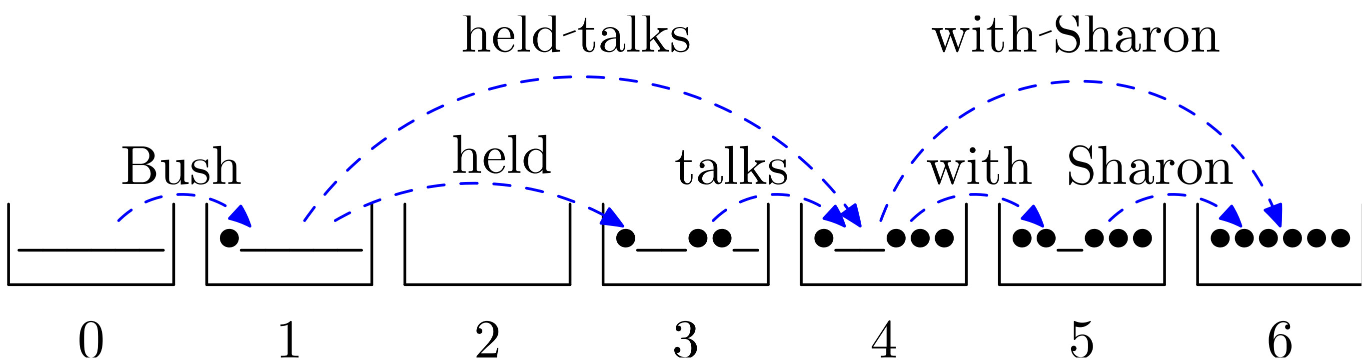
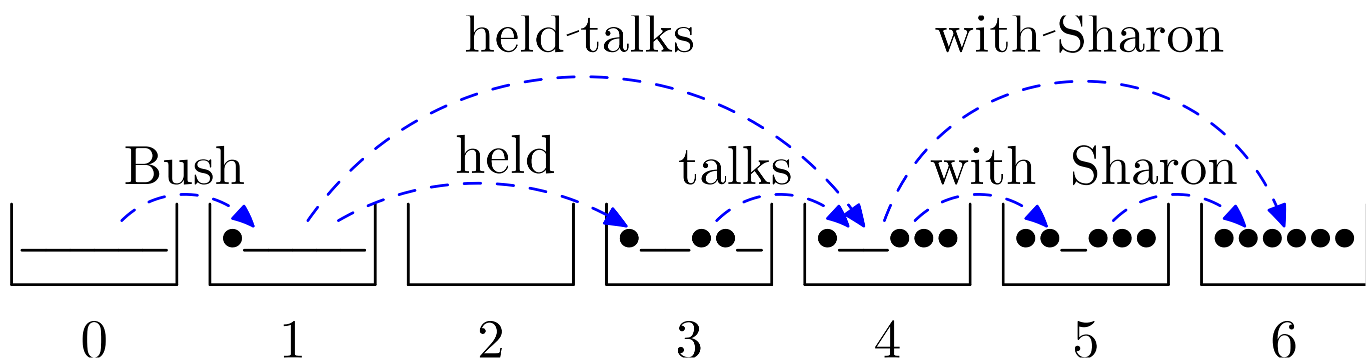
MaxForce: Max-Violation Perceptron and Forced Decoding for Scalable MT Training



Heng Yu — Chinese Acad. of Sciences
Liang Huang — CUNY
Haitao Mi — IBM T. J. Watson
Kai Zhao — CUNY

Scalable Training for MT Finally Made Successful

MaxForce: Max-Violation Perceptron and Forced Decoding for Scalable MT Training



Heng Yu — Chinese Acad. of Sciences

Liang Huang — CUNY

Haitao Mi — IBM T. J. Watson

Kai Zhao — CUNY

Discriminative Training for SMT

- discriminative training is dominant in parsing / tagging
  - can use arbitrary, overlapping, lexicalized features
  - but not very successful yet in machine translation
- most efforts on MT training tune feature weights on the small dev set (~1k sents) not the training set!
  - as a result can only use ~10 dense features (MERT)
  - or ~10k rather impoverished features (MIRA/PRO)
- Liang et al (2006) train on the training set but failed

training set (>100k sentences)	dev set (~1k sents)	test set (~1k sents)

Timeline for MT Training

MERT
(Och ’02)

(dense features)

training set (>100k sentences)

dev set
(~1k sents)

test set
(~1k sents)

Timeline for MT Training

MERT (Och '02)

(dense features)

Standard Perceptron (a noble failure)
(Liang et al 2006)

training set (>100k sentences)

dev set (~1k sents)

test set (~1k sents)

Timeline for MT Training

MERT
(Och ’02)

(dense features)

Standard Perceptron (a noble failure)
(Liang et al 2006)

MIRA
(Watanabe+ ’07)
(Chiang+ ’08-’12)

(pseudo sparse
features)

training set (>100k sentences)

dev set
(~1k sents)

test set
(~1k sents)

Timeline for MT Training

MERT
(Och ’02)

(dense features)

Standard Perceptron (a noble failure)
(Liang et al 2006)

MIRA
(Watanabe+ ’07)
(Chiang+ ’08-’12)

(pseudo sparse
features)

PRO
(Hopkins+May ’11)

Regression
(Bazrafshan+ ’12)

training set (>100k sentences)

dev set
(~1k sents)

test set
(~1k sents)

Timeline for MT Training

Standard Perceptron (a noble failure)
(Liang et al 2006)

MERT
(Och '02)

(dense features)

MIRA
(Watanabe+ '07)
(Chiang+ '08-'12)

(pseudo sparse features)

PRO
(Hopkins+May '11)

Regression
(Bazrafshan+ '12)

HOLS
(Flanigan+ '13)

(sparse features as one dense feature)

training set (>100k sentences)

dev set
(~1k sents)

test set
(~1k sents)

Timeline for MT Training

Standard Perceptron (a noble failure)
(Liang et al 2006)

our work (2013): violation-fixing
perceptron with truly sparse features

MERT
(Och '02)

(dense features)

MIRA
(Watanabe+ '07)
(Chiang+ '08-'12)

(pseudo sparse
features)

PRO
(Hopkins+May '11)

Regression
(Bazrafshan+ '12)

HOLS
(Flanigan+ '13)

(sparse features as
one dense feature)

training set (>100k sentences)

dev set
(~1k sents)

test set
(~1k sents)

Timeline for MT Training

Standard Perceptron (a noble failure)
(Liang et al 2006)

?

our work (2013): violation-fixing perceptron with truly sparse features

training set (>100k sentences)

MERT
(Och ’02)

(dense features)

MIRA
(Watanabe+ ’07)
(Chiang+ ’08-’12)

(pseudo sparse features)

PRO
(Hopkins+May ’11)

Regression
(Bazrafshan+ ’12)

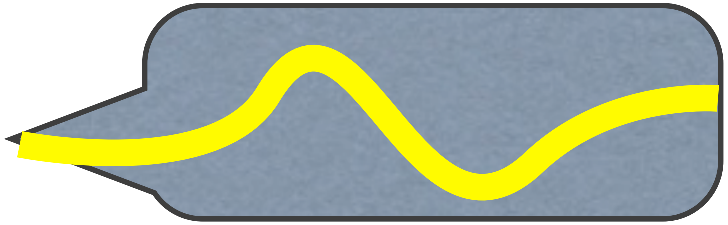
HOLS
(Flanigan+ ’13)

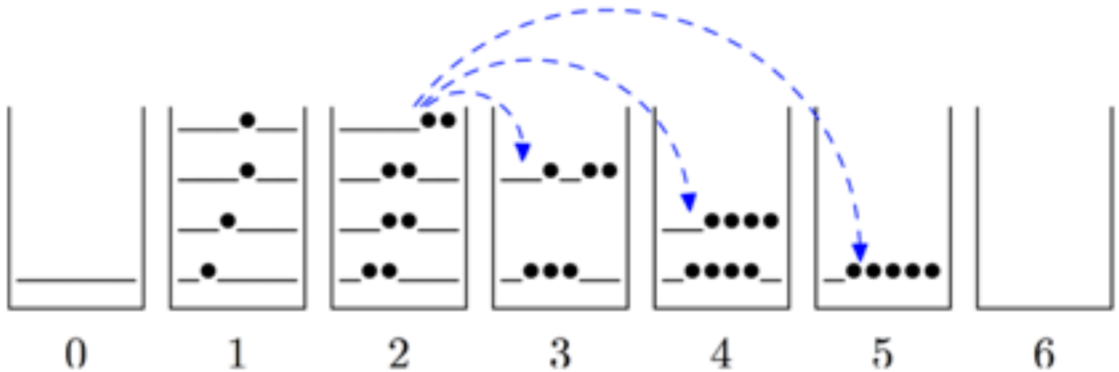
(sparse features as one dense feature)

dev set
(~1k sents)

test set
(~1k sents)

