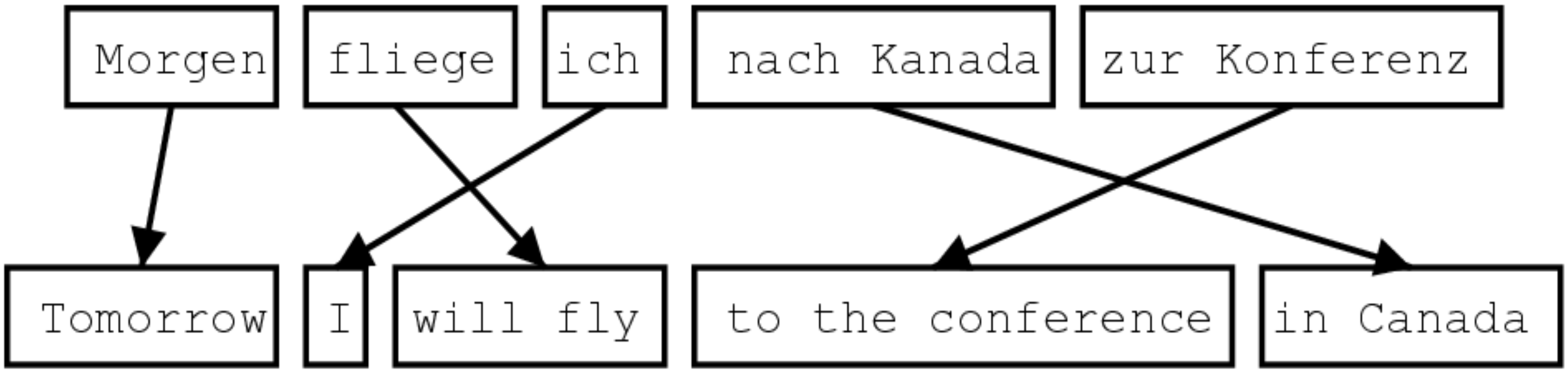
CSEP 517
Natural Language Processing
Autumn 2013

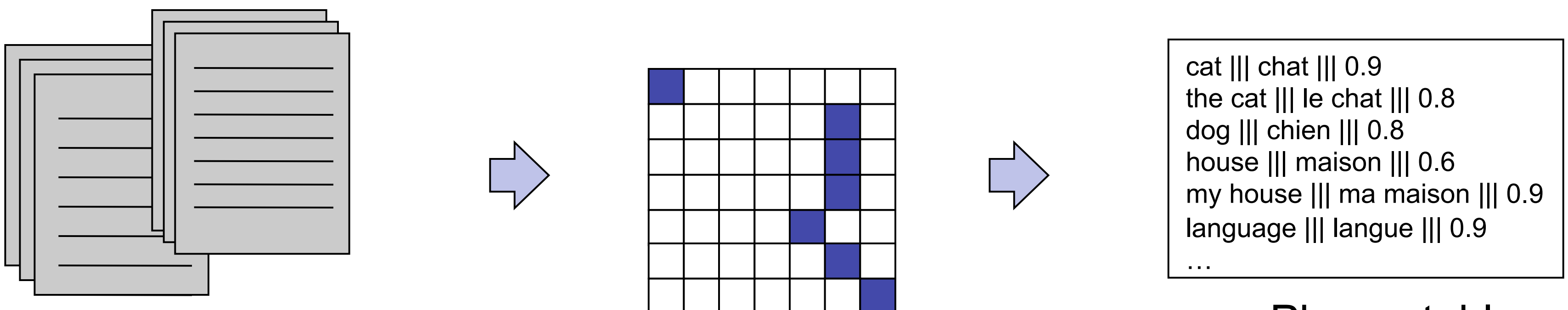
Phrase Based Translation

Luke Zettlemoyer

Slides from Philipp Koehn and Dan Klein

Phrase-Based Systems





Sentence-aligned corpus

Word alignments

Phrase table (translation model)

Phrase Translation Tables

- Defines the space of possible translations
  - each entry has an associated "probability"
- One learned example, for "den Vorschlag" from Europarl data

English	$\phi(\bar{e}\|\bar{f})$	English	$\phi(\bar{e}\|\bar{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

  - This table is noisy, has errors, and the entries do not necessarily match our linguistic intuitions about consistency….

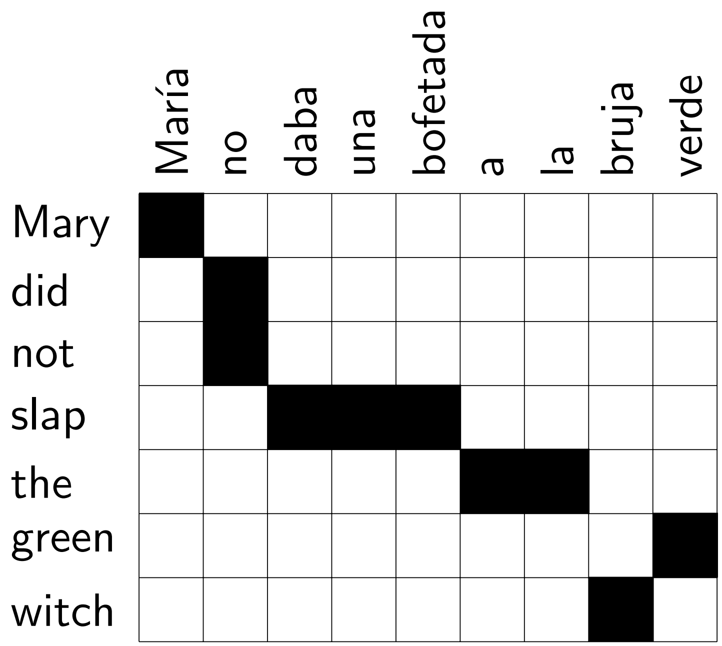
Phrase-Based Decoding

这	7人	中包括	来自	法国	和	俄罗斯	的	宇航	员	.
the	7 people	including	by some		**and**	the russian	**the**	the astronauts		,
it	7 people included		by france		and the	the russian		international astronautical	of rapporteur .	
this	7 out	including the	**from**	the french	and the russian		the fifth		.	
these	7 among	including from		the french and		of the russian	of	space	members	.
that	7 persons	including from the		of france	and to	russian	of the	aerospace	members .	
	7 include		from the	of france and		russian		**astronauts**		. the
	7 numbers include		**from france**		and russian		of astronauts who			. "
	7 populations include		those from france		and russian			astronauts .		
	7 deportees included		come from	**france**	**and russia**		in	astronautical	personnel	;
	7 philtrum	including those from		**france and**		**russia**	a space		**member**	
		including representatives from		france and the		**russia**		astronaut		
		include	came from	**france and russia**			by cosmonauts			
		include representatives from		french	**and russia**			cosmonauts		
		include	came from france		and russia 's			cosmonauts .		
		includes	coming from	french and		russia 's		cosmonaut		
				french and russian			's	astronavigation	member .	
				french	**and russia**		**astronauts**			
					and russia 's				special rapporteur	
					, and	**russia**			rapporteur	
					, and russia				rapporteur .	
					, and russia					
					or	russia 's				

Decoder design is important: [Koehn et al. 03]

Extracting Phrases

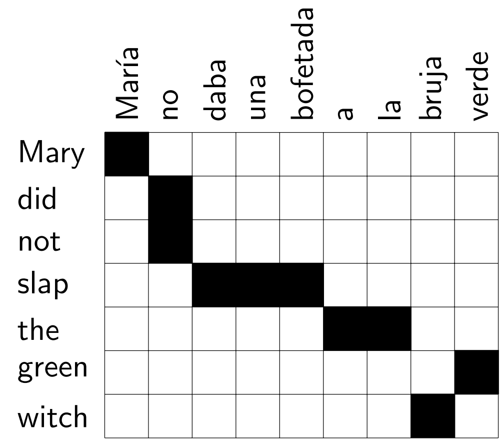
- We will use word alignments to find phrases

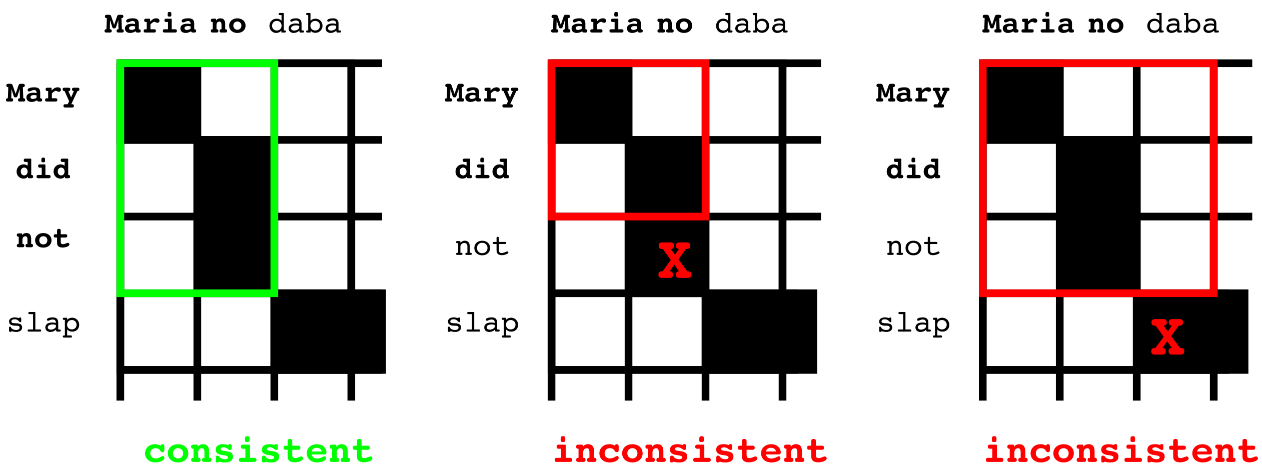


- Question: what is the best set of phrases?

