation to **binary classification**.

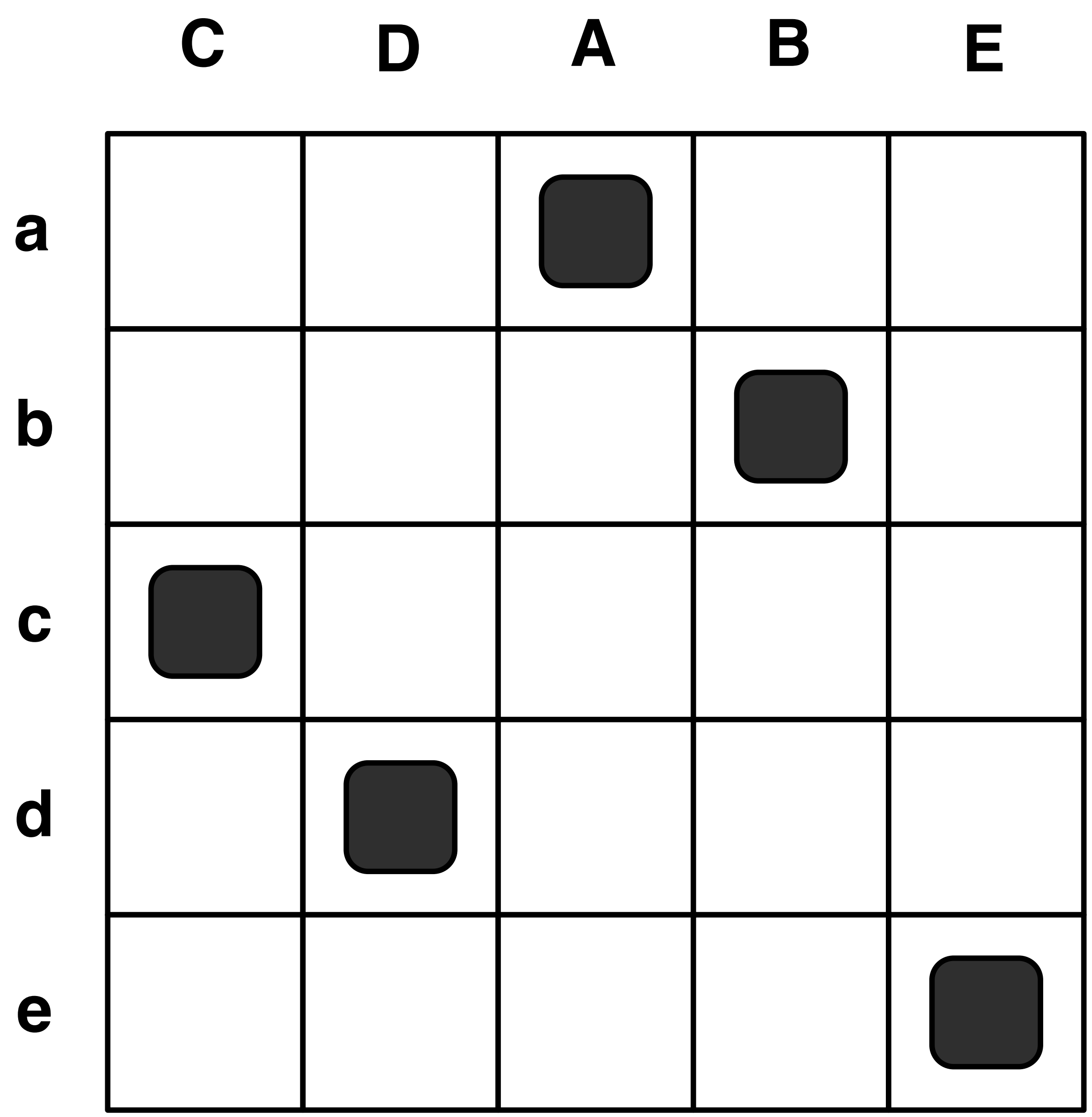
Training the Model

- What do we need to train the model?