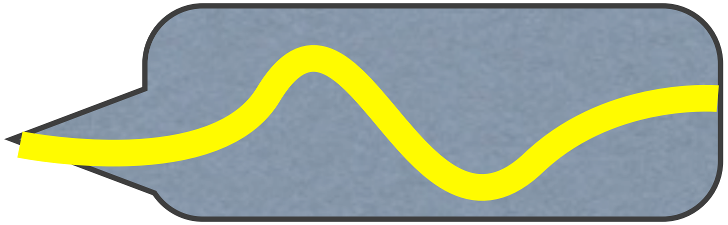
Why previous work fails

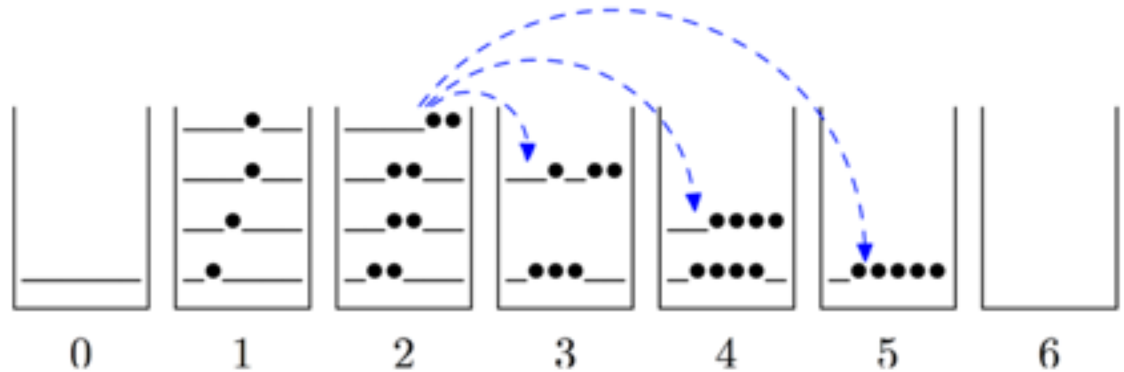




- their learning methods are based on exact search
  - MT has huge search spaces => severe search errors
    - learning algorithms should fix search errors
    - full updates (perceptron/MIRA/PRO) can't fix search errors
  - MT involves latent variables (derivations not annotated)
    - perceptron/MIRA was not designed for latent variables
- we need better variants for perceptron

Why our approach works

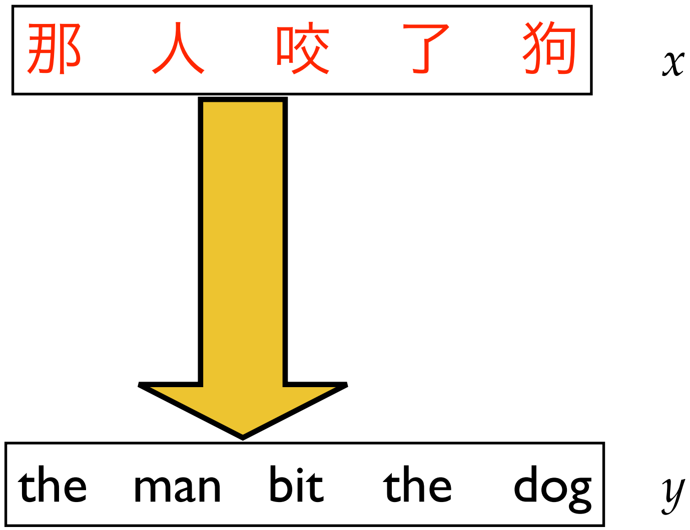




- use a variant of perceptron tailored for inexact search
  - fix search errors in the middle of the search
  - "partial updates" instead of "full updates"
- use forced decoding lattice as the target to update to
- use parallelized minibatch to speed up learning
- result: scaled to a large portion of the training data
  - 20M sparse features => +2.0 BLEU over MERT/PRO

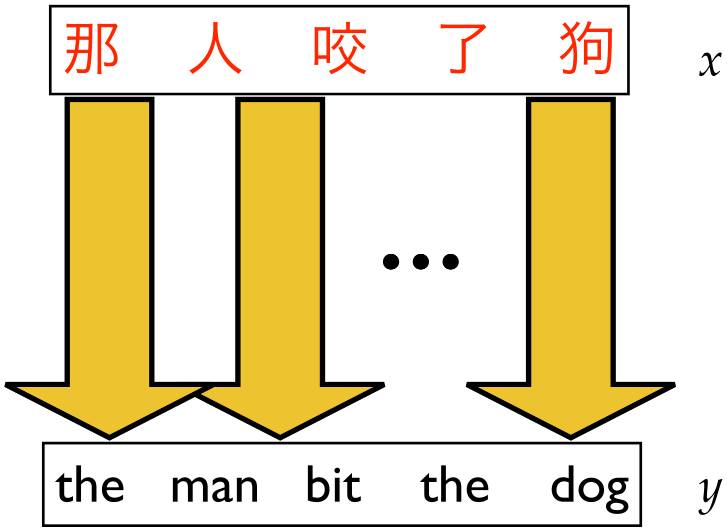
MT as Structured Classification

- with latent variables (hidden derivations)



MT as Structured Classification

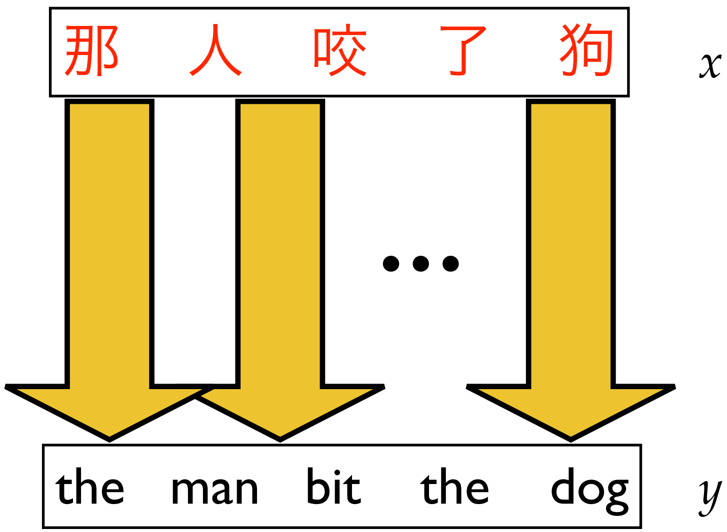
- with latent variables (hidden derivations)



all gold derivations

MT as Structured Classification

- with latent variables (hidden derivations)