Extracting Phrases

- Phrase alignment must
  - Contain at least one alignment edge
  - Contain all alignments for phrase pair





- Extract all such phrase pairs!

Phrase Pair Extraction Example

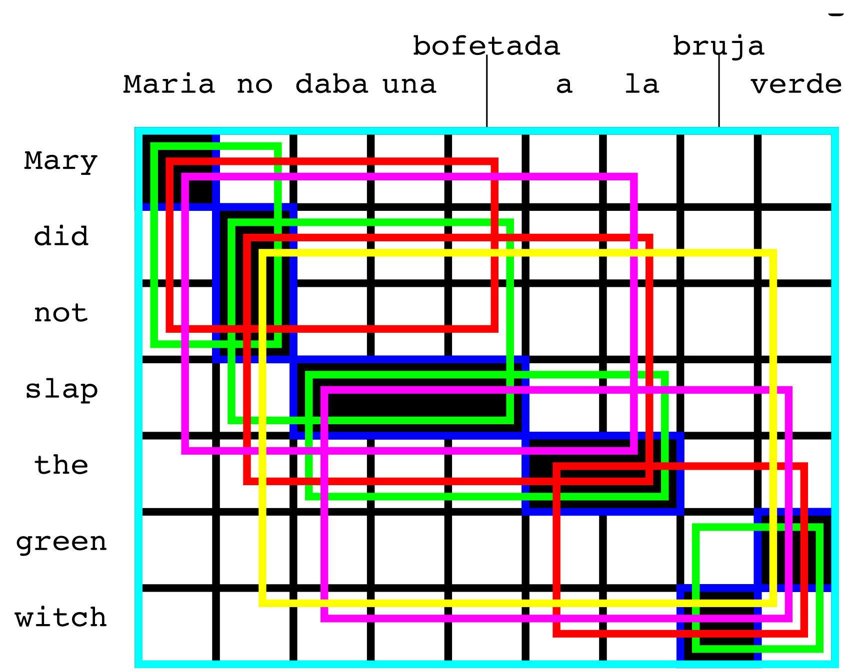
(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)

(Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch)

(Maria no daba una bofetada, Mary did not slap), (no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch)

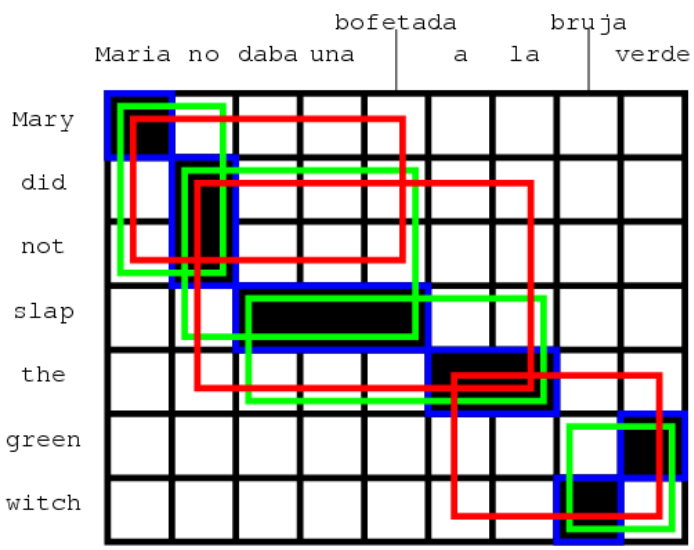
(Maria no daba una bofetada a la, Mary did not slap the), (daba una bofetada a la bruja verde, slap the green witch)

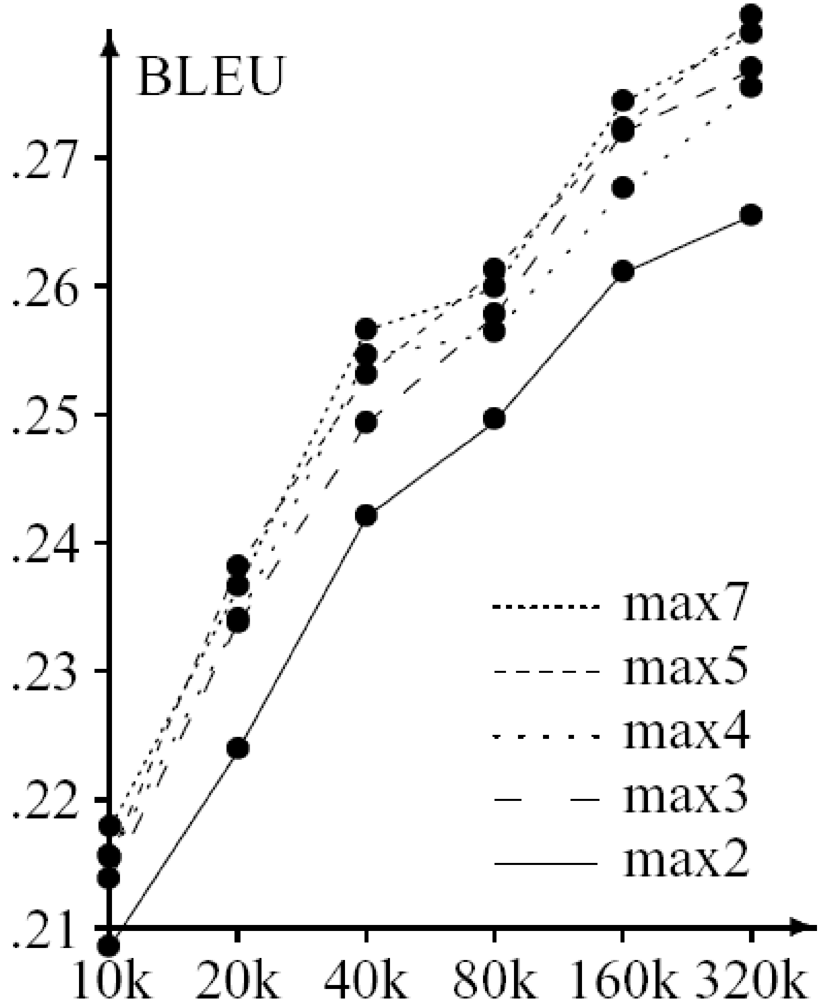
(Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)



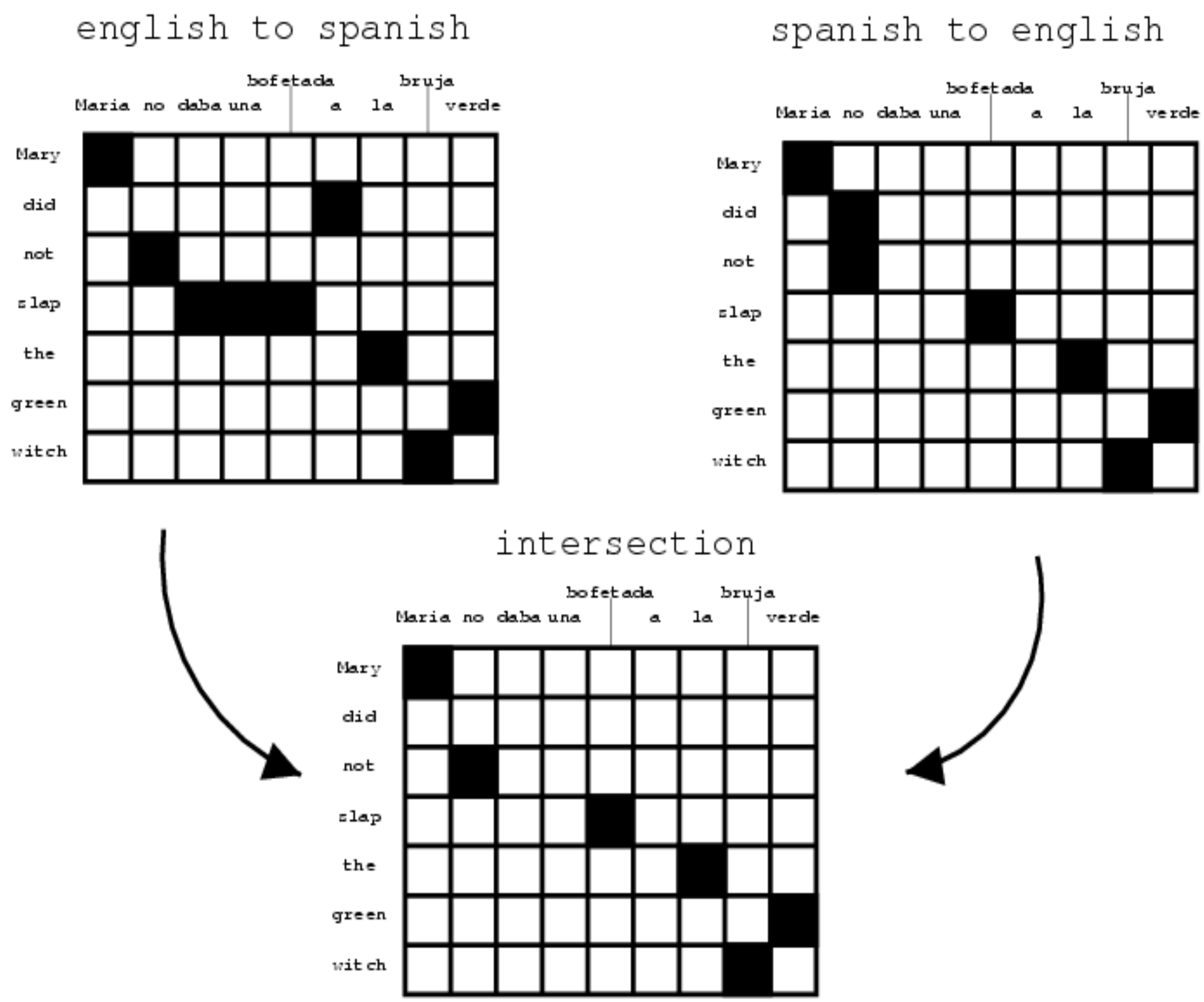
Phrase Size

- Phrases do help
  - But they don't need to be long
  - Why should this be?