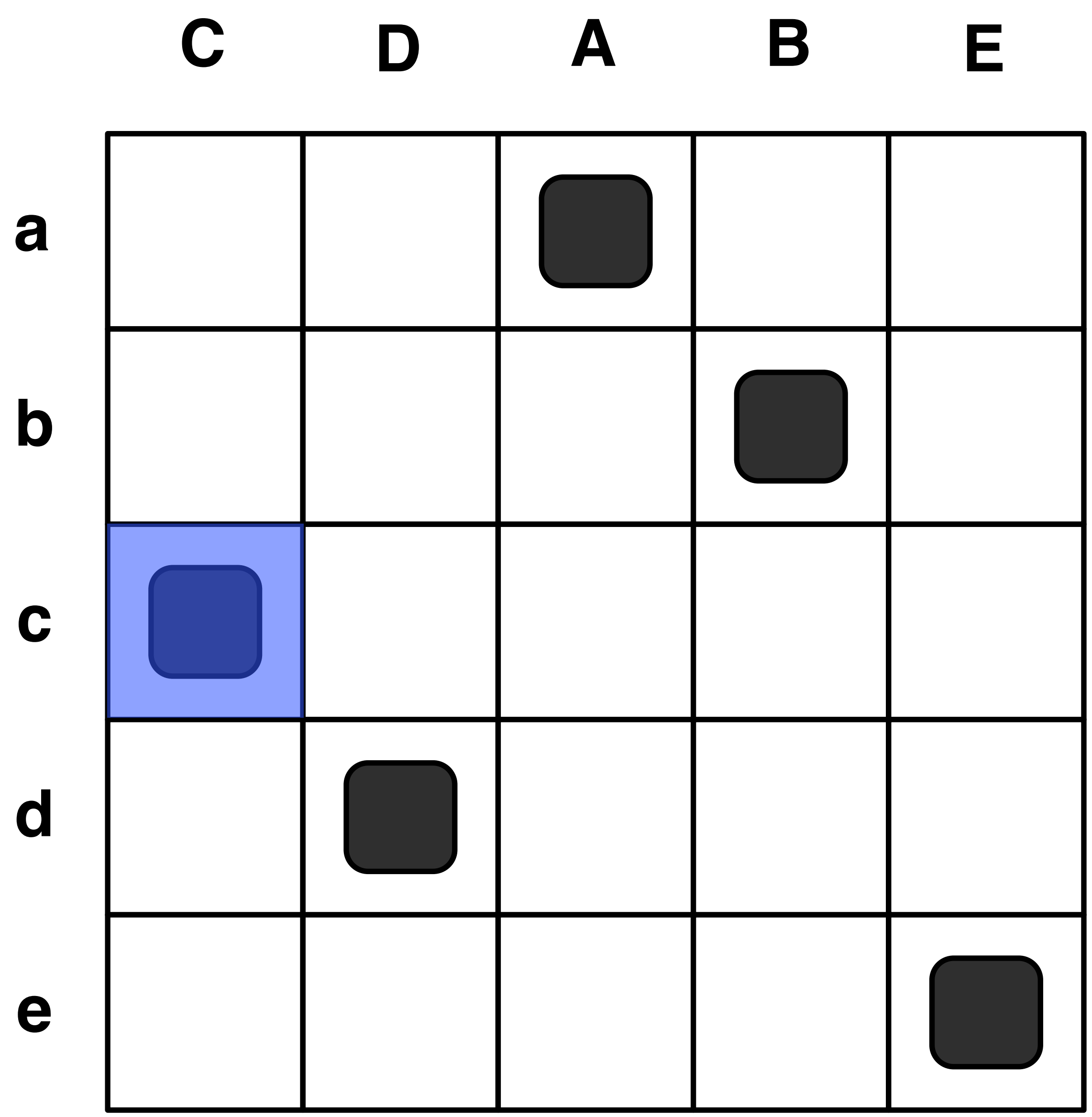
Training the Model

- What do we need to train the model?

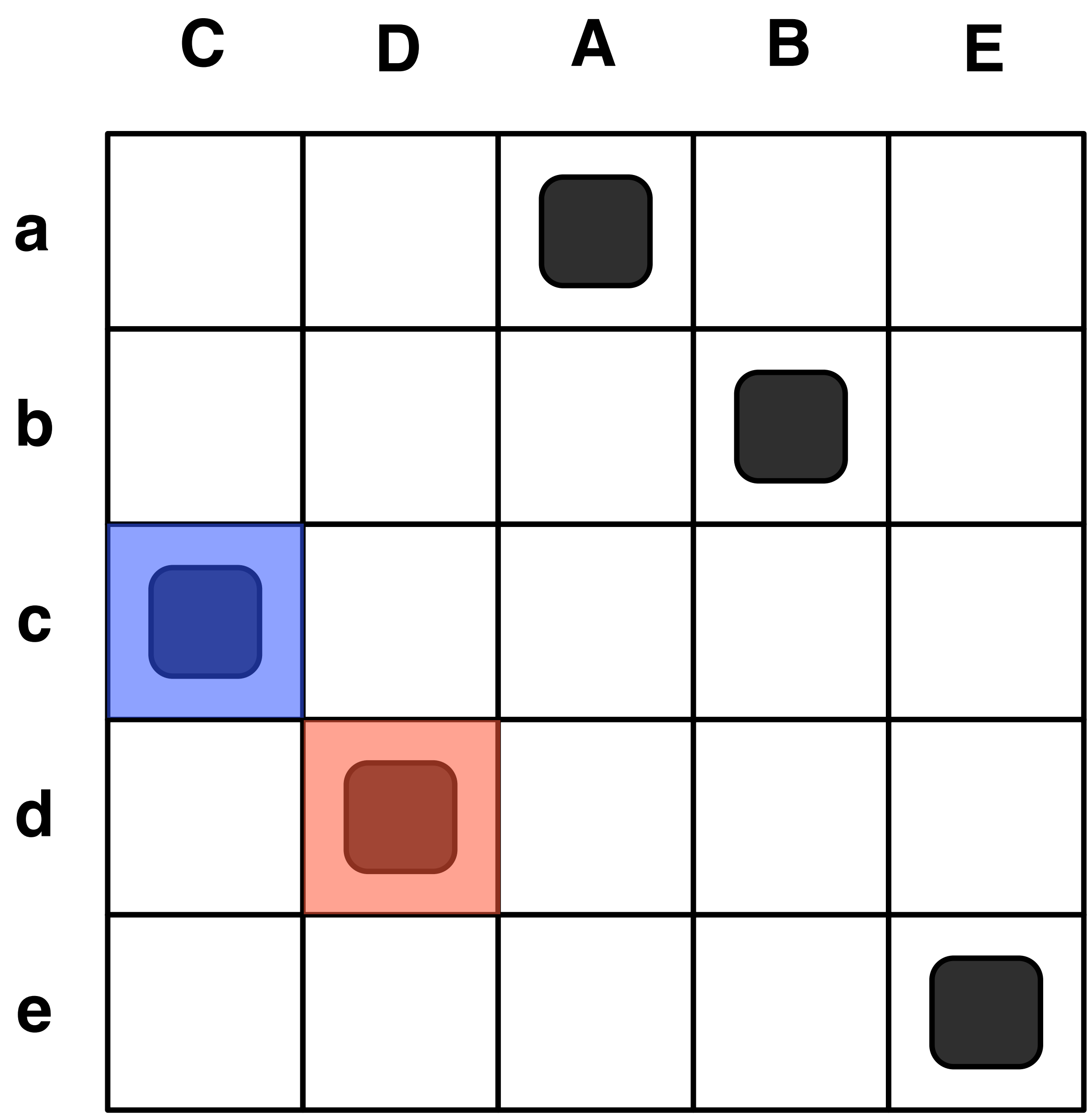
- How do we extract training examples from the training data?