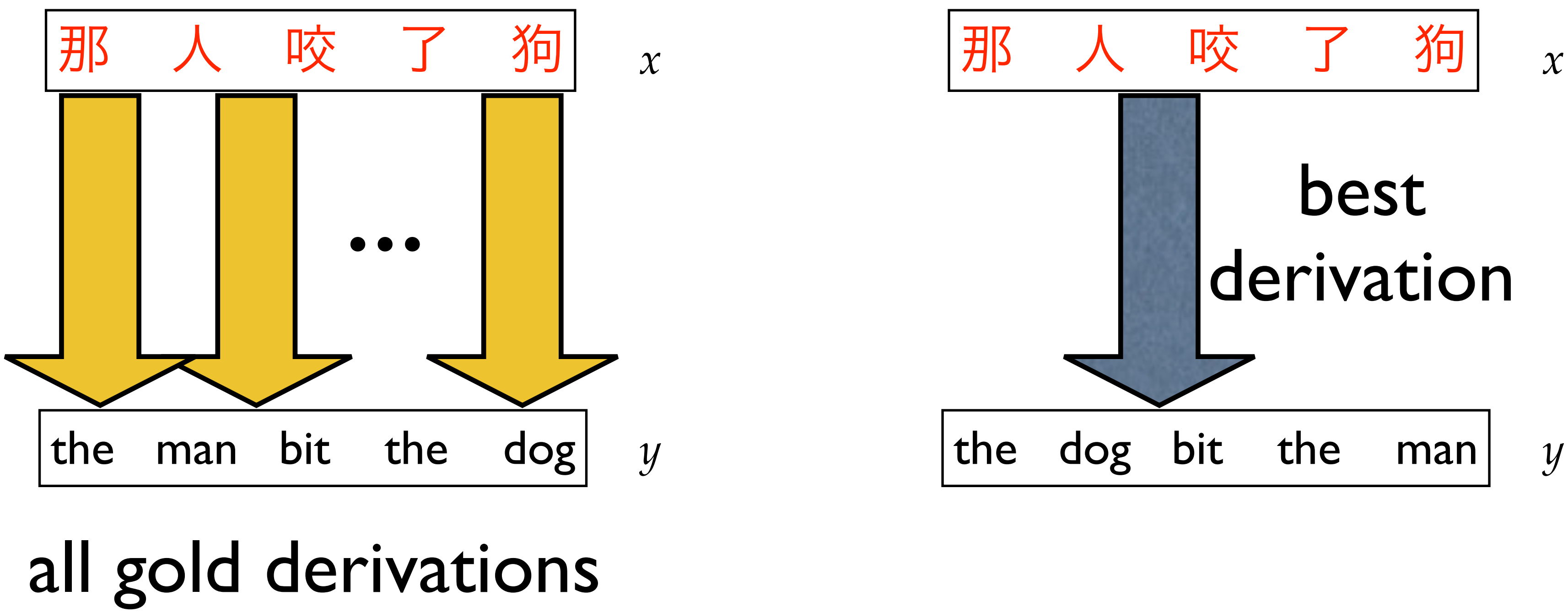


all gold derivations

那 人 咬 了 狗 x

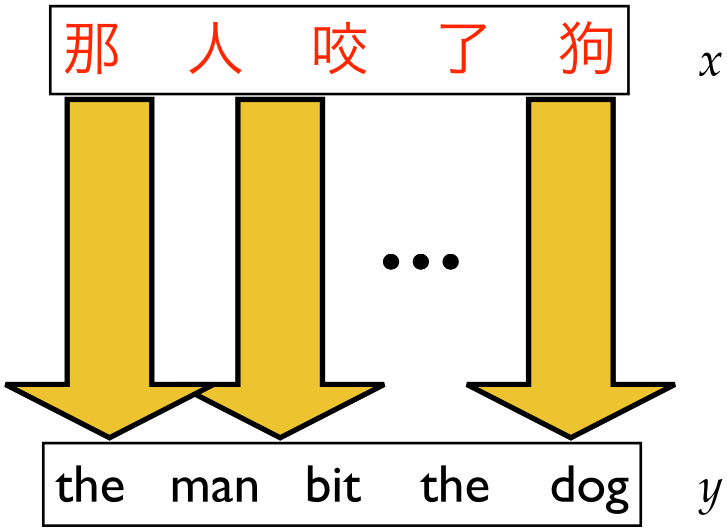
MT as Structured Classification

- with latent variables (hidden derivations)

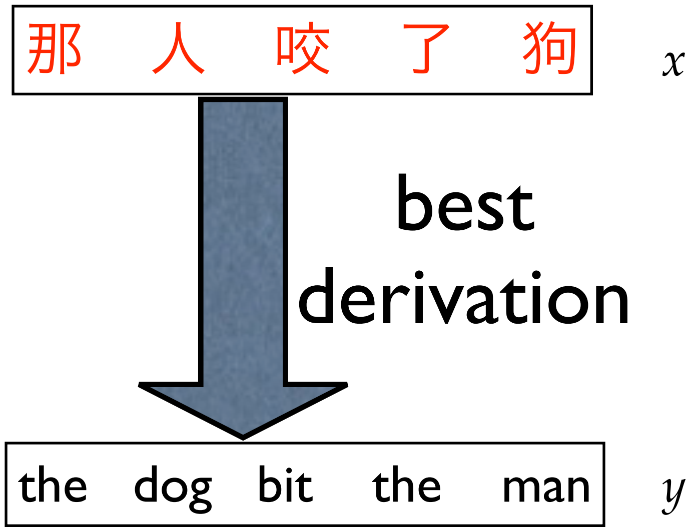


MT as Structured Classification

- with latent variables (hidden derivations)