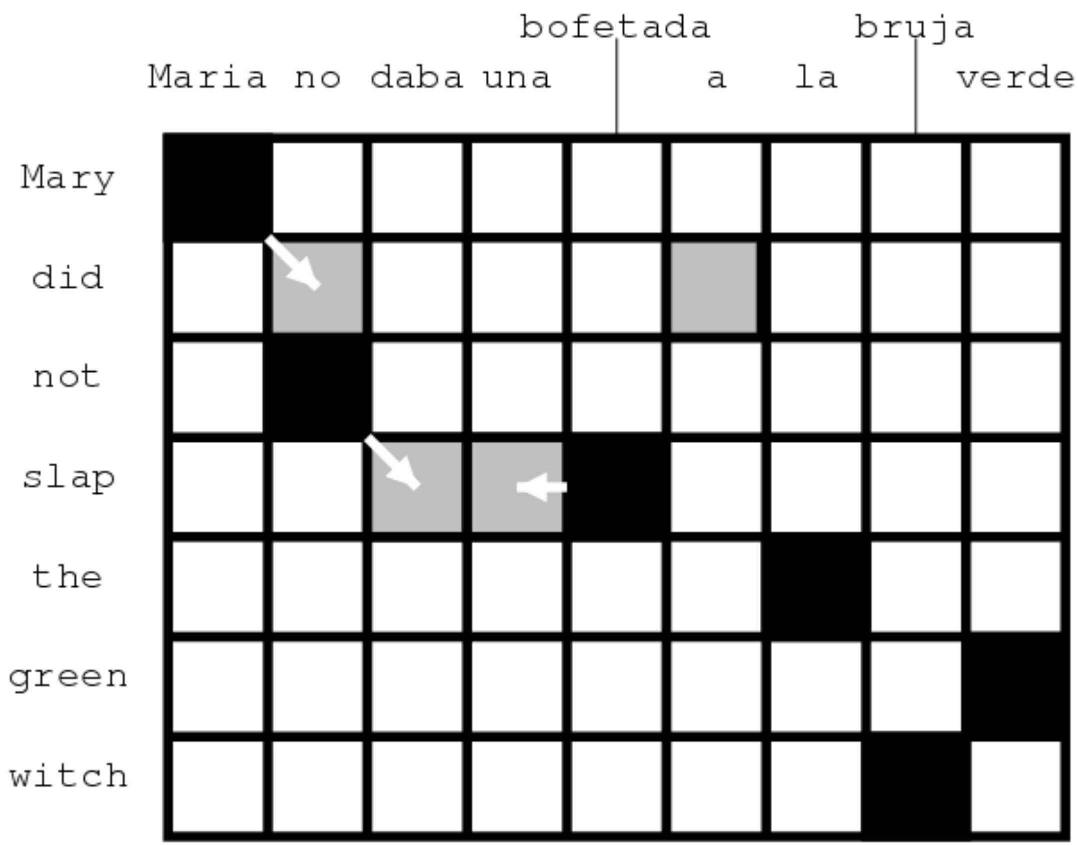


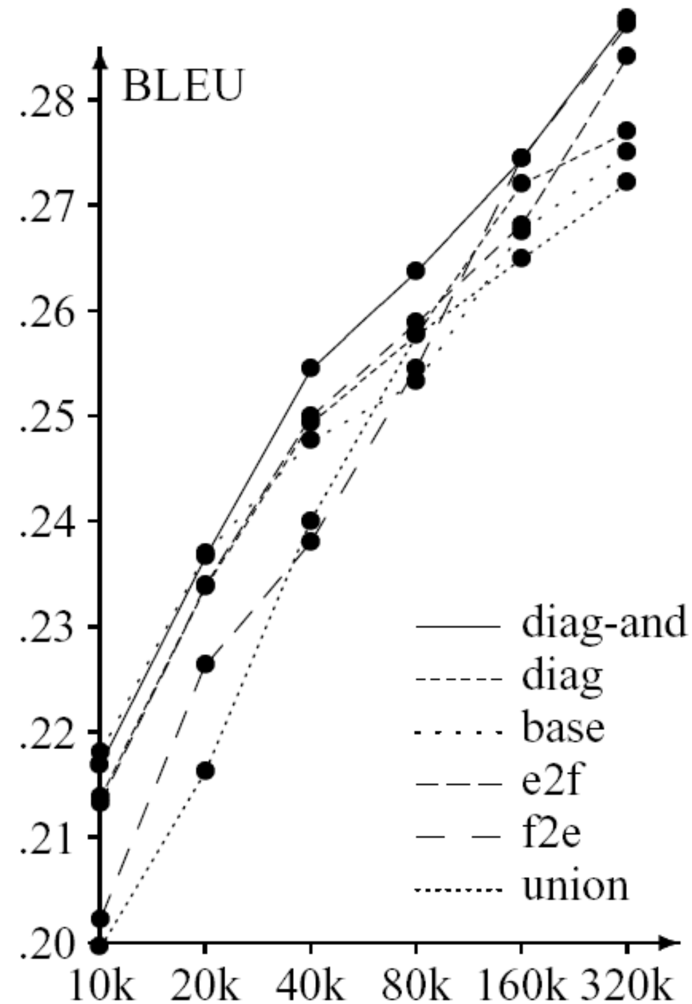


Bidirectional Alignment



Alignment Heuristics

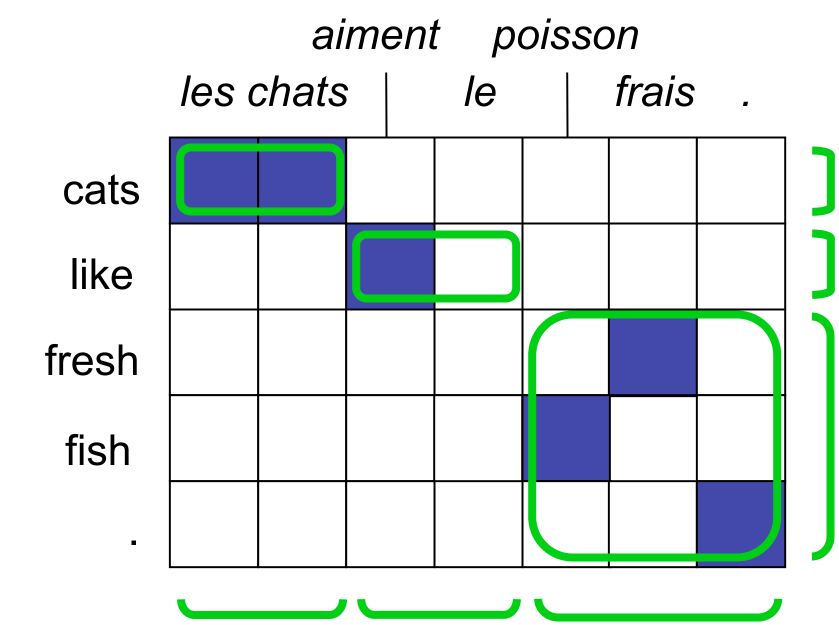




Phrase Scoring

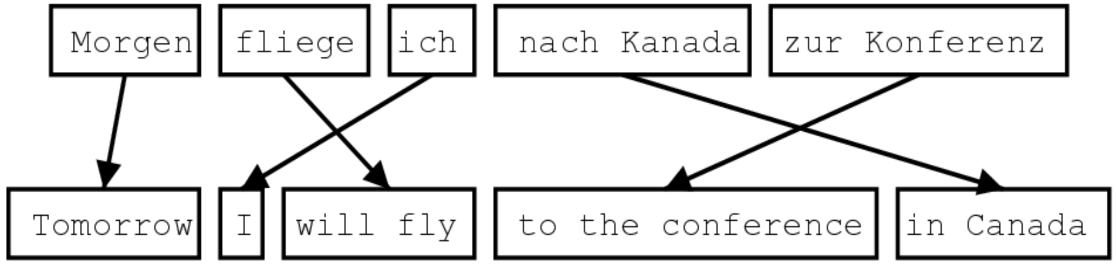
$$g(f, e) = \log \frac{c(e, f)}{c(e)}$$

$$g(\text{les chats}, \text{cats}) = \log \frac{c(\text{cats}, \text{les chats})}{c(\text{cats})}$$



- Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - … and others
- Seems not to work well, for a variety of partially understood reasons
- Main issue: big chunks get all the weight, obvious priors don't help
  - Though, [DeNero et al 08]

Scoring:



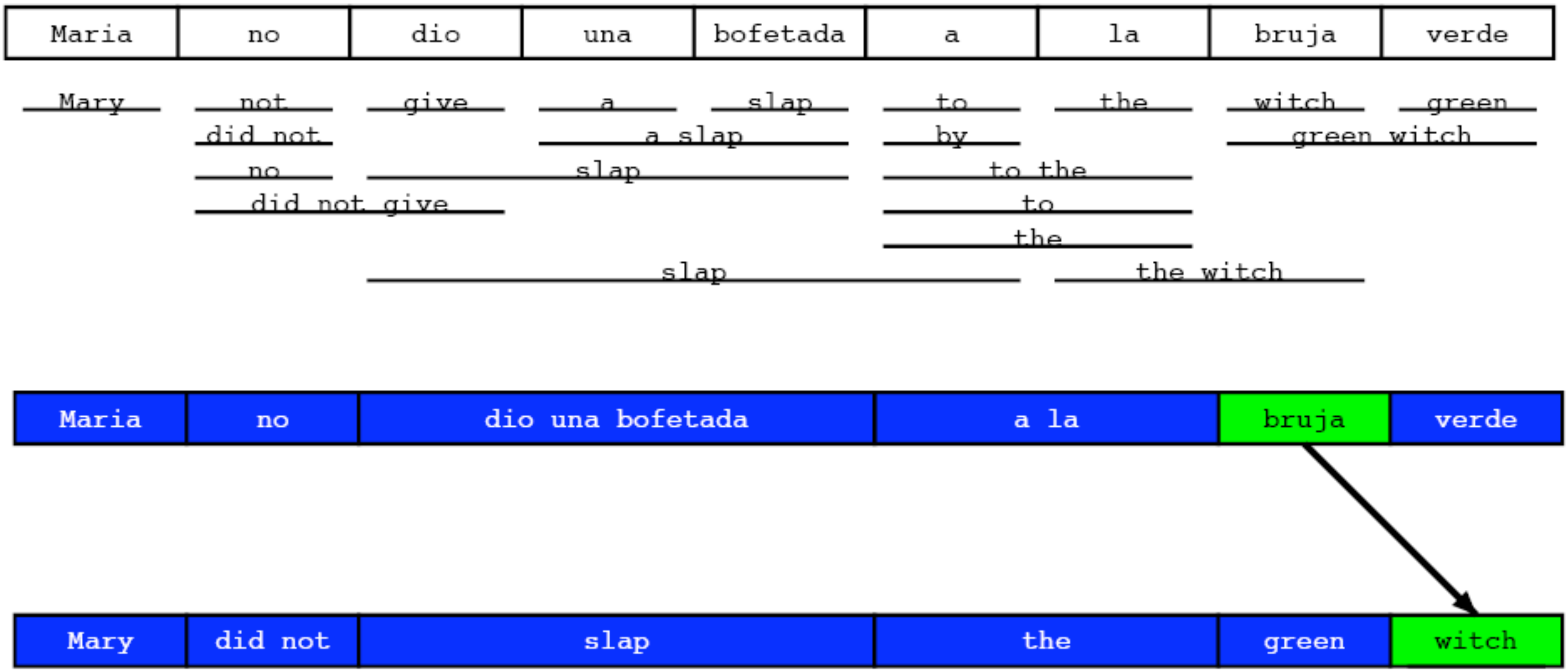
- Basic approach, sum up phrase translation scores and a language model
  - Define y = $p_1 p_2 \ldots p_L$ to be a translation with phrase pairs p_i
  - Define e(y) be the output English sentence in y
  - Let h() be the log probability under a tri-gram language model
  - Let g() be a phrase pair score (from last slide)
  - Then, the full translation score is:

$$f(y) = h(e(y)) + \sum_{k=1}^{L} g(p_k)$$

- Goal, compute the best translation

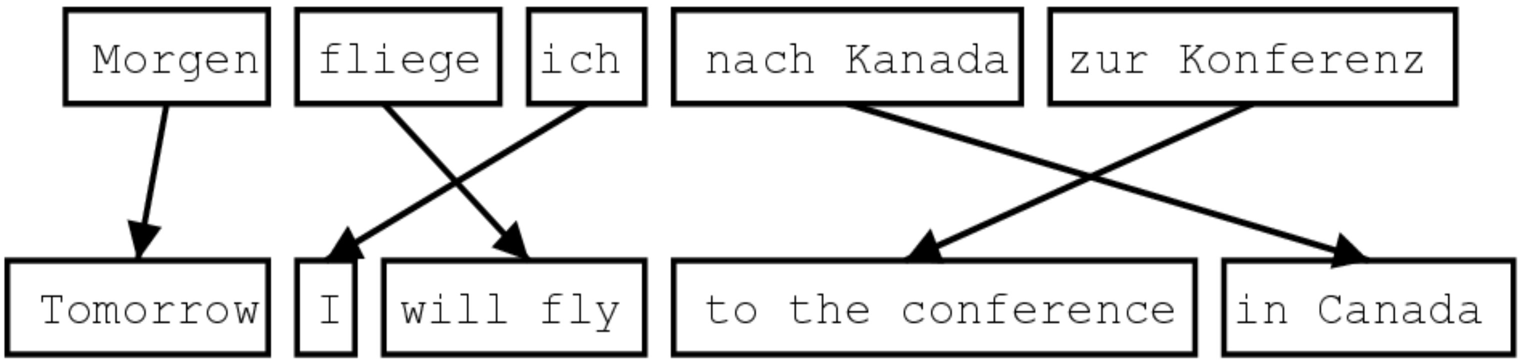
$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} f(y)$$

The Pharaoh Decoder



- Scores at each step include LM and TM

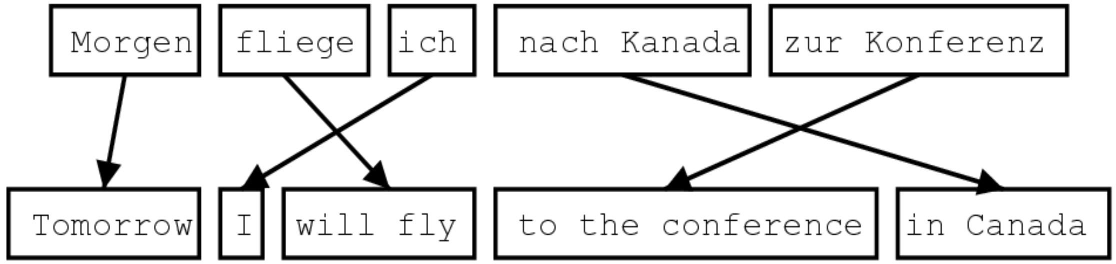
The Pharaoh Decoder



Space of possible translations

- Phrase table constrains possible translations
- Output sentence is built left to right
  - but source phrases can match any part of sentence
- Each source word can only be translated once
- Each source word must be translated

Scoring:



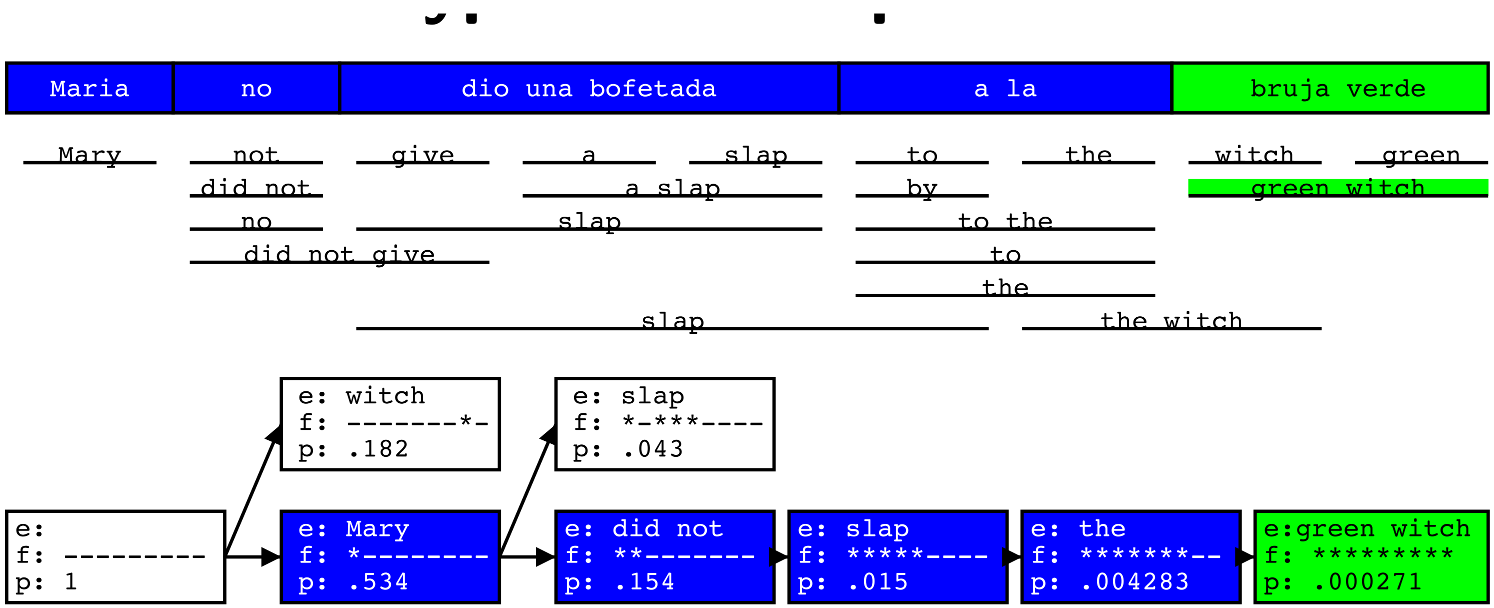
- In practice, much like for alignment models, also include a distortion penalty
  - Define $y = p_1 p_2 \ldots p_L$ to be a translation with phrase pairs p_i
  - Let $s(p_i)$ be the start position of the foreign phrase
  - Let $t(p_i)$ be the end position of the foreign phrase
  - Define η to be the distortion score (usually negative!)
  - Then, we can define a score *with distortion penalty*:

$$f(y) = h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})|$$

- Goal, compute the best translation

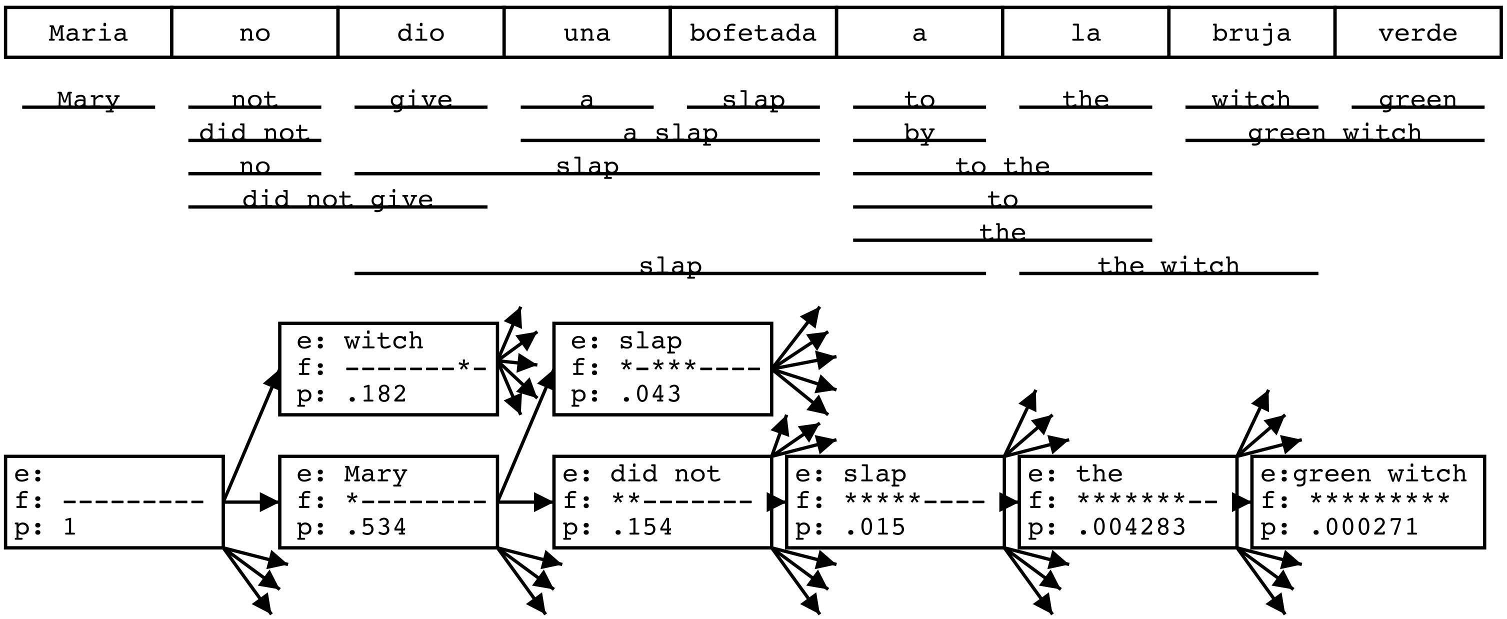
$$y^*(x) = \arg \max_{y \in \mathcal{Y}(x)} f(y)$$

Hypothesis Expansion



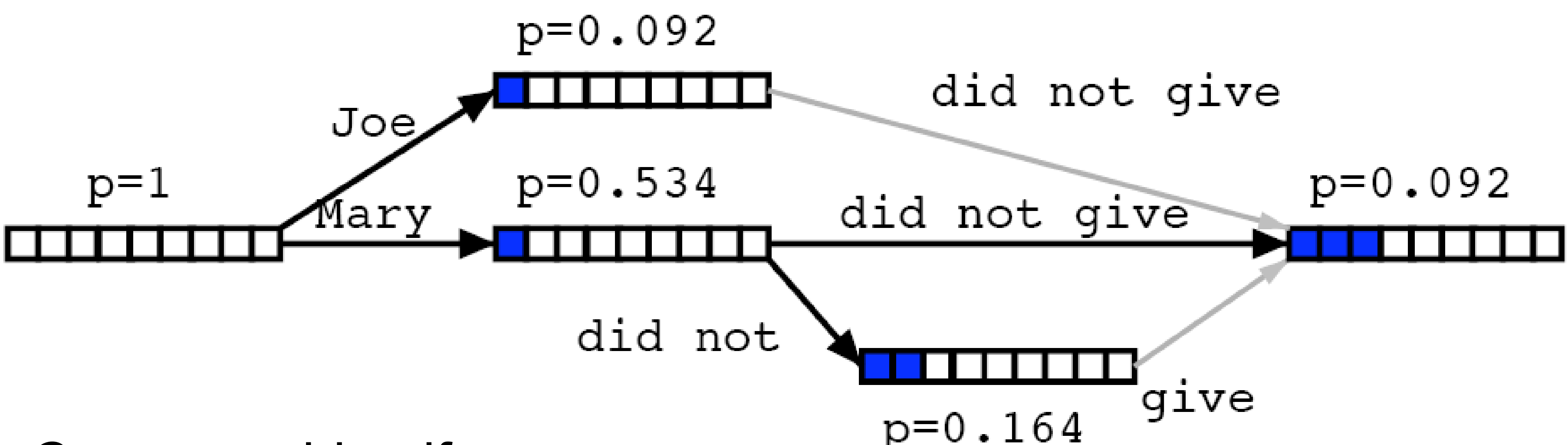
- ... until all foreign words *covered*
  - find *best hypothesis* that covers all foreign words
  - *backtrack* to read off translation

Hypothesis Explosion!



- Q: How much time to find the best translation?
  - Exponentially many translations, in length of source sentence
  - NP-hard, just like for word translation models
  - So, we will use approximate search techniques!

Hypothesis Lattices



Can recombine if:

- Last two English words match
- Foreign word coverage vectors match

Decoder Pseudocode

Initialization: Set beam $Q=\{q_0\}$ where q_0 is initial state with no words translated

For i=0 … n-1 [where n in input sentence length]

- For each state $q \in beam(Q)$ and phrase $p \in ph(q)$
  1. q'=next(q,p) [compute the new state]
  2. Add(Q,q',q,p) [add the new state to the beam]

Notes:

- ph(q): set of phrases that can be added to partial translation in state q
- next(q,p): updates the translation in q and records which words have been translated from input
- Add(Q,q',q,p): updates beam, q' is added to Q if it is in the top-n overall highest scoring partial translations

Decoder Pseudocode

Initialization: Set beam $Q=\{q_0\}$ where q_0 is initial state with no words translated

For i=0 … n-1 [where n in input sentence length]

- For each state $q \in beam(Q)$ and phrase $p \in ph(q)$
  1. q'=next(q,p) [compute the new state]
  2. Add(Q,q',q,p) [add the new state to the beam]

Possible State Representations:

- Full: $q = (e, b, \alpha)$, e.g. ("Joe did not give," 11000000, 0.092)
  - e is the partial English sentence
  - b is a bit vector recorded which source words are translated
  - α is score of translation so far

Decoder Pseudocode

Initialization: Set beam $Q=\{q_0\}$ where q_0 is initial state with no words translated

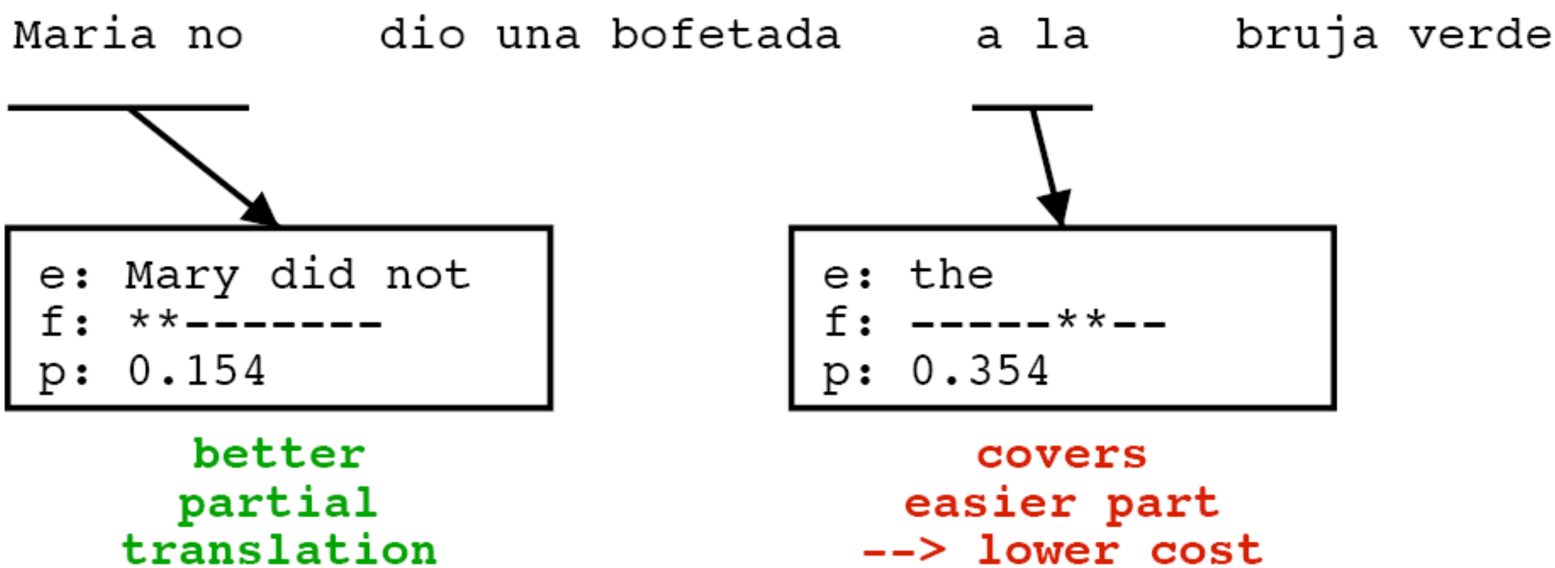
For i=0 … n-1 [where n in input sentence length]