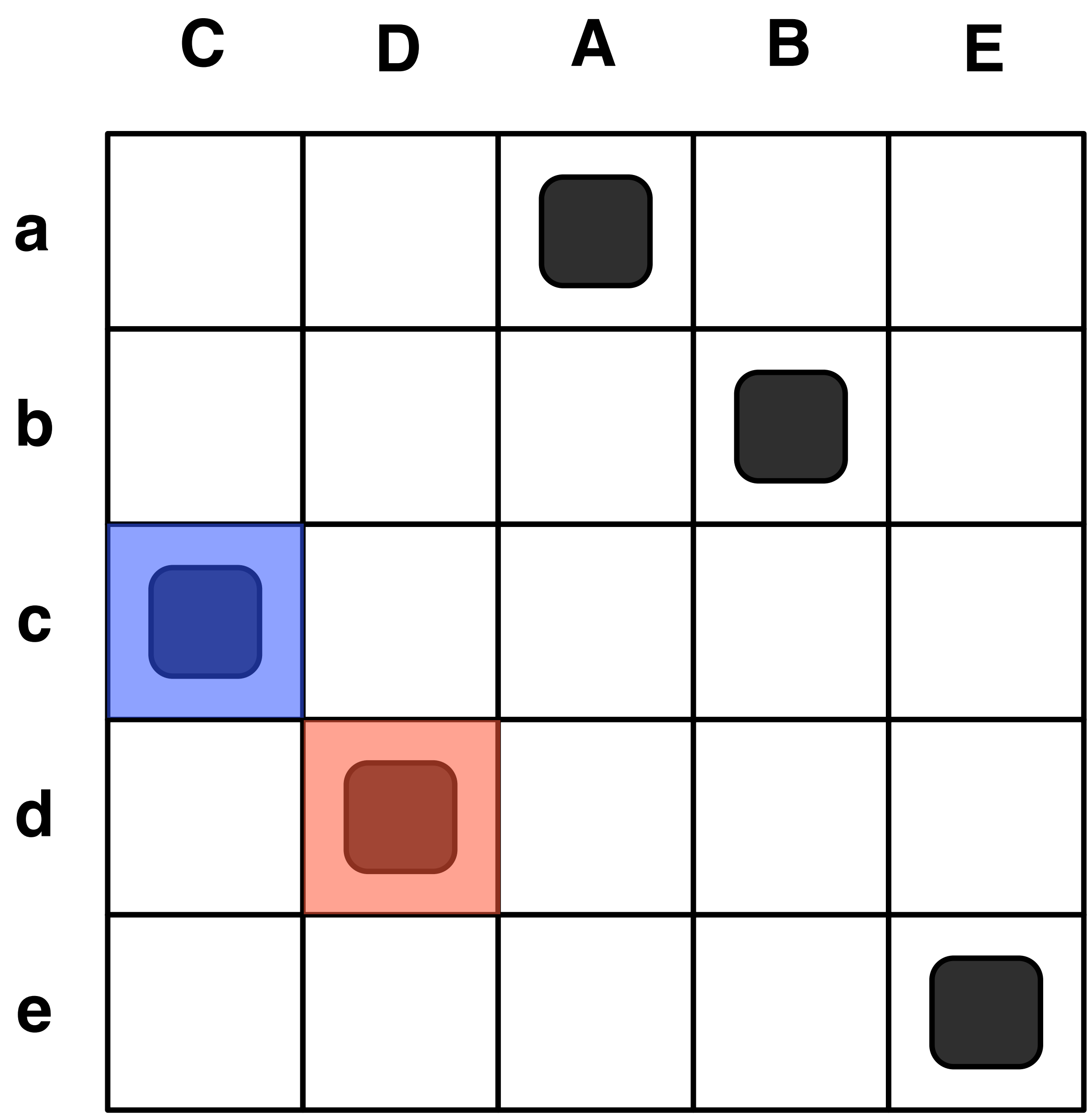
Extract examples from word aligned training data.



Extract examples from word aligned training data.

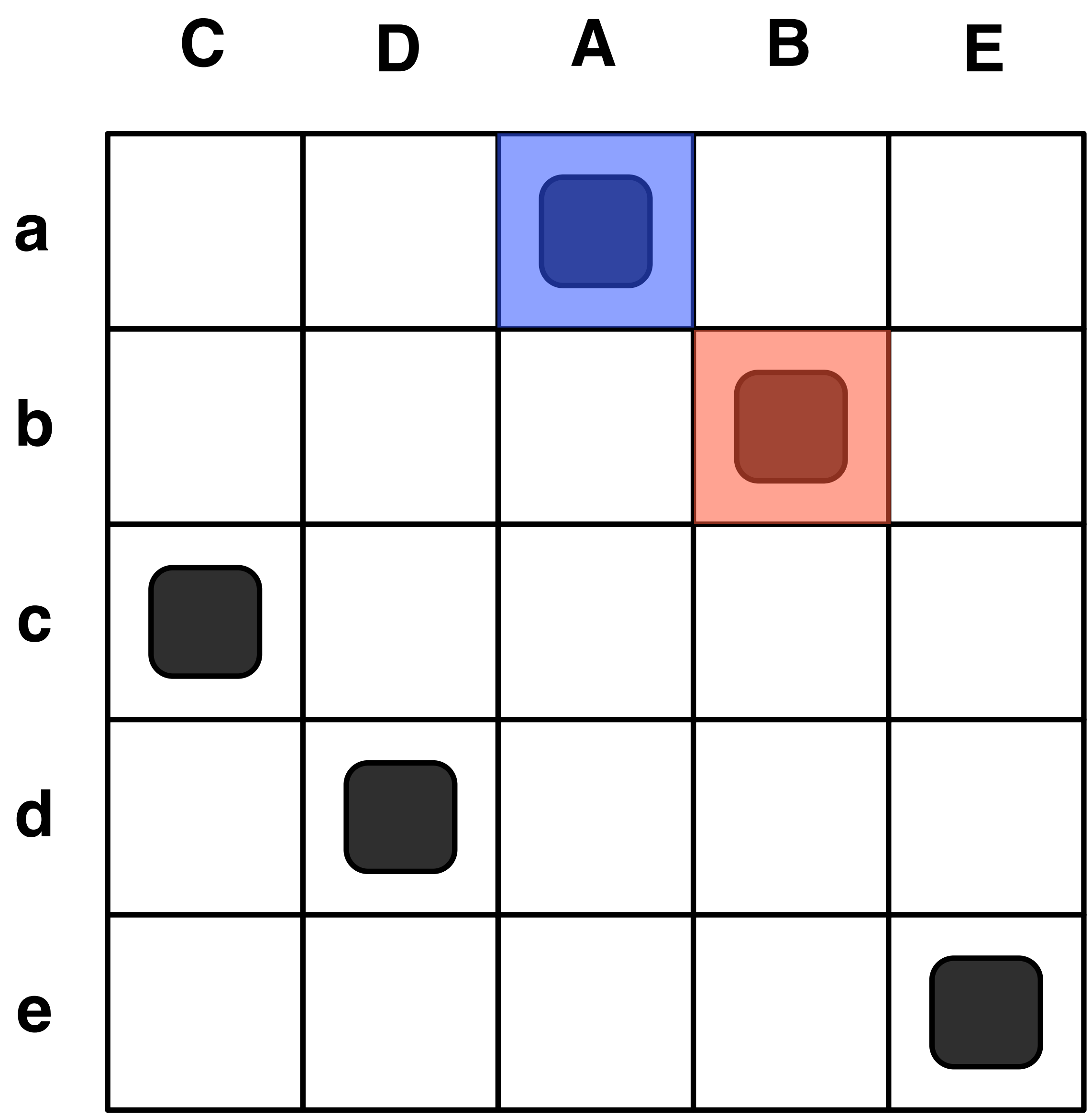


Extract examples from word aligned training data.