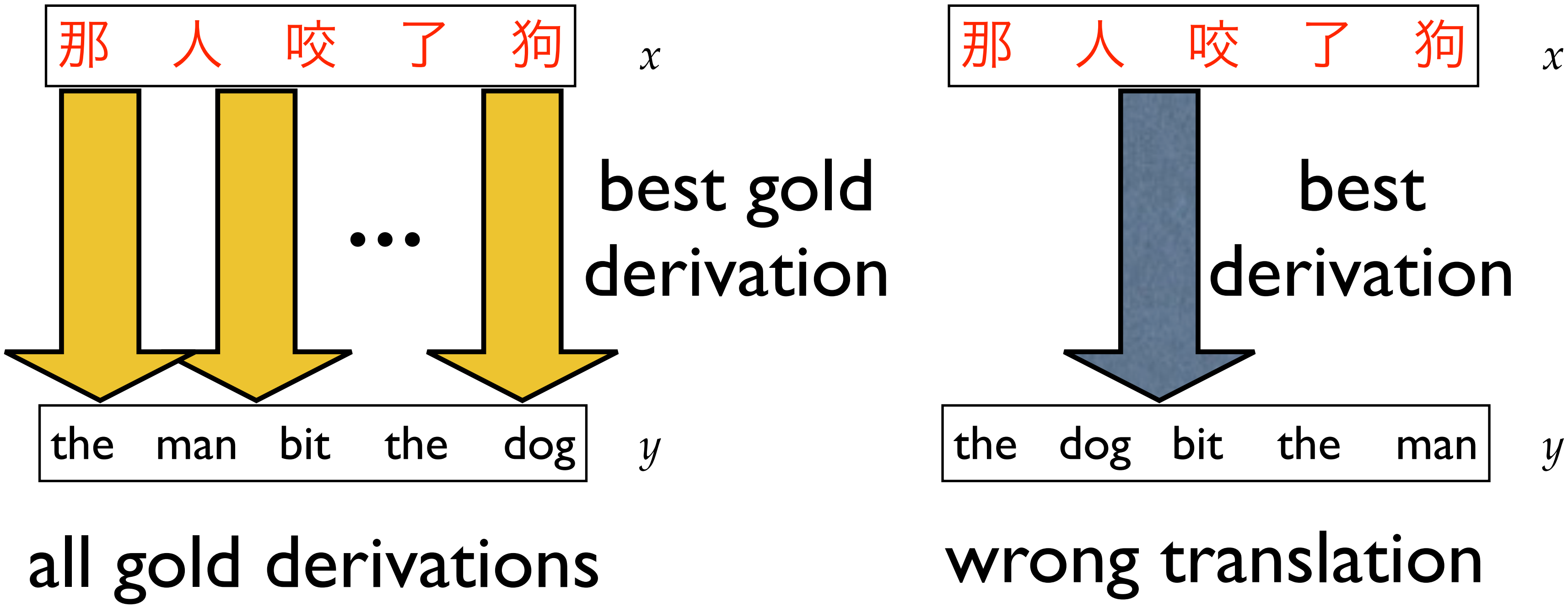
all gold derivations



wrong translation

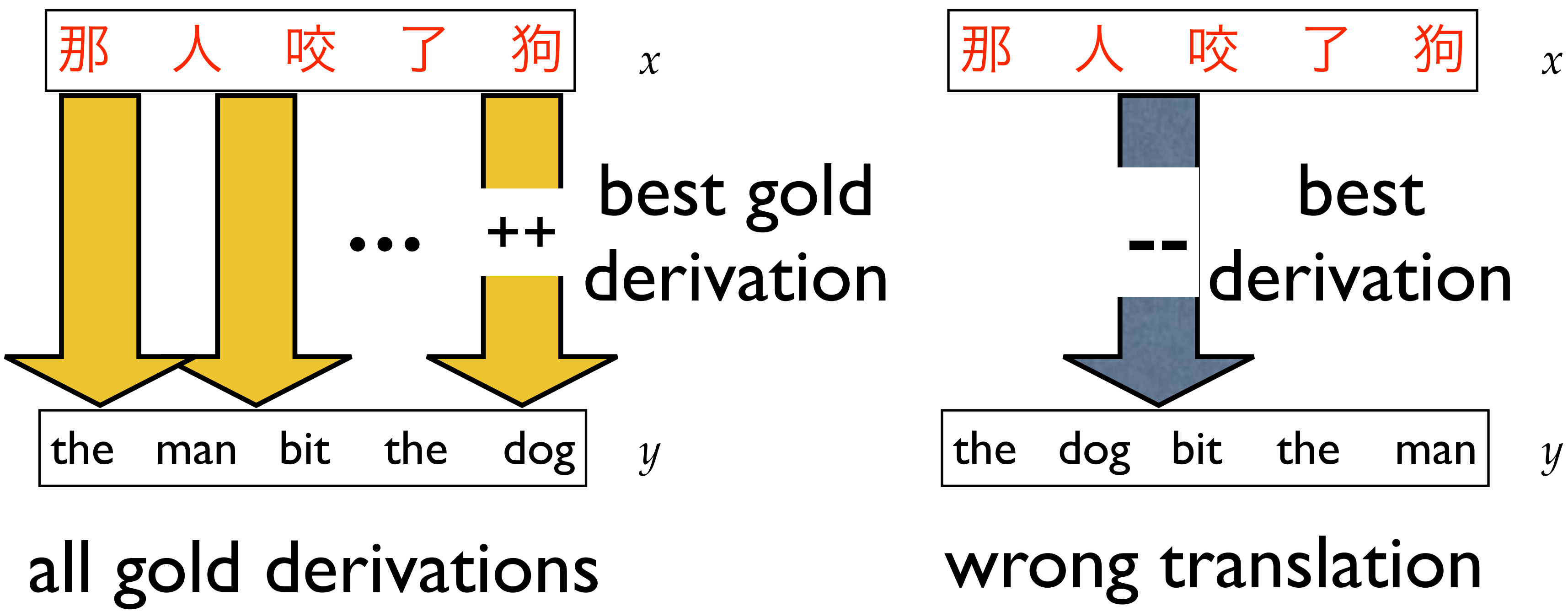
MT as Structured Classification

- with latent variables (hidden derivations)



MT as Structured Classification

- with latent variables (hidden derivations)



update: penalize best derivation
and reward best gold derivation

Outline

- Motivations
- Phrase-based Translation and Forced Decoding
- Violation-Fixing Perceptron for SMT
  - Update Strategies: Early Update and Max-Violation
  - Feature Design
- Experiments

Phrase-based translation

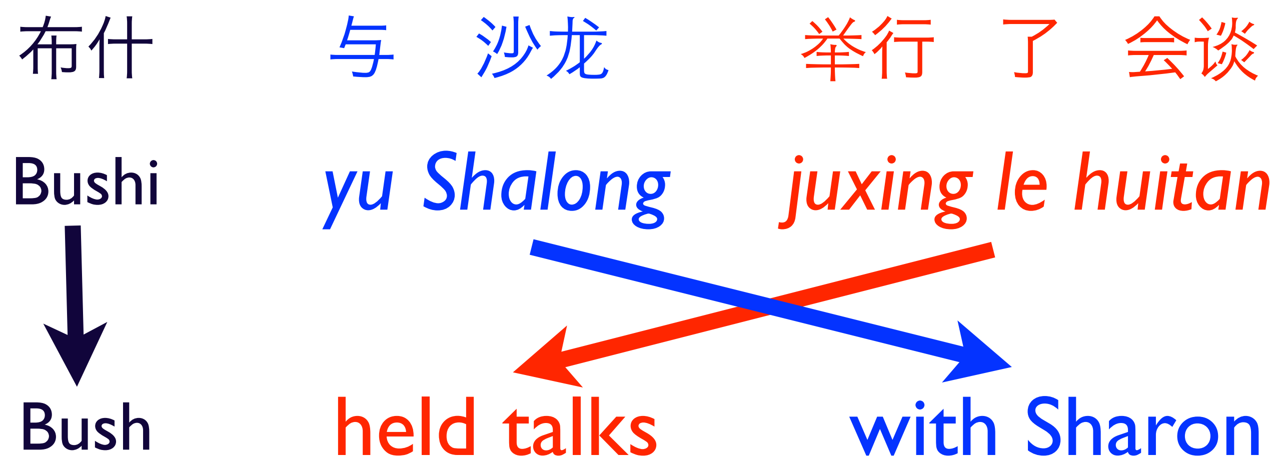
布什　与　沙龙　举行　了　会谈

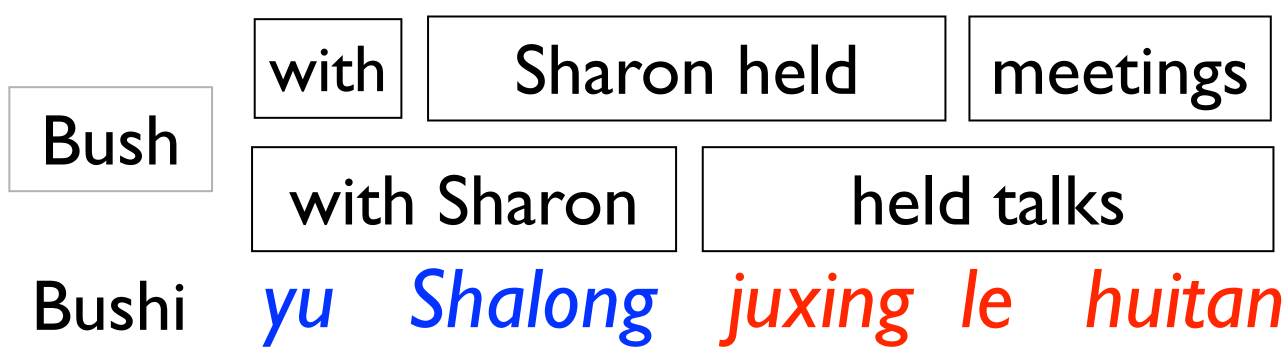
Bushi　*yu Shalong*　*juxing le huitan*

Bush　held talks　with Sharon

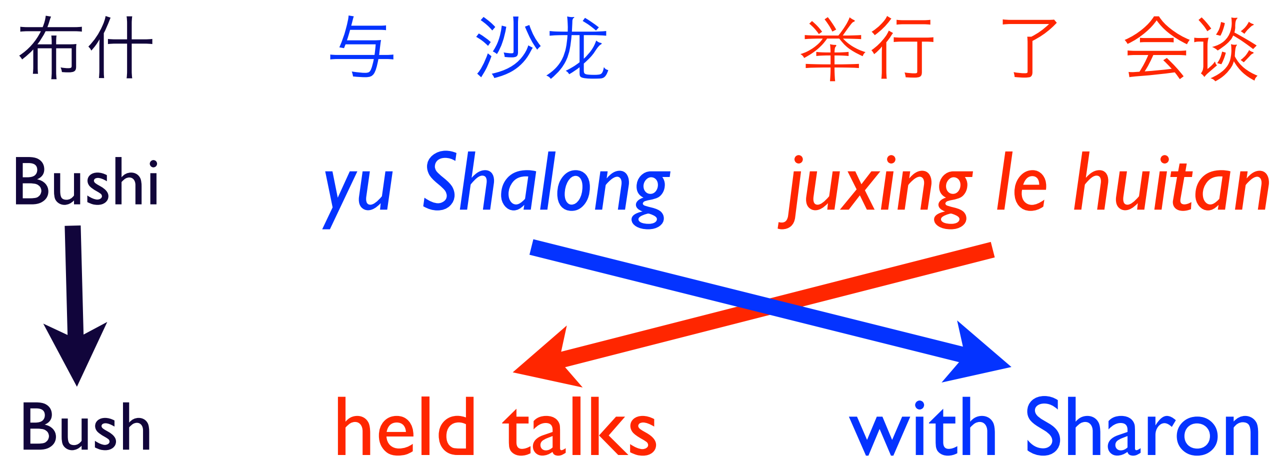
	with	Sharon held		meetings
Bush	with Sharon		held talks	
Bushi	*yu*	*Shalong*	*juxing le*	*huitan*