- For each state $q \in beam(Q)$ and phrase $p \in ph(q)$
  1. q'=next(q,p) [compute the new state]
  2. Add(Q,q',q,p) [add the new state to the beam]

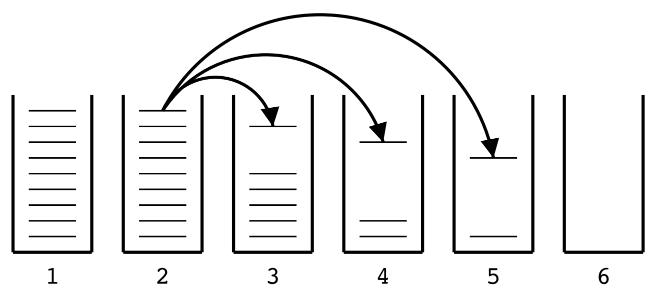
Possible State Representations:

- Full: q = (e, b, α), e.g. ("Joe did not give," 11000000, 0.092)
- Compact: $q = (e_1, e_2, b, r, \alpha)$,
  - e.g. ("not," "give," 11000000, 4, 0.092)
  - e_1 and e_2 are the last two words of partial translation
  - r is the length of the partial translation
- Compact representation is more efficient, but requires back pointers to get the final translation

Pruning



- Problem: easy partial analyses are cheaper
  - Solution 1: separate bean for each number of foreign words
  - Solution 2: estimate forward costs (A*-like)



Decoder Pseudocode (Multibeam)

Initialization:

- set $Q_0=\{q_0\}$, $Q_i=\{\}$ for I = 1 … n [n is input sent length]

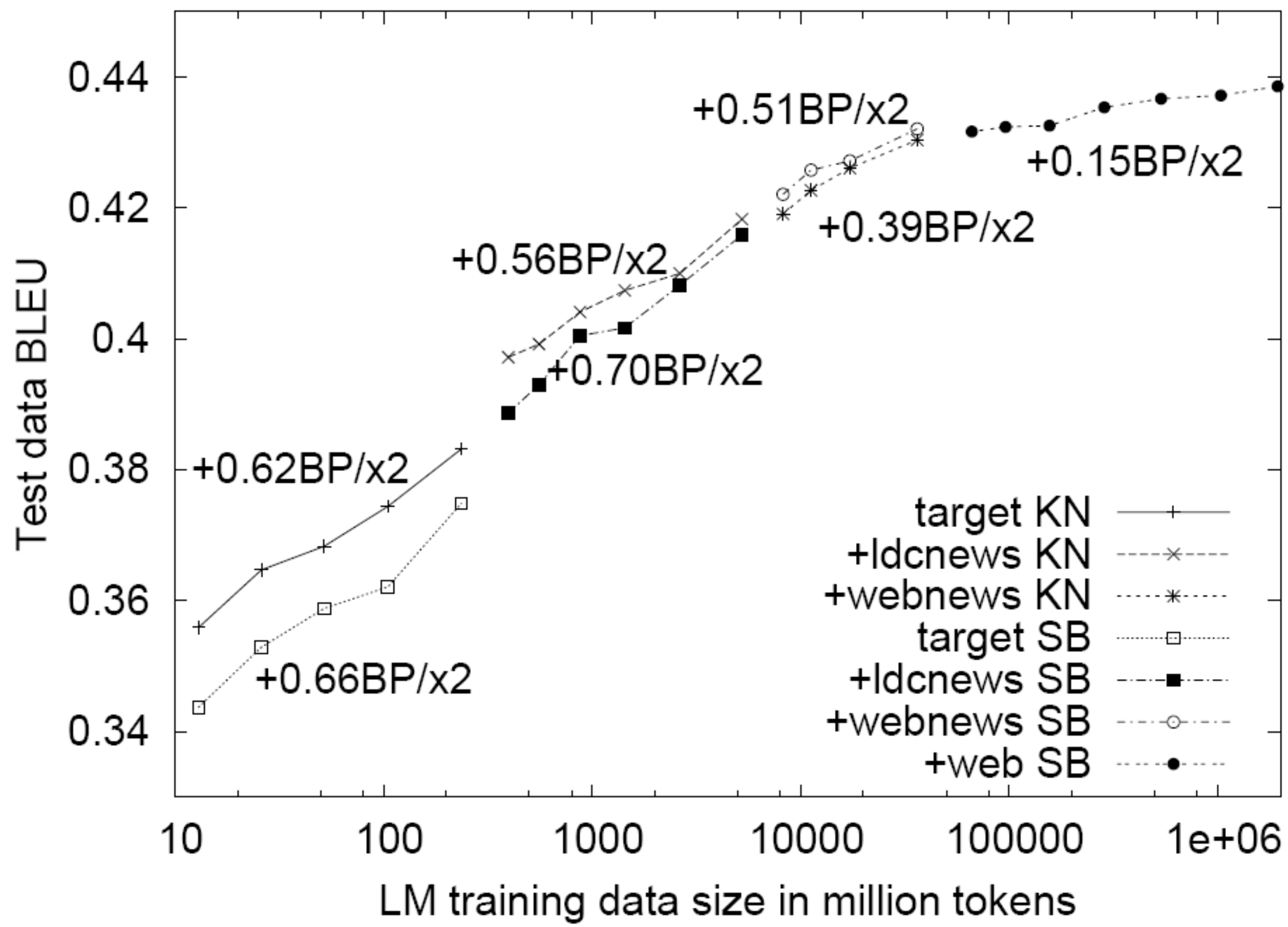
For i=0 … n-1

- For each state $q \in beam(Q_i)$ and phrase $p \in ph(q)$
  1. q'=next(q,p)
  2. Add(Q_j,q',q,p) where j = len(q')

Notes:

- Q_i is a beam of all partial translations where i input words have been translated
- len(q) is the number of bits equal to one in q (the number of words that have been translated)

Tons of Data?



- Discussed for LMs, but can new understand full model!

Tuning for MT

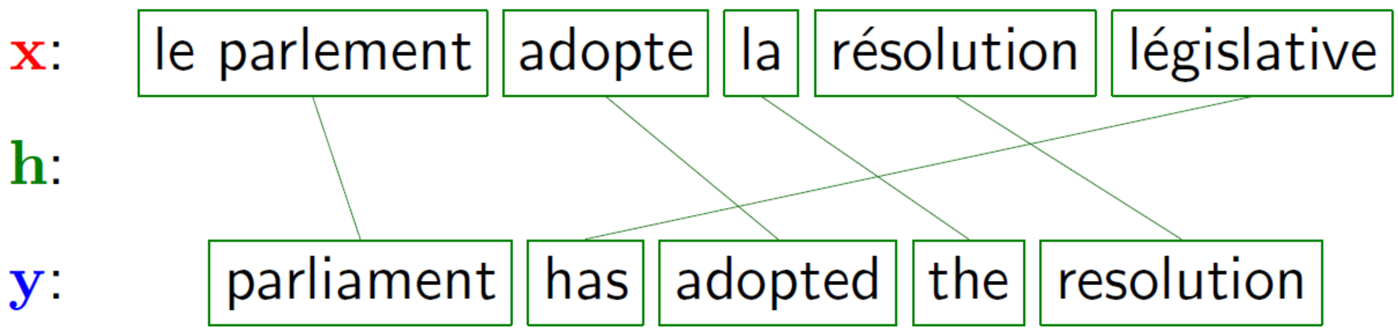
- Features encapsulate lots of information
  - Basic MT systems have around 6 features
  - P(e|f), P(f|e), lexical weighting, language model

- How to tune feature weights?