(m | a,b,c,d,e,2-3,3-4)

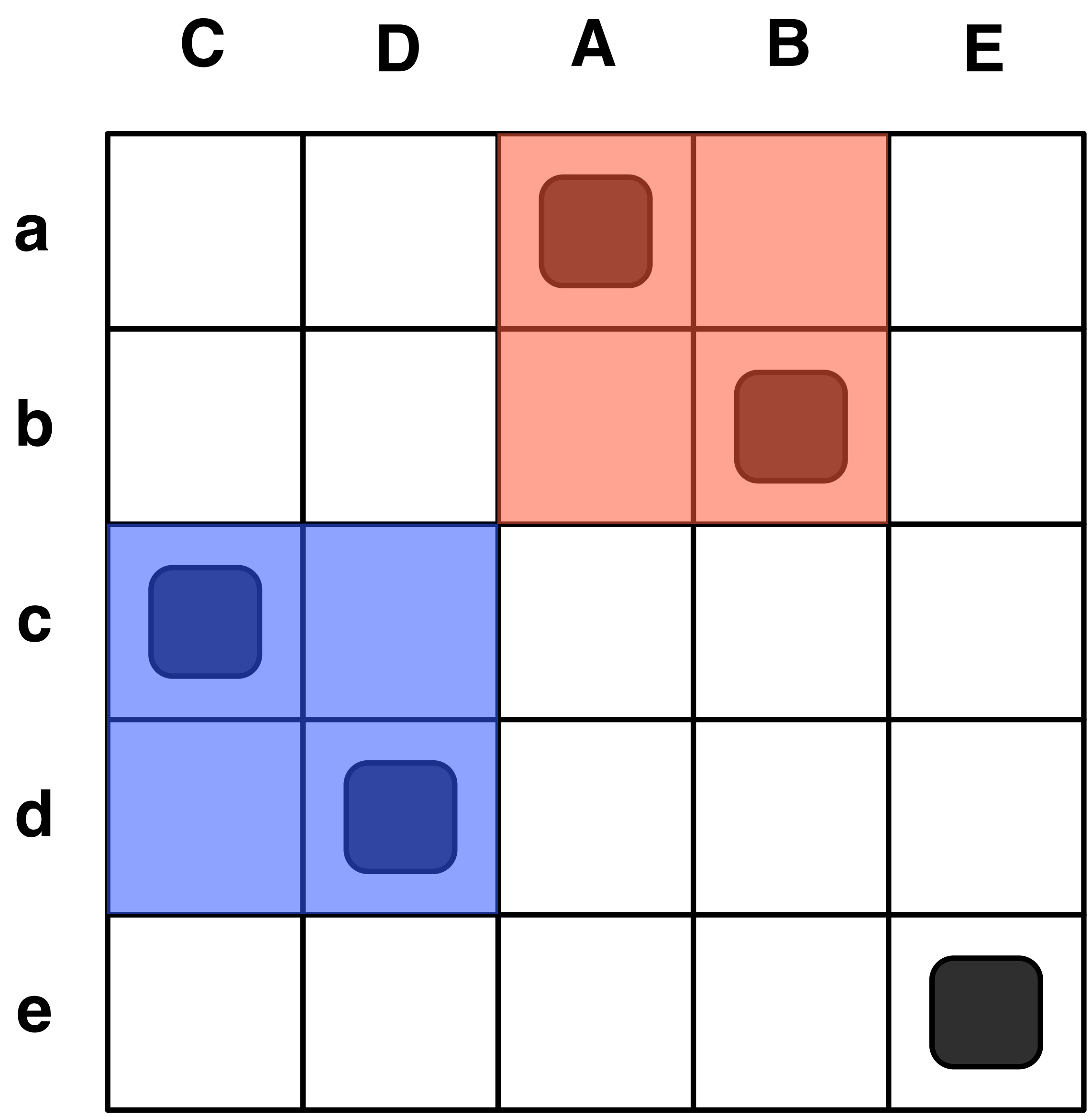
Extract examples from word aligned training data.



(m | a,b,c,d,e,2-3,3-4)

(m | a,b,c,d,e,0-1,1-2)

Extract examples from word aligned training data.

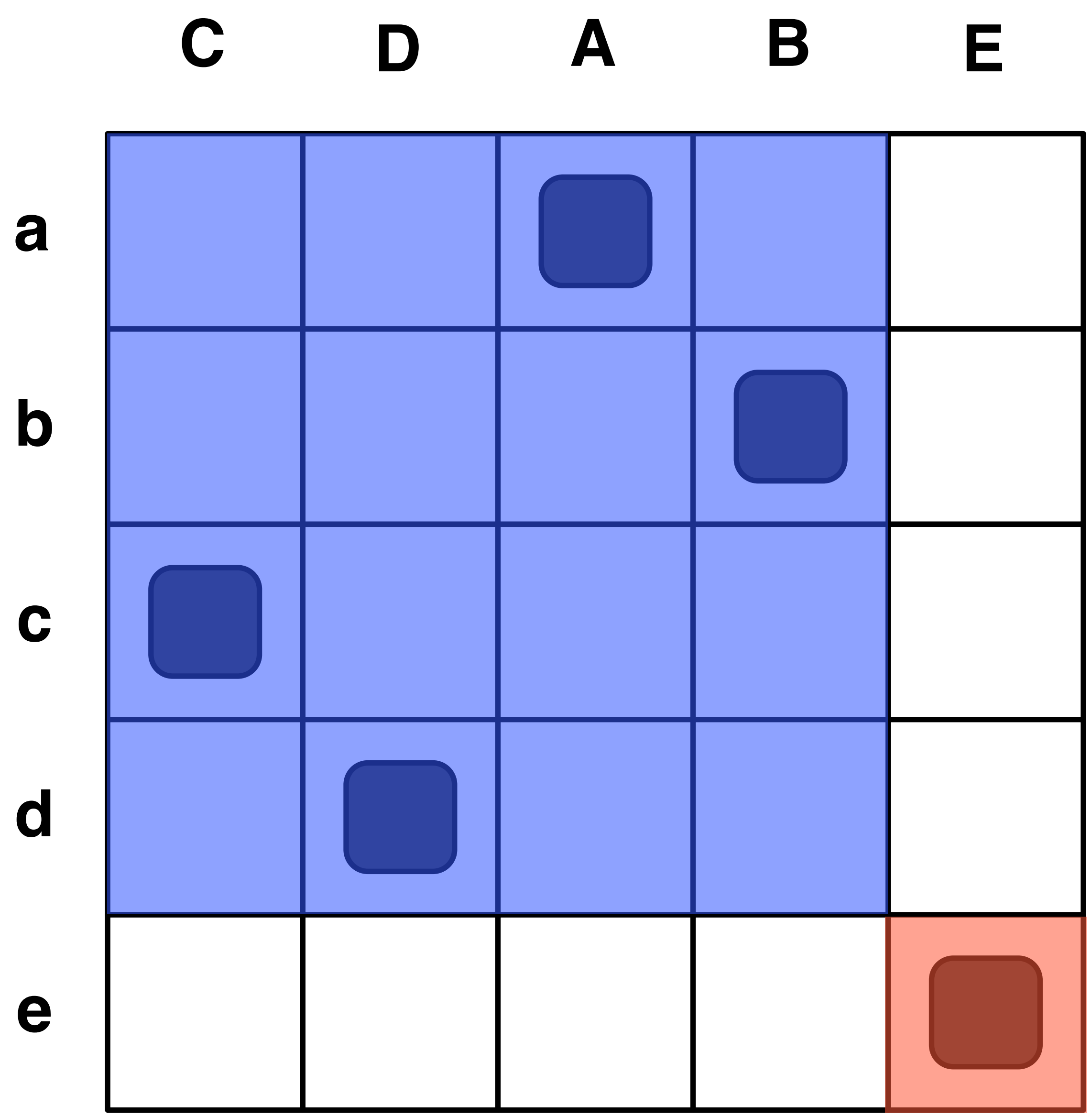


(m | a,b,c,d,e,2-3,3-4)

(m | a,b,c,d,e,0-1,1-2)

(i | a,b,c,d,e,2-4,0-2)

Extract examples from word aligned training data.