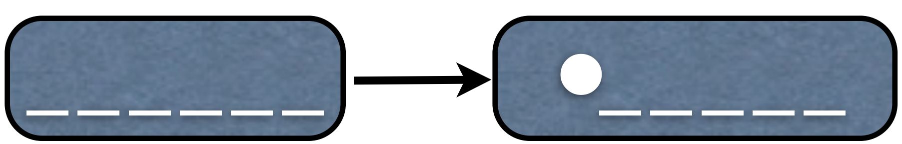
Phrase-based translation

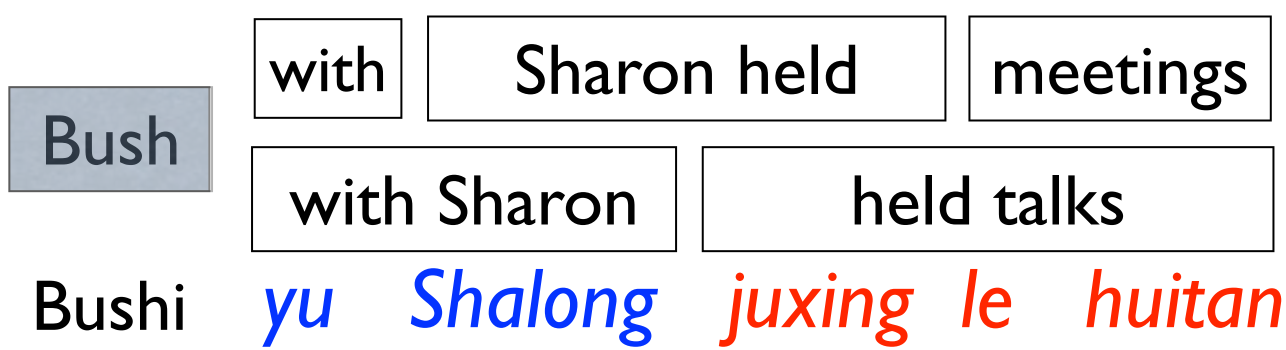




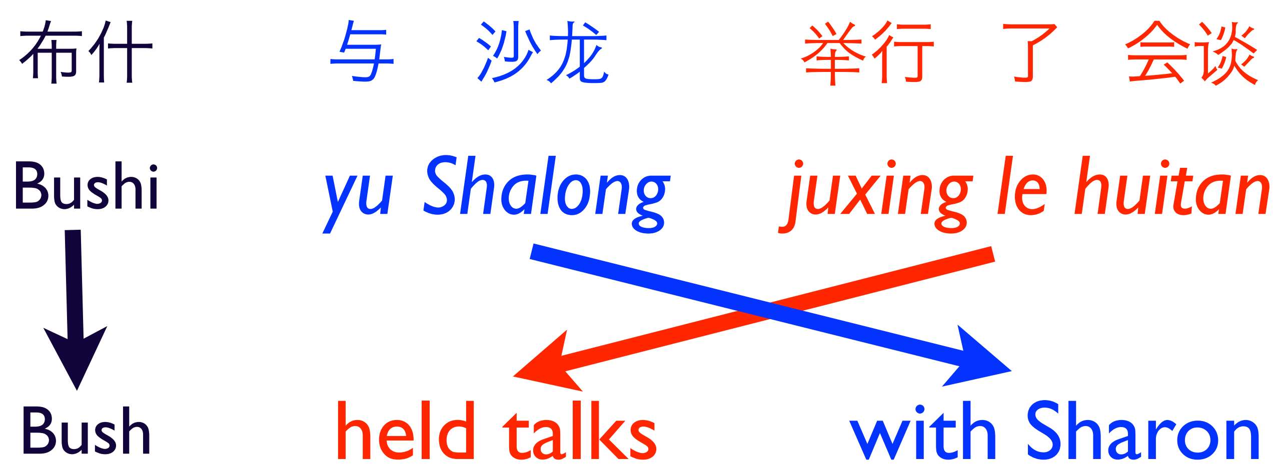
Phrase-based translation

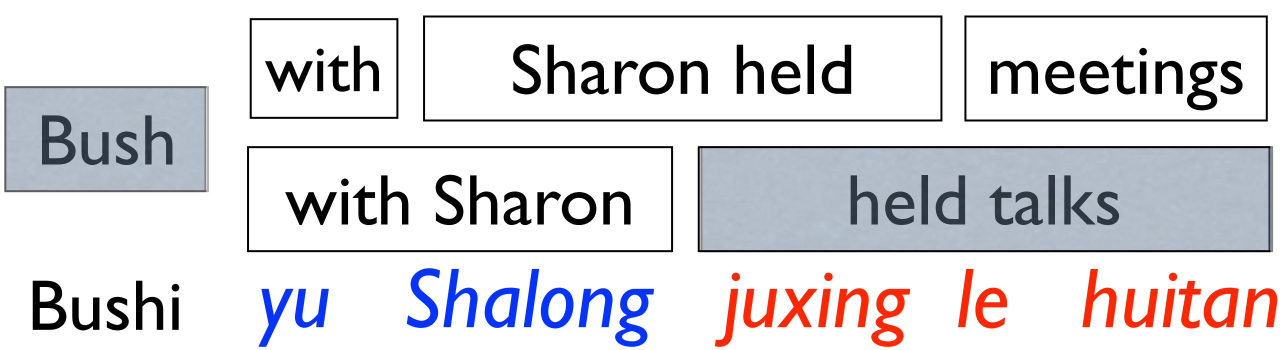




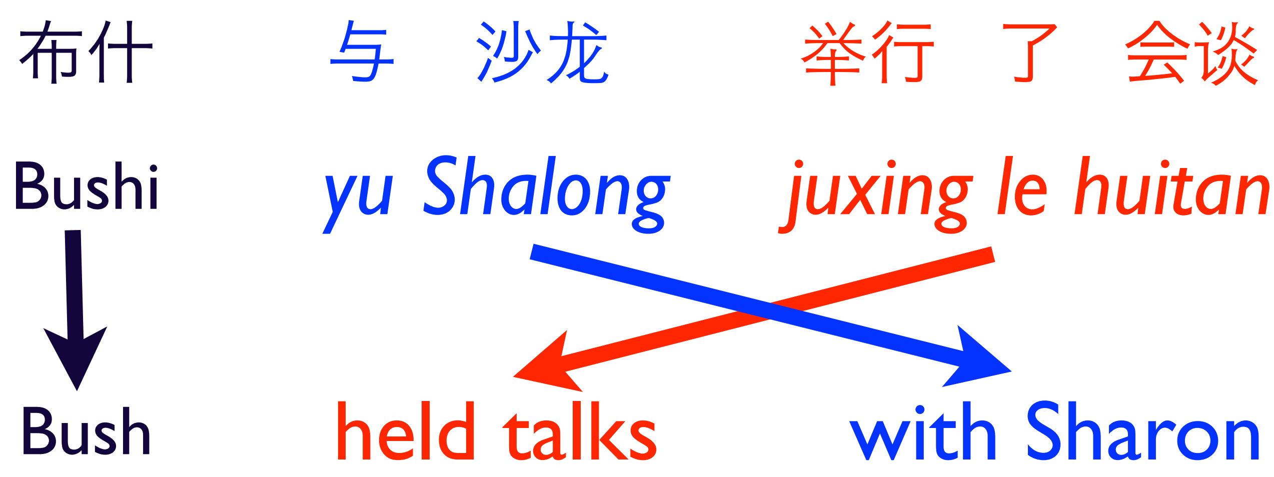


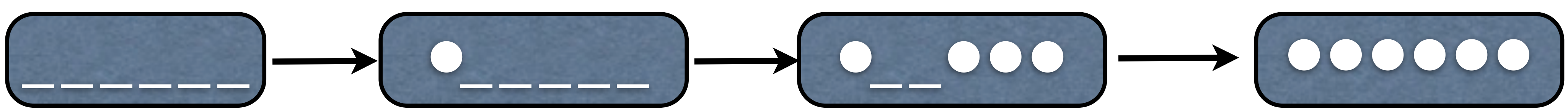
Phrase-based translation