- Idea 1: Use your favorite classifier

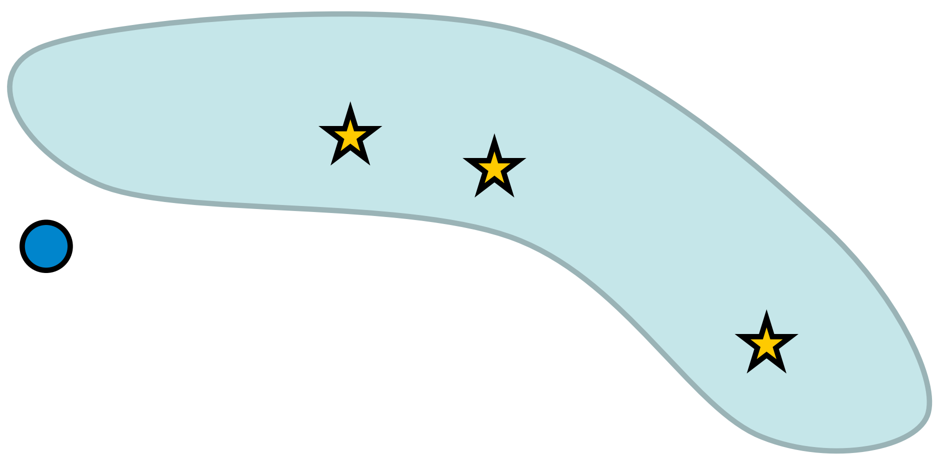
Why Tuning is Hard

- Problem 1: There are latent variables
  - Alignments and segementations
  - Possibility: forced decoding (but it can go badly)



Why Tuning is Hard

- Problem 2: There are many right answers
  - The reference or references are just a few options
  - No good characterization of the whole class



  - BLEU isn't perfect, but even if you trust it, it's a corpus-level metric, not sentence-level

Linear Models: Perceptron

- The perceptron algorithm
  - Iteratively processes the training set, reacting to training errors
  - Can be thought of as trying to drive down training error
- The (online) perceptron algorithm:
  - Start with zero weights
  - Visit training instances (x_i,y_i) one by one
    - Make a prediction

$$y^* = \arg\max_y w \cdot \phi(x_i, y)$$

    - If correct ($y^*==y_i$): no change, goto next example!
    - If wrong: adjust weights

$$w = w + \phi(x_i, y_i) - \phi(x_i, y^*)$$

Perceptron training

For each training example $(\mathbf{x}, \mathbf{y})$: [Collins '02]

$$\mathbf{w} \leftarrow \mathbf{w} \; +\Phi(\mathbf{x}, \mathbf{y}_{\mathrm{t}}) \; -\Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}) \qquad \begin{array}{|lcl} \mathbf{y}_{\mathrm{t}} & = & \mathbf{y} \\ \mathbf{y}_{\mathrm{p}} & = & \textsc{Decode}(\mathbf{x}) \end{array}$$

$$\mathbf{w} \leftarrow \mathbf{w} \; +\Phi(\mathbf{x}, \mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}) \; -\Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}}) \qquad \begin{array}{|lcl} \mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}} & = & ??? \\ \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}} & = & \textsc{Decode}(\mathbf{x}) \end{array}$$

Update strategies

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \boxed{\mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}}) - \Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$$

Training example (reference)

$\mathbf{x}$: voté sur demande d ' urgence

$\mathbf{y}$: vote on a request for urgent procedure

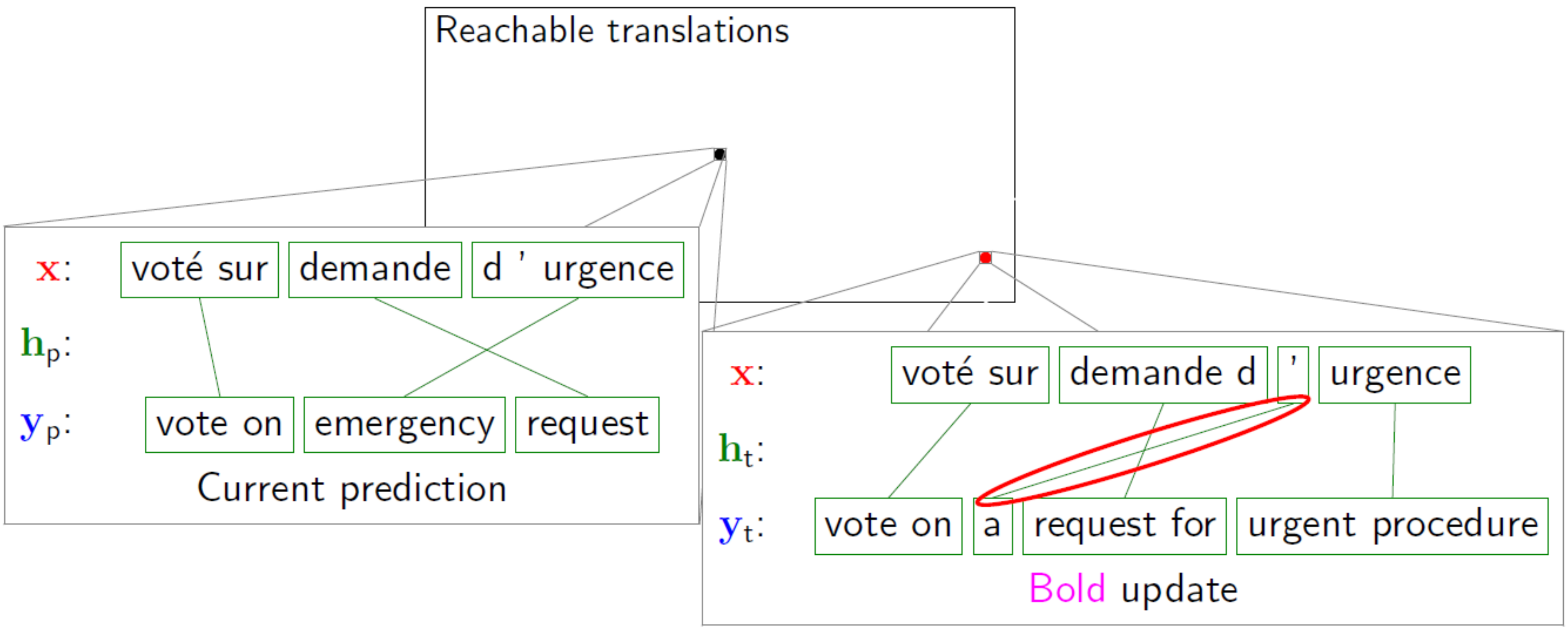
Update strategies

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \boxed{\mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}}\,) - \Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$$

Training example (reference)

$\mathbf{x}$: voté sur demande d ' urgence

$\mathbf{y}$: vote on a request for urgent procedure



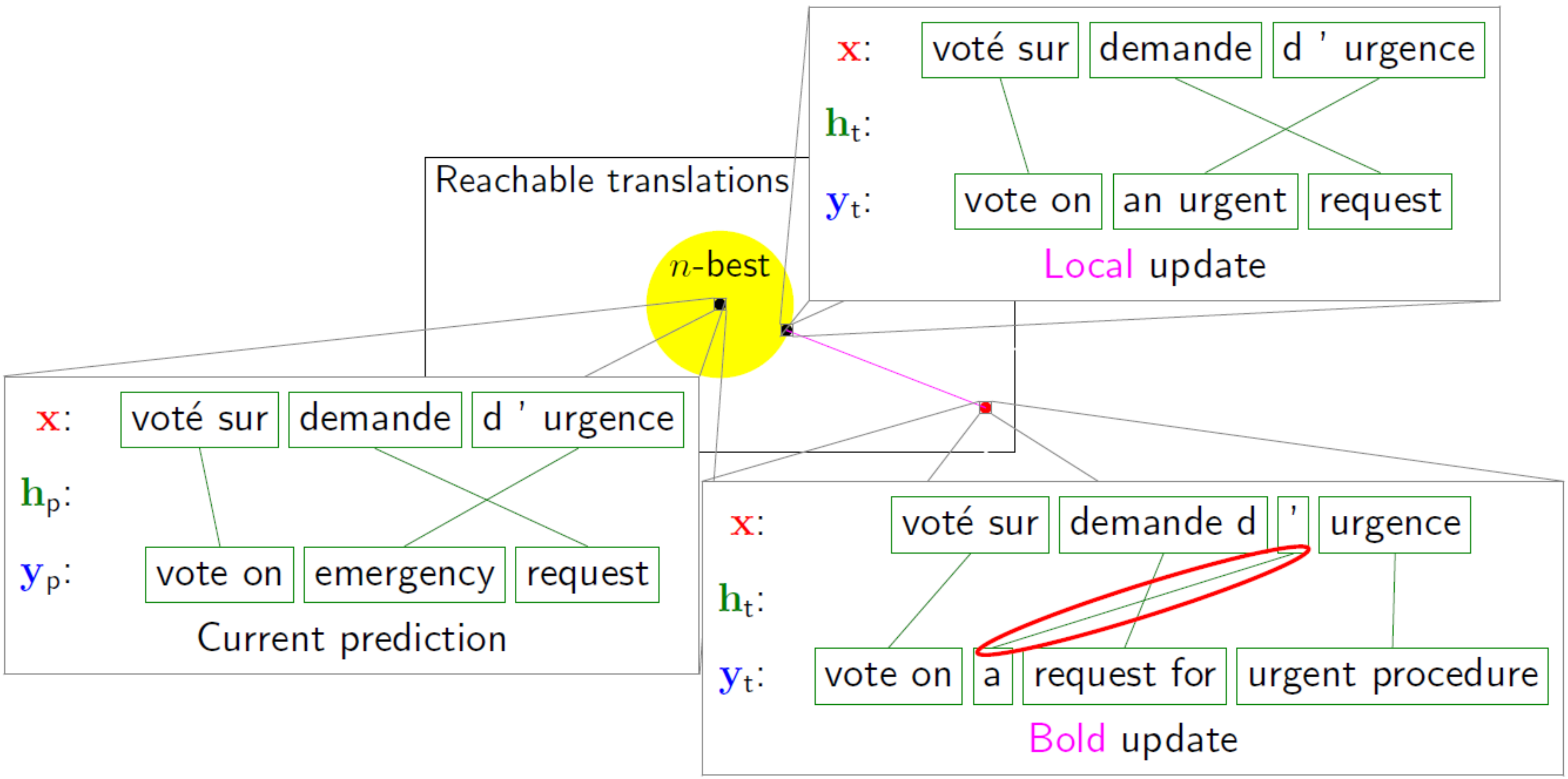
Update strategies

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \boxed{\mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}}) - \Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$$