(m | a,b,c,d,e,2-3,3-4)

(m | a,b,c,d,e,0-1,1-2)

(i | a,b,c,d,e,2-4,0-2)

(m | a,b,c,d,e,0-4,4-5)

Extract examples from word aligned training data.

Results

Condition	BLEU
monotonic	20.1
no-model	19.6
size of constituent	20.9
MaxEnt	22.2

General Insights

- Decoders make local translation and reordering decisions
  - Standard approach: relative frequency
  - Alternative: using "local classifiers"
  - Challenge: extract (noisy) training instances from the training data
  - Benefits: no decoding required for training these local classifiers
- The source is given: **use it!**

Questions?

Modeling the Lexicon

bestehen	→	consist	0.5
	→	insist	0.4
	→	pass	0.1

Modeling the Lexicon

bestehen	consist	0.5
	insist	0.4
	pass	0.1

Es wird morgen eine Pruefung geben — *There's a test tomorrow.*

Ob ich bestehen werde? — *Will I pass?*

Modeling the Lexicon

bestehen	→ consist	0.5
	→ insist	0.4
	→ pass	0.1

Es wird morgen eine Pruefung geben — *There's a test tomorrow.*

Ob ich bestehen werde? — *Will I pass?*

Google *Whether I shall consist?*

Modeling the Lexicon

bestehen	consist	0.5
	insist	0.4
	pass	0.1

Es wird morgen eine Pruefung geben — *There's a test tomorrow.*

Ob ich bestehen werde? — *Will I pass?*

Google *Whether I shall consist?*

Goal:

$$p_{\text{new model}}(\text{pass} \mid \text{bestehen}, C = \text{Prüfung}) > p(w \mid \text{bestehen})$$

The DWL

- "Discriminative Word Lexicon"

$$p(\mathcal{E} \mid \mathcal{F})$$

Set of English words Set of Foreign words

The DWL

- "Discriminative Word Lexicon"

$$p(\mathcal{E} \mid \mathcal{F}) = \prod_{e \in \mathcal{E}} p(\text{contains } e \mid \mathcal{F}) \times$$
$$\prod_{e \in \mathcal{E}^C} (1 - p(\text{contains } e \mid \mathcal{F}))$$

Model inclusion as conditionally independent
binary decisions

Binary Classifiers

- Downside
  - Independence assumptions are harsh
- Upside
  - Training for every word in the vocabulary can be carried out in parallel

Training the Model

- What do we need to train the model?

- How do we extract training examples from the training data?

$$\Sigma = \{\text{the}, \text{and}, \text{of}, \text{cat}, \ldots, \text{pass}, \text{test}, \ldots \text{resulting}, \text{xylophone}\}$$

$$\Sigma = \{\text{the}, \text{and}, \text{of}, \text{cat}, \ldots, \text{pass}, \text{test}, \ldots \text{resulting}, \text{xylophone}\}$$

Sentence pair:

you will pass the test *du wirst die Pruefung bestehen*

$$\Sigma = \{\text{the}, \text{and}, \text{of}, \text{cat}, \ldots, \text{pass}, \text{test}, \ldots \text{resulting}, \text{xylophone}\}$$

Sentence pair:

you will pass the test *du wirst die Pruefung bestehen*

Classifier	y	Feature Vector (x)
pass?	+	du=1 wirst=1 Pruefung=1 bestehen=1
will?	+	du=1 wirst=1 Pruefung=1 bestehen=1
insist?	-	du=1 wirst=1 Pruefung=1 bestehen=1
insist?	-	du=1 wirst=1 Pruefung=1 bestehen=1
cat?	-	du=1 wirst=1 Pruefung=1 bestehen=1
xylophone?	-	du=1 wirst=1 Pruefung=1 bestehen=1

$$\Sigma = \{\text{the}, \text{and}, \text{of}, \text{cat}, \ldots, \text{pass}, \text{test}, \ldots \text{resulting}, \text{xylophone}\}$$

Sentence pair:

you will pass the test *du wirst die Pruefung bestehen*

Classifier	y	Feature Vector (x)
pass?	+	du=1 wirst=1 Pruefung=1 bestehen=1
will?	+	du=1 wirst=1 Pruefung=1 bestehen=1
insist?	-	du=1 wirst=1 Pruefung=1 bestehen=1
insist?	-	du=1 wirst=1 Pruefung=1 bestehen=1
cat?	-	du=1 wirst=1 Pruefung=1 bestehen=1
xylophone?	-	du=1 wirst=1 Pruefung=1 bestehen=1

O(N x V) training instances

Rescoring with the DWL

- The DWL assigns probabilities to **sets** of words
  - Once a word is used once, subsequent uses are "free"
  - This makes dynamic programming difficult
- A simple strategy: reranking
  - Get *k*-best lists from baseline decoder, compute DWL score on each entry
  - Train a second model (using PRO, MERT, etc.) as if the *k*-best lists were the decoder
  - Search errors are very possible!

Arabic-English

Condition	BLEU
Baseline	42.0
+DWL	42.4

Chinese-English

Condition	BLEU
Baseline	25.3
+DWL	26.2

Source	目前 , 事故 抢险 组 正在 紧急 恢复 通风 系统 .
Baseline	at present, the accident and rescue teams are currently emergency re-covery ventilation systems.
DWL	at present, the emergency rescue teams are currently restoring the ventilation system.

Reference	right now, the accident emergency rescue team is making emergency repair on the ventilation system.