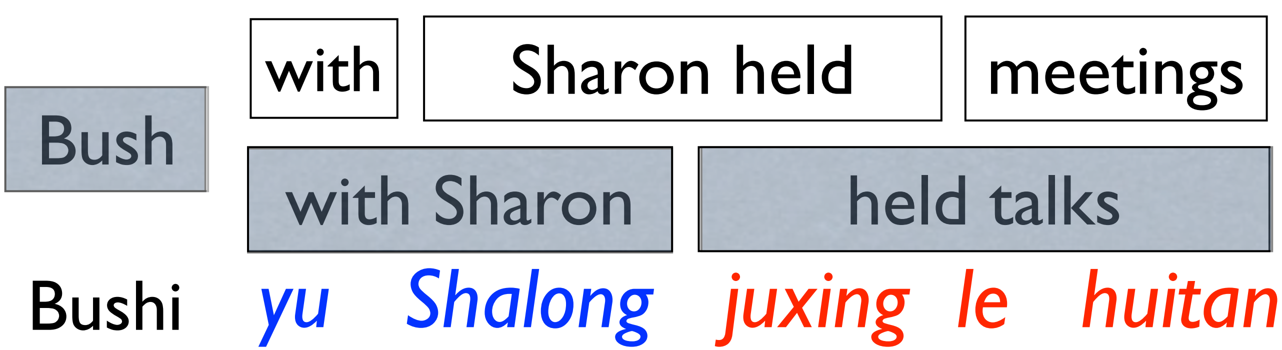




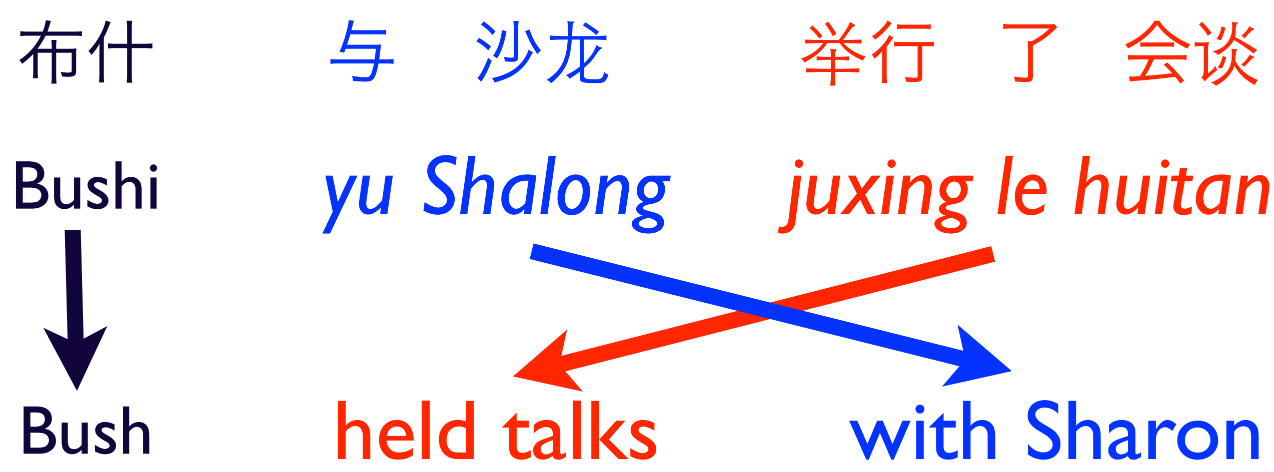
Phrase-based translation

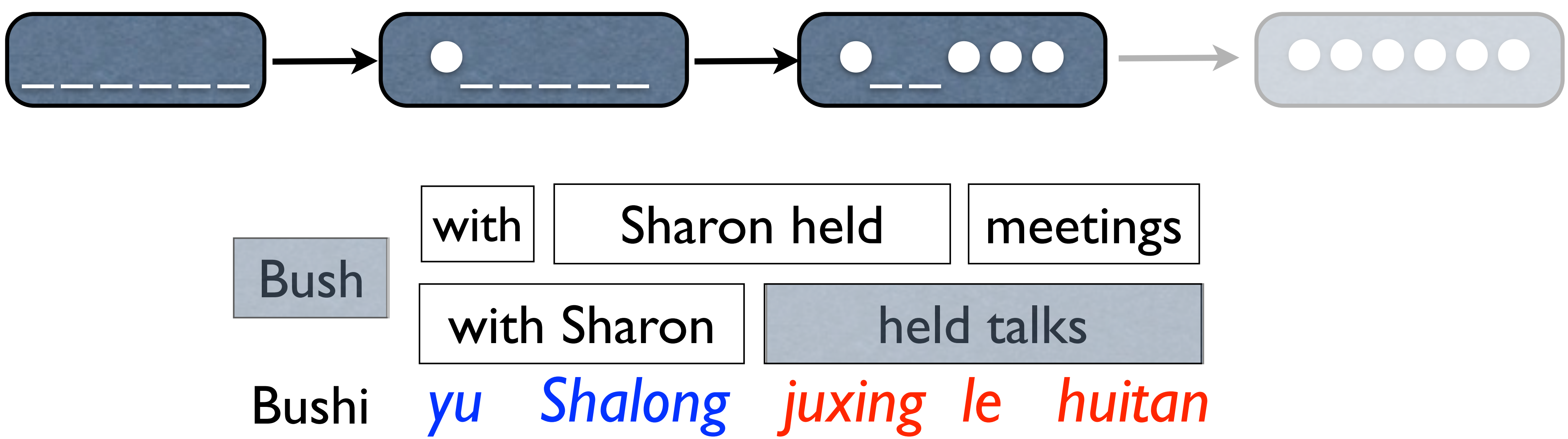




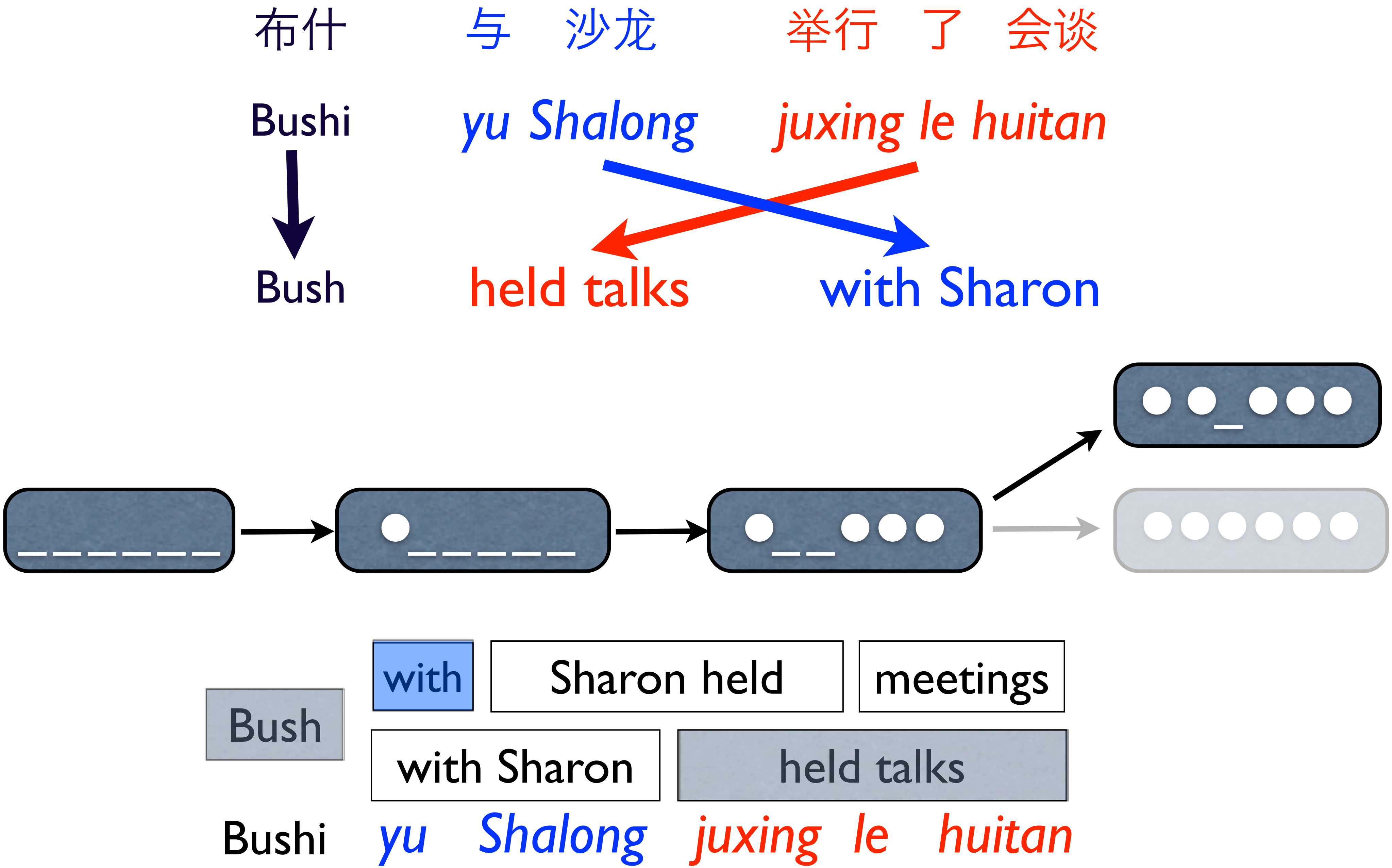


Phrase-based translation





Phrase-based translation



Language Model and Beam Search

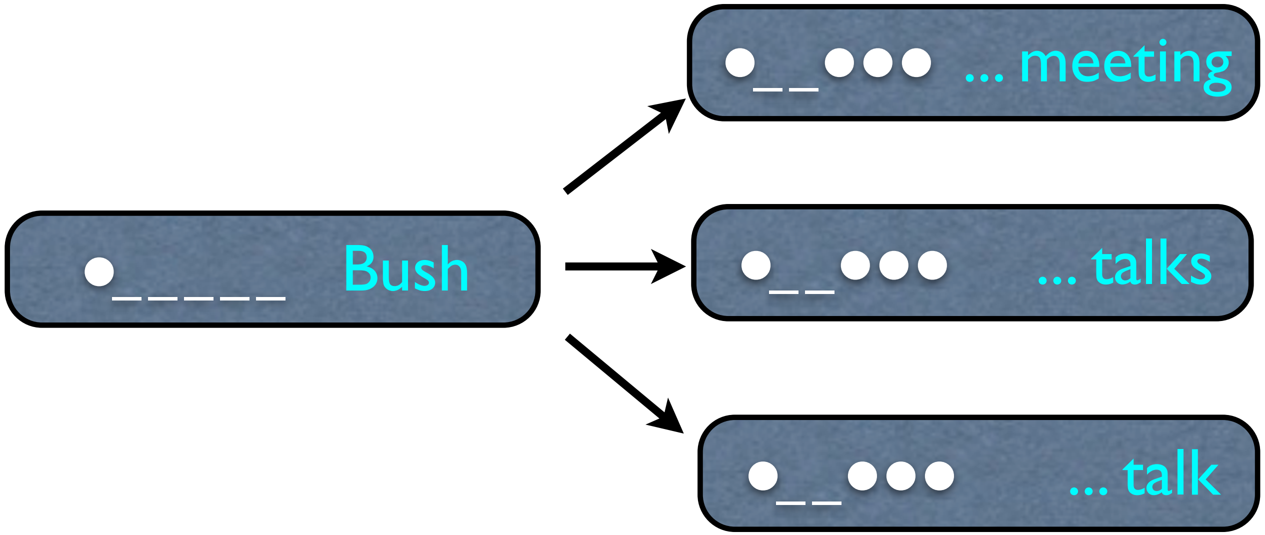
- split each -LM state into many +LM states