Training example (reference)

$\mathbf{x}$: voté sur demande d ' urgence

$\mathbf{y}$: vote on a request for urgent procedure



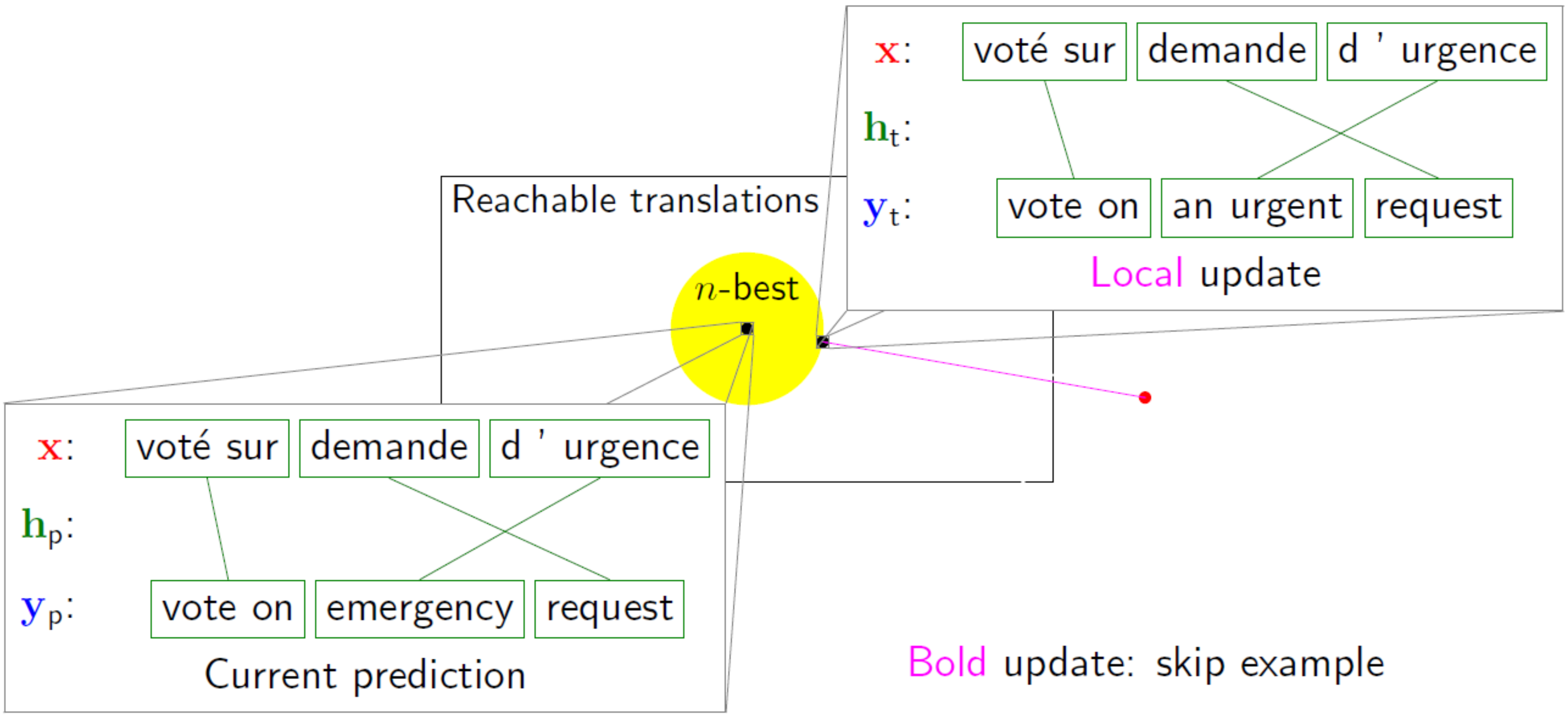
Update strategies

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \boxed{\mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}}\,) - \Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$$

Training example (reference)

$\mathbf{x}$: voté sur demande d ' urgence

$\mathbf{y}$: vote on a request for urgent procedure



Bold update: skip example

Update strategies

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}, \boxed{\mathbf{y}_{\mathrm{t}}, \mathbf{h}_{\mathrm{t}}}\,) - \Phi(\mathbf{x}, \mathbf{y}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$$

Training example (reference)

$\mathbf{x}$: voté sur demande d ' urgence

$\mathbf{y}$: vote on a request for urgent procedure

$\mathbf{x}$:

$\mathbf{h}_{\mathrm{t}}$:

te

Decoder	Bold	Local
Monotonic	34.3	**34.6**
Limited distortion	33.5	**34.7**

$\mathbf{x}$:

$\mathbf{h}_{\mathrm{p}}$:

$\mathbf{y}_{\mathrm{p}}$:

Current prediction

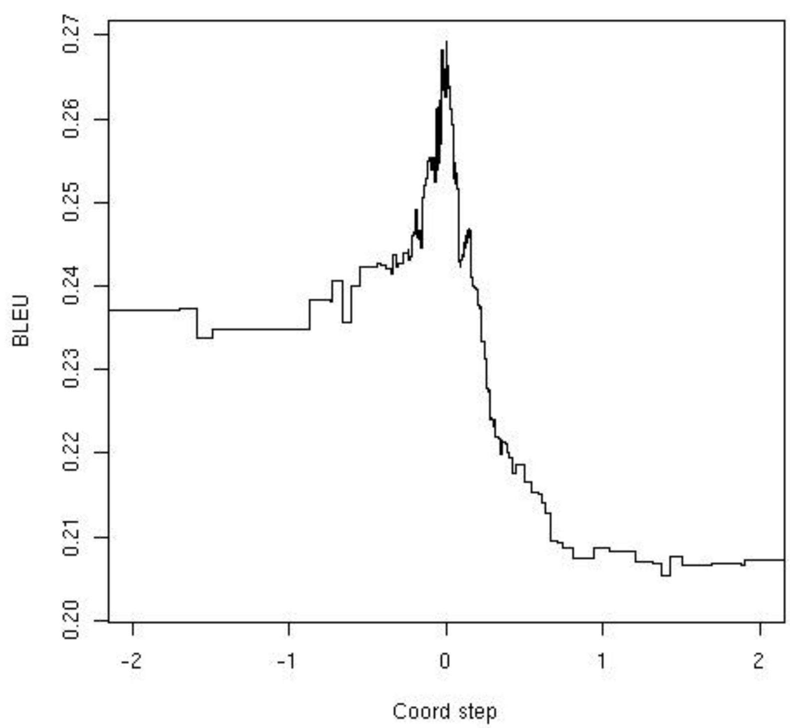
Bold update: skip example

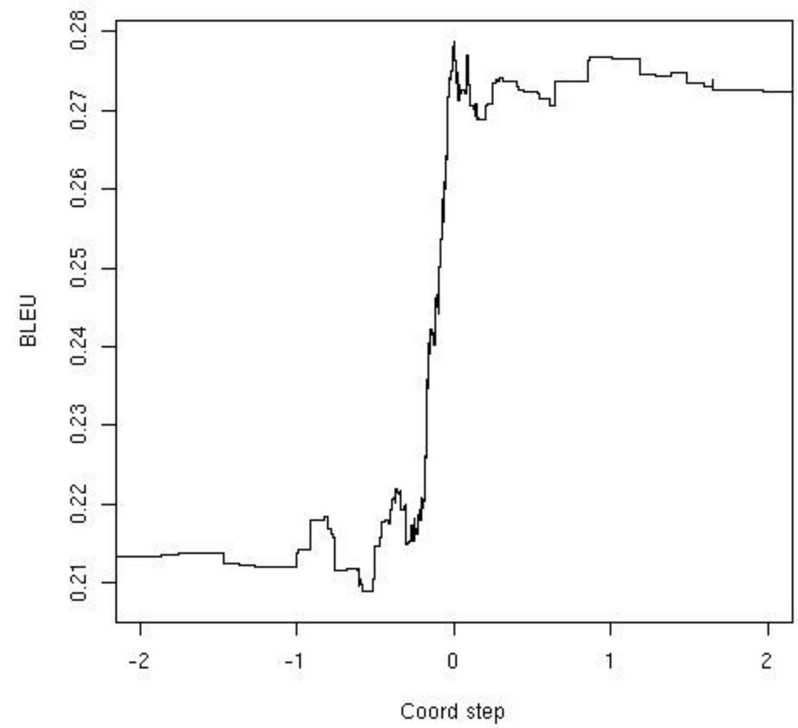
Why Tuning is Hard

- Problem 3: Computational constraints
  - Discriminative training involves repeated decoding
  - Very slow! So people tune on sets much smaller than those used to build phrase tables

Minimum Error Rate Training

- Standard method: minimize BLEU directly (Och 03)
  - MERT is a discontinuous objective
  - Only works for max ~10 features, but works very well then
  - Here: k-best lists, but forest methods exist (Machery et al 08)





MERT: Convex Upper Bound of BLEU