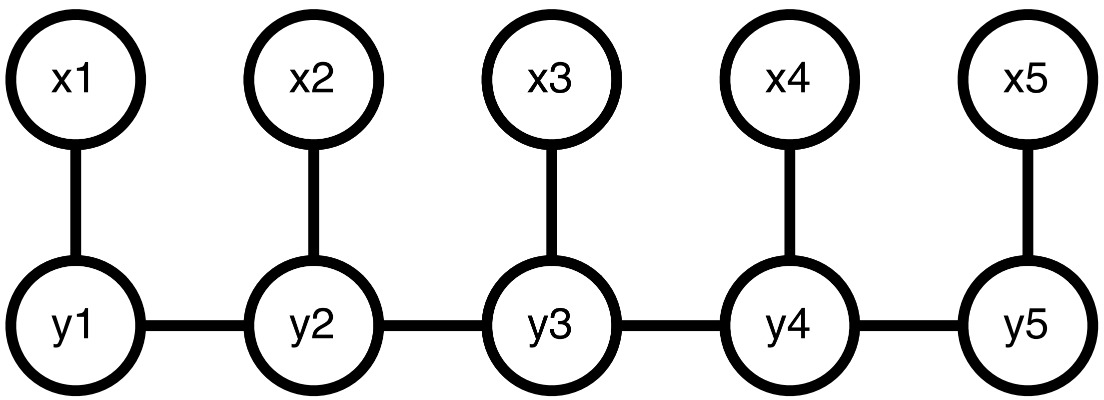
DWL	
emergency	0.894
currently	0.330
current	0.175
emergencies	0.133
present	0.133
accident	0.119
recovery	0.053
group	0.046
dealing	0.042
ventilation	0.034

Possible Extensions

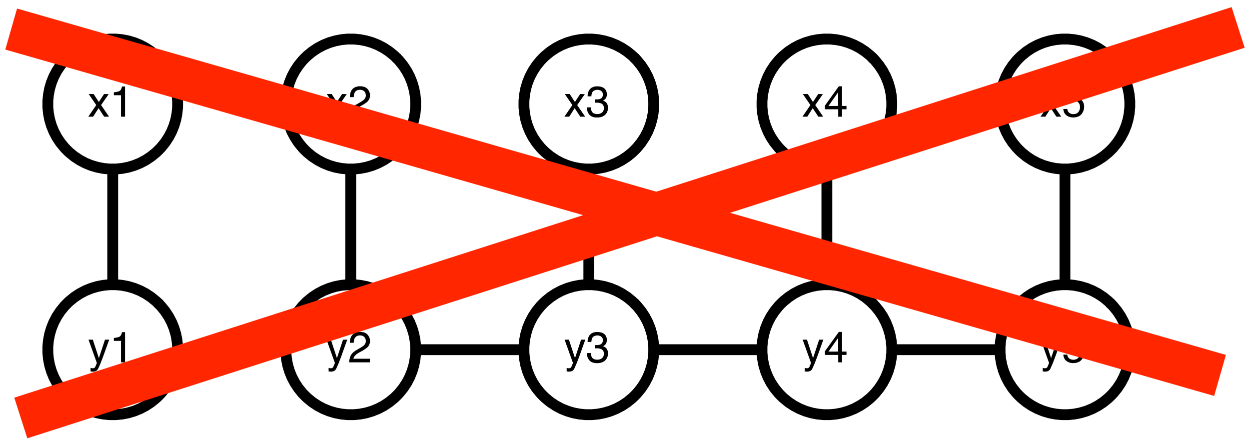
- Condition on more context than the sentence (e.g., document)
- Model units larger than words (e.g., phrases)
- Model only words/phrases that have ambiguous translations
  - Measure of ambiguity: **entropy**

Questions?