Language Model and Beam Search

- split each -LM state into many +LM states

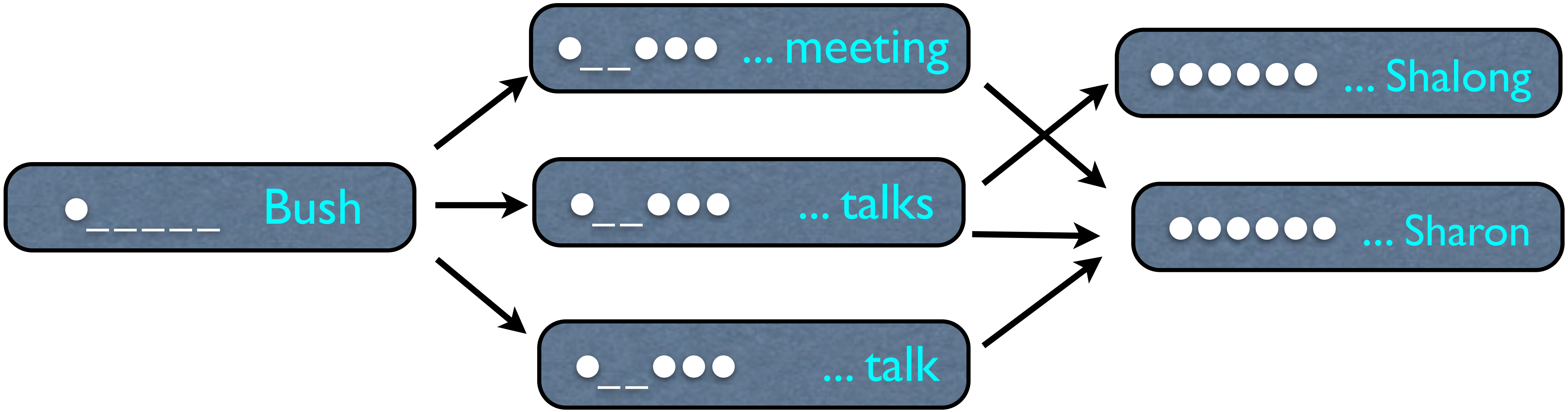
Language Model and Beam Search

- split each -LM state into many +LM states



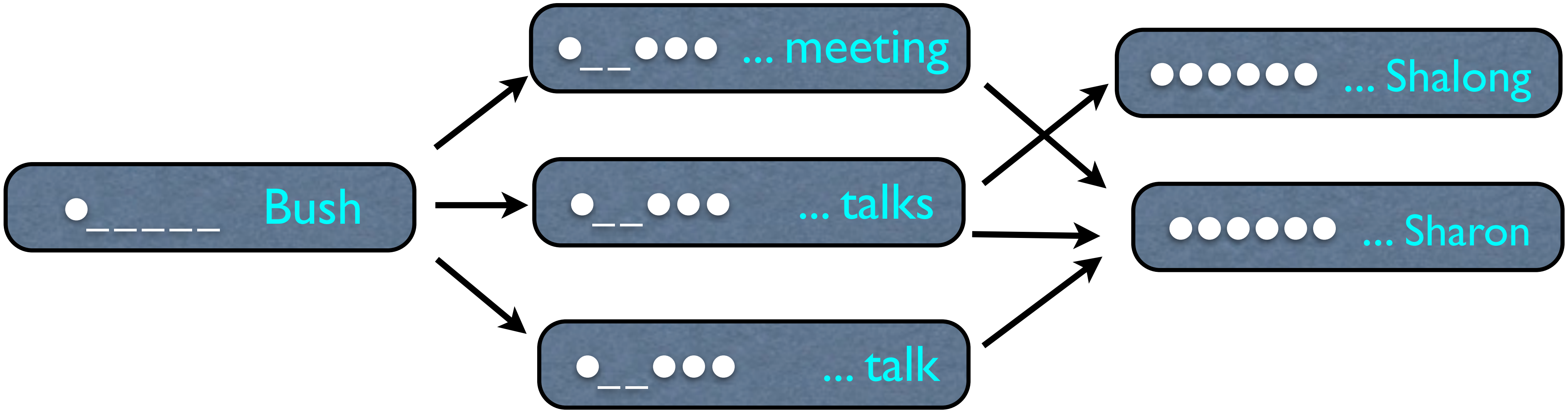
Language Model and Beam Search

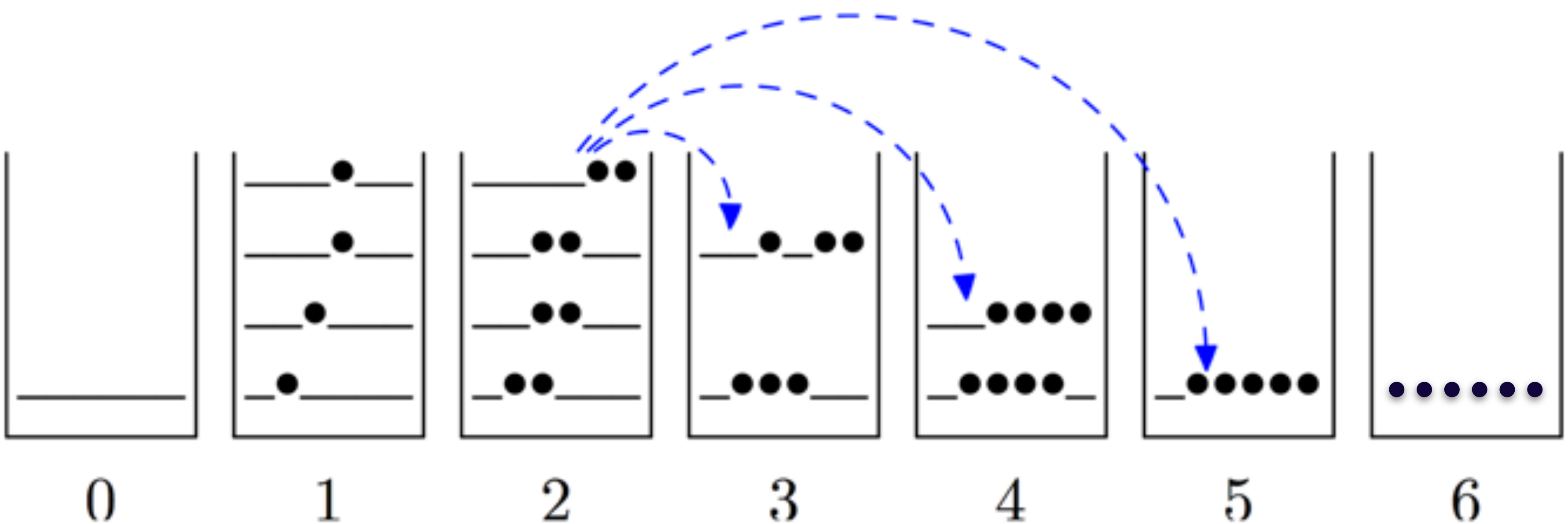
- split each -LM state into many +LM states



Language Model and Beam Search

- split each -LM state into many +LM states

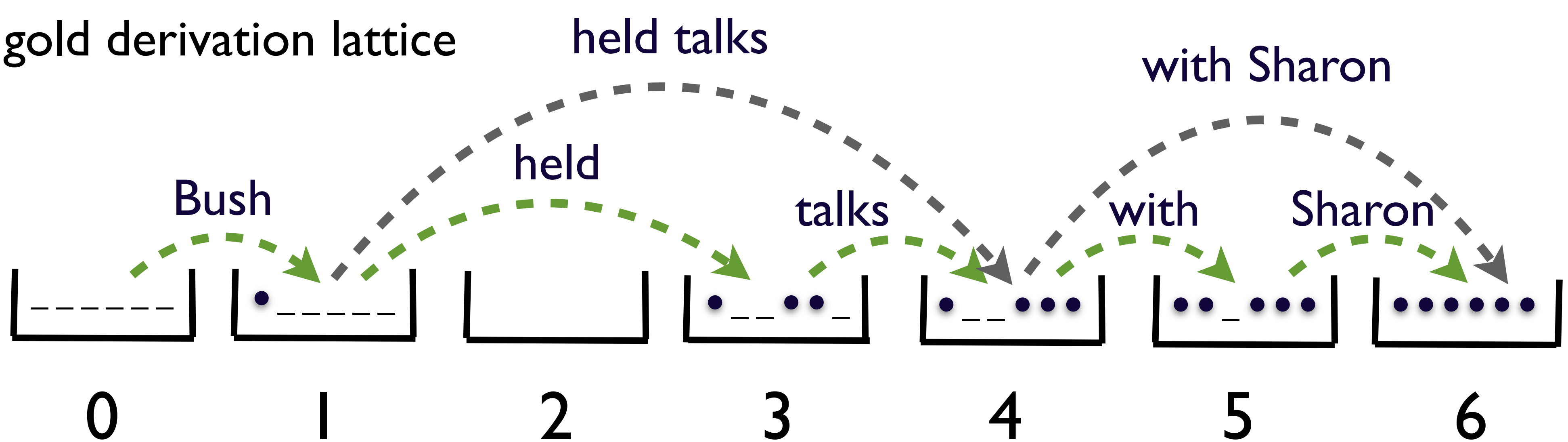




Forced Decoding

- both as data selection (more literal) and oracle derivations

Bushi yu Shalong juxing le huitan → *Bush held talks with Sharon*



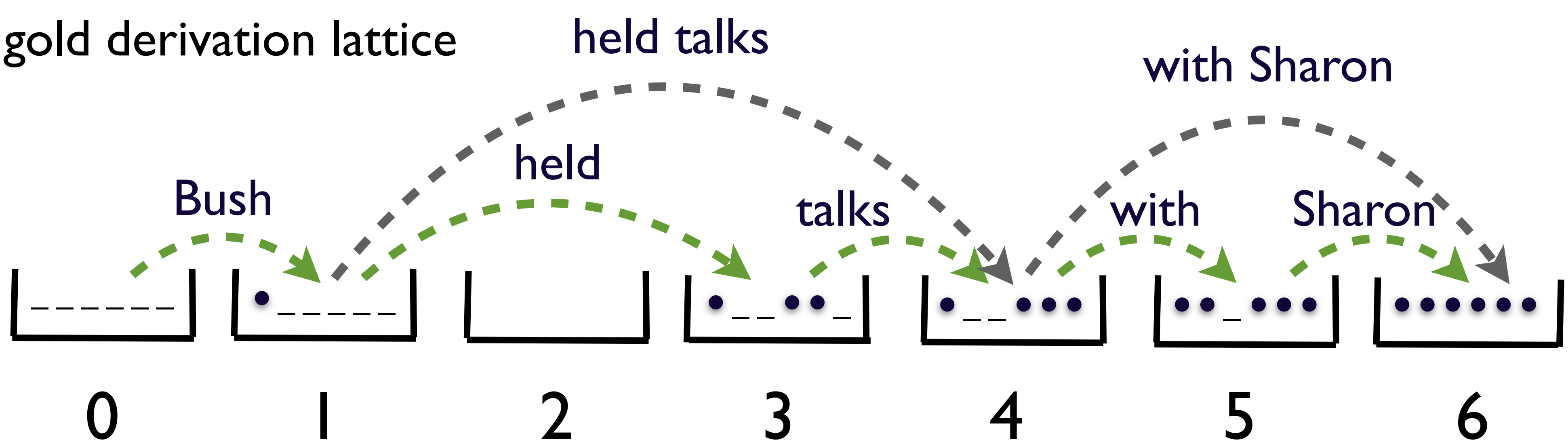
Forced Decoding

- both as data selection (more literal) and oracle derivations

Bushi yu Shalong juxing le huitan → *Bush held talks with Sharon*

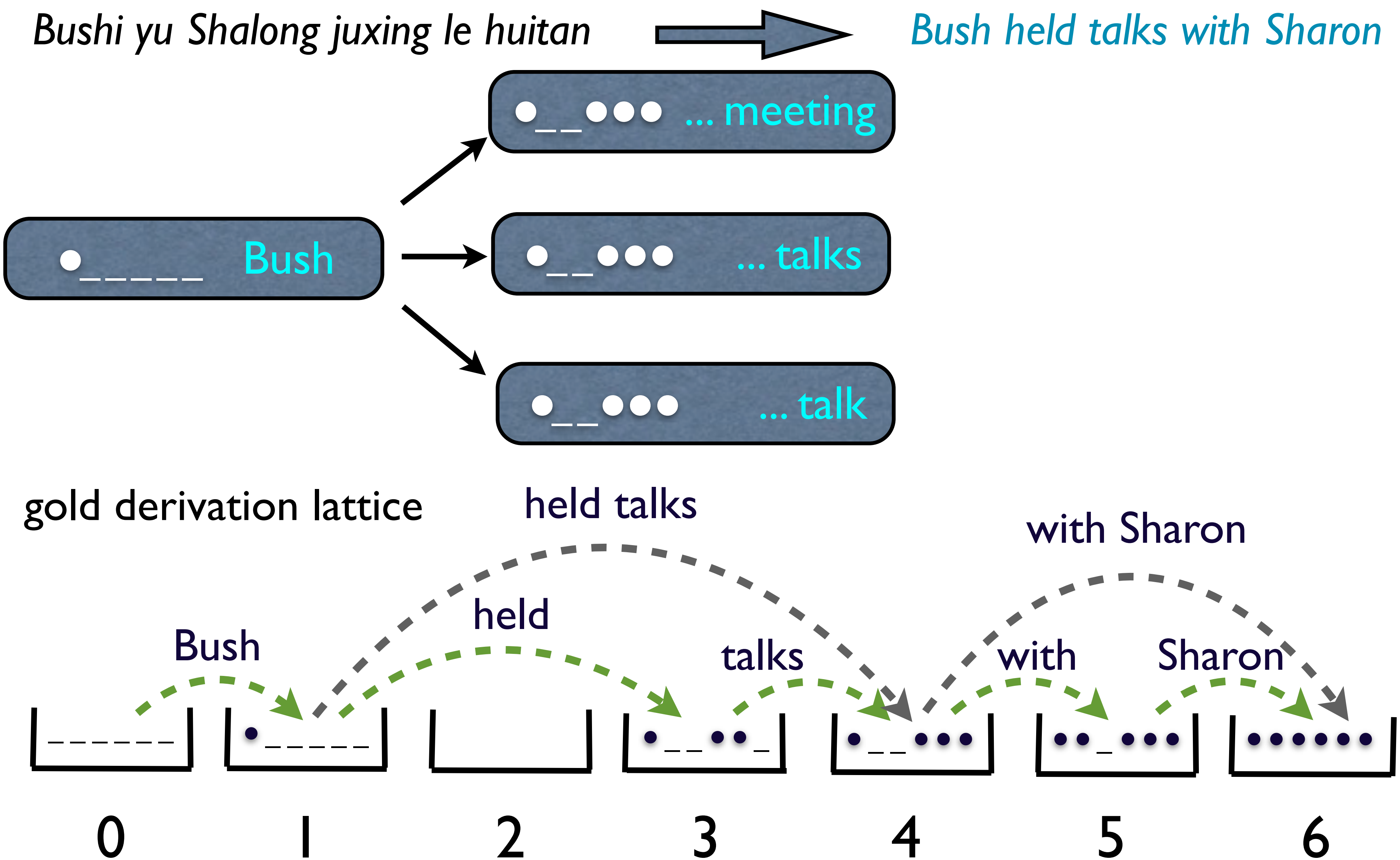
●______ Bush

gold derivation lattice