Translation as CRFs



Translation as CRFs

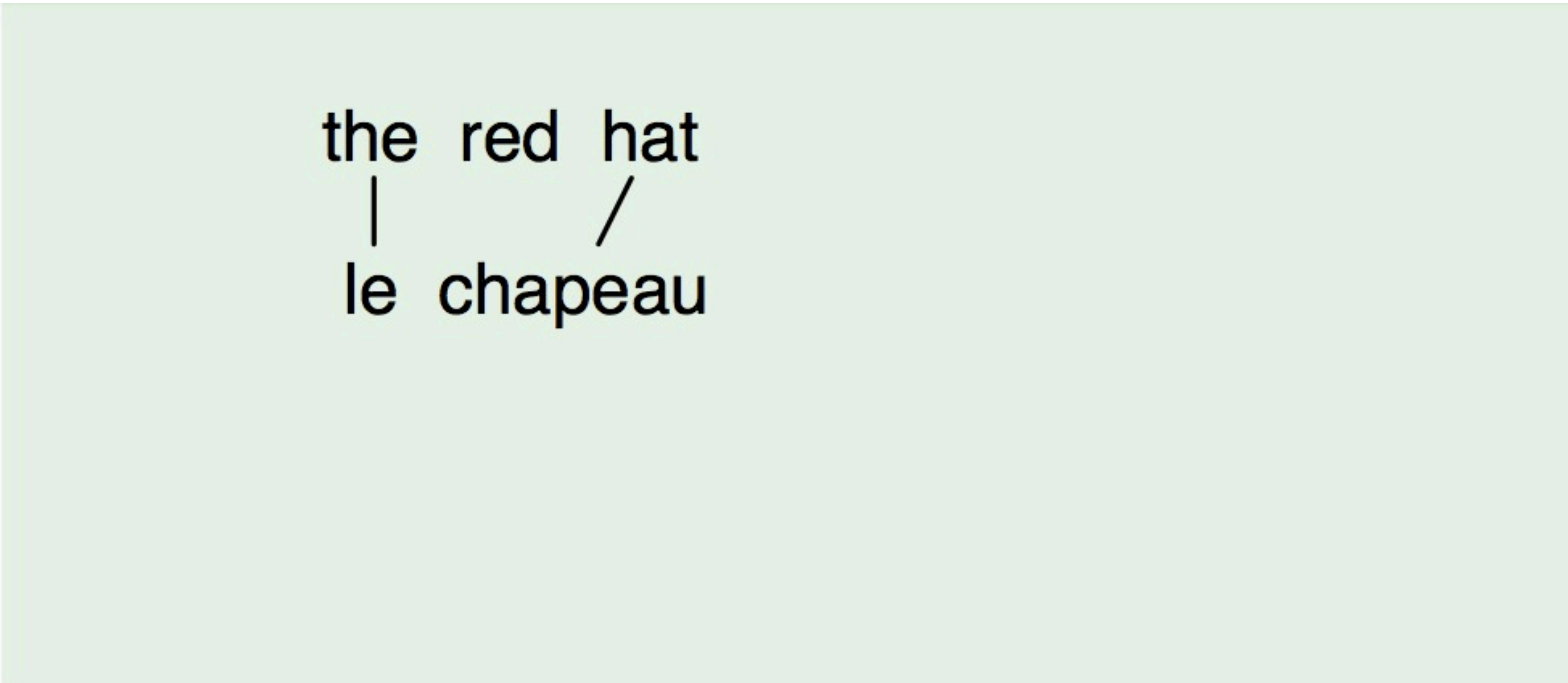


No linear chains

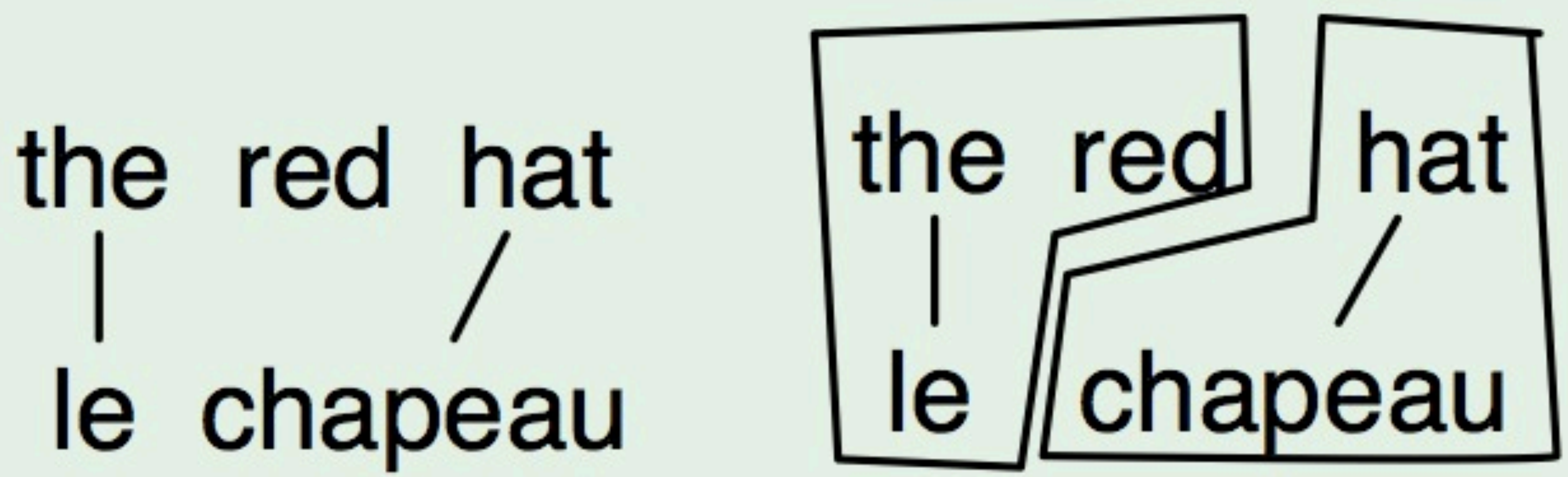
CRFs on Synchronous Trees

- Model: discriminate "good" parses under an SCFG from bad ones
- Challenges:
  - What is a good parse?
    - One that produces a good translation
  - What is a good translation?
    - The one we see in the training data
  - What about the zillions of ways to derive a sentence pair?
    - Let's marginalize them
  - What about non-literal translations?
    - Regularize so we don't memorize "bad" stuff

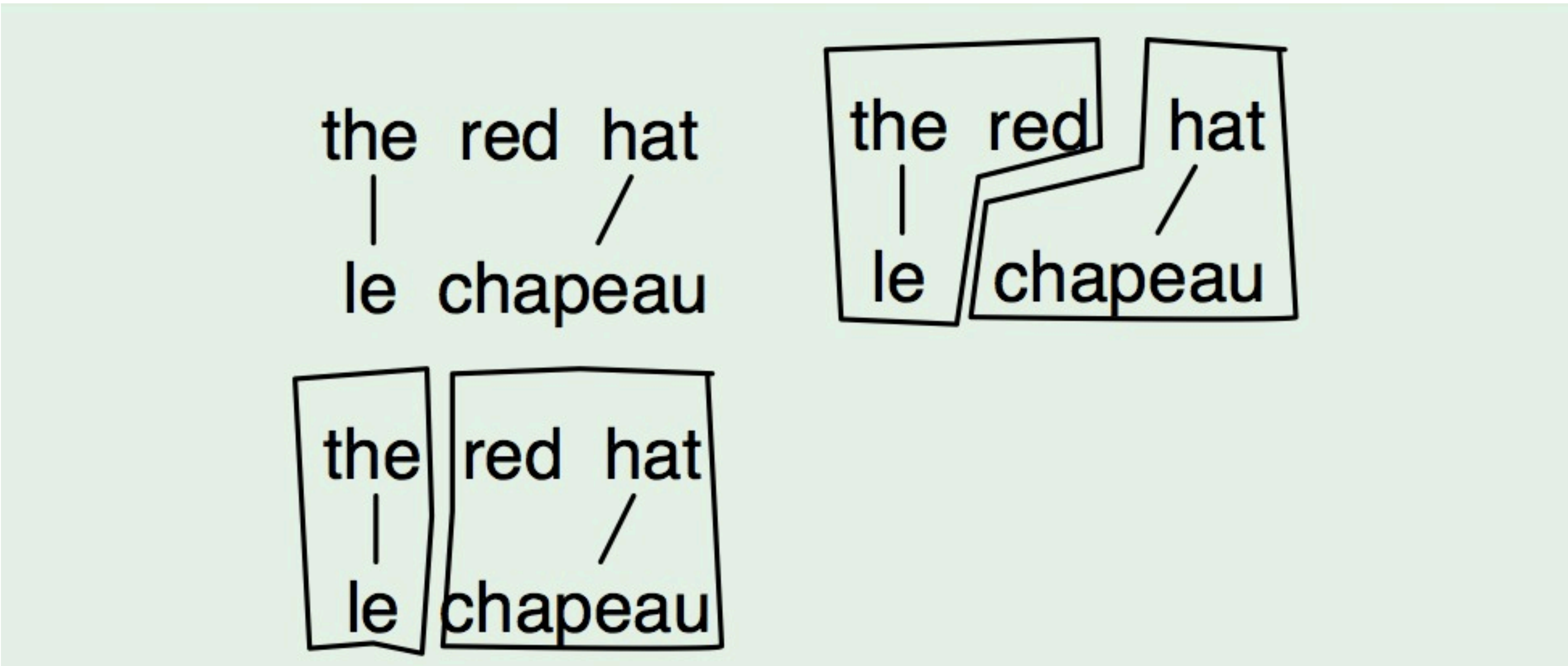
Which Derivation to Optimize?



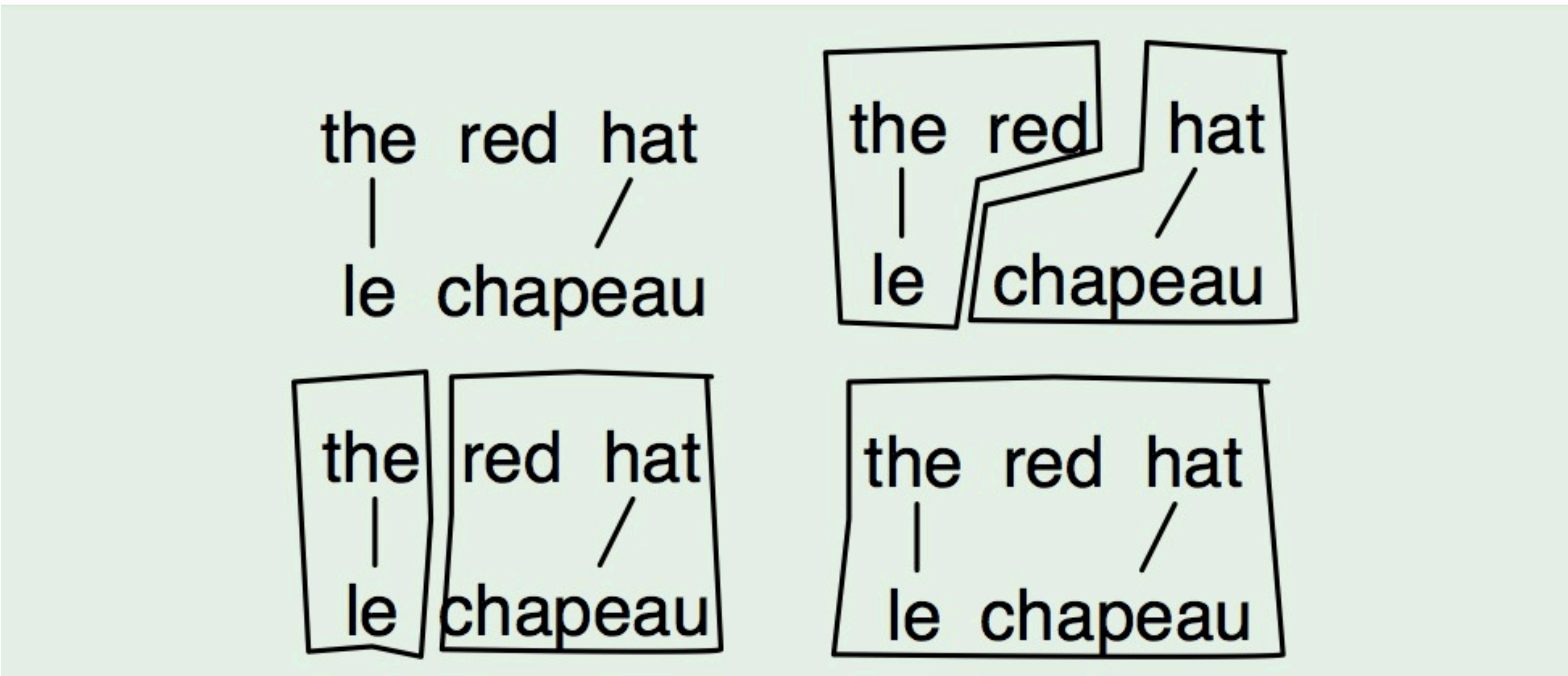
Which Derivation to Optimize?



Which Derivation to Optimize?



Which Derivation to Optimize?



Parametric Form

Conditional probability of a derivation

$$p_\Lambda(\mathbf{d}, \mathbf{e}|\mathbf{f}) = \frac{\exp \sum_k \lambda_k H_k(\mathbf{d}, \mathbf{e}, \mathbf{f})}{Z_\Lambda(\mathbf{f})}.$$

Conditional probability of a translation

Features

The features must decompose with the rules:

$$H_k(\mathbf{d}, \mathbf{e}, \mathbf{f}) = \sum_{r \in \mathbf{d}} h_k(\mathbf{f}, r, q(r, \mathbf{d}))$$

- Any part of the **source** may be used
  - Source syntax
  - Morphology
  - Lexical context
  - POS information

Training

$$\mathcal{L} = \sum_{(\mathbf{e},\mathbf{f})\in\mathcal{D}} \sum_{\mathbf{d}} \log p_\Lambda(\mathbf{e},\mathbf{d} \mid \mathbf{f}) + \sum_m \frac{\lambda_m^2}{2\sigma^2}$$

Training

$$\mathcal{L} = \sum_{(\mathbf{e},\mathbf{f})\in\mathcal{D}} \sum_{\mathbf{d}} \log p_\Lambda(\mathbf{e},\mathbf{d} \mid \mathbf{f}) + \sum_m \frac{\lambda_m^2}{2\sigma^2}$$

Differentiable:

$$\frac{\partial \mathcal{L}}{\partial w_i} = \sum_{(\mathbf{e},\mathbf{f})\in\mathcal{D}} \mathbb{E}_{p_\Lambda(\mathbf{e},\mathbf{d}|\mathbf{f})} h_i(\mathbf{e},\mathbf{d},\mathbf{f}) - \mathbb{E}_{p_\Lambda(\mathbf{d}|\mathbf{e},\mathbf{f})} h_i(\mathbf{e},\mathbf{d},\mathbf{f}) - \frac{\lambda_i}{\sigma^2}$$

Inference

- How do we compute the following feature expectations?

$$\frac{\partial \mathcal{L}}{\partial w_i} = \sum_{(\mathbf{e},\mathbf{f}) \in \mathcal{D}} \mathbb{E}_{p_\Lambda(\mathbf{e},\mathbf{d}|\mathbf{f})} h_i(\mathbf{e},\mathbf{d},\mathbf{f}) - \mathbb{E}_{p_\Lambda(\mathbf{d}|\mathbf{e},\mathbf{f})} h_i(\mathbf{e},\mathbf{d},\mathbf{f}) - \frac{\lambda_i}{\sigma^2}$$

Effect of Regularization

Grammar Rules	ML ($\sigma^2 = \infty$)	MAP ($\sigma^2 = 1$)
$\langle X \rangle \rightarrow \langle carte,\ map \rangle$	1.0	0.5
$\langle X \rangle \rightarrow \langle carte,\ notice \rangle$	0.0	0.5
$\langle X \rangle \rightarrow \langle sur,\ on \rangle$	1.0	1.0
$\langle X \rangle \rightarrow \langle la,\ the \rangle$	1.0	1.0
$\langle X \rangle \rightarrow \langle table,\ table \rangle$	1.0	0.5
$\langle X \rangle \rightarrow \langle table,\ chart \rangle$	0.0	0.5
$\langle X \rangle \rightarrow \langle carte\ sur,\ notice\ on \rangle$	1.0	0.5
$\langle X \rangle \rightarrow \langle carte\ sur,\ map\ on \rangle$	0.0	0.5
$\langle X \rangle \rightarrow \langle sur\ la,\ on\ the \rangle$	1.0	1.0
$\langle X \rangle \rightarrow \langle la\ table,\ the\ table \rangle$	0.0	0.5
$\langle X \rangle \rightarrow \langle la\ table,\ the\ chart \rangle$	1.0	0.5

Training data:

carte sur la table $\leftrightarrow$ map on the table

carte sur la table $\leftrightarrow$ notice on the chart

Condition	BLEU
Hiero -LM	28.1
Hiero +LM	32.0
CRF - max deriv	25.8
CRF - max trans	27.7

Questions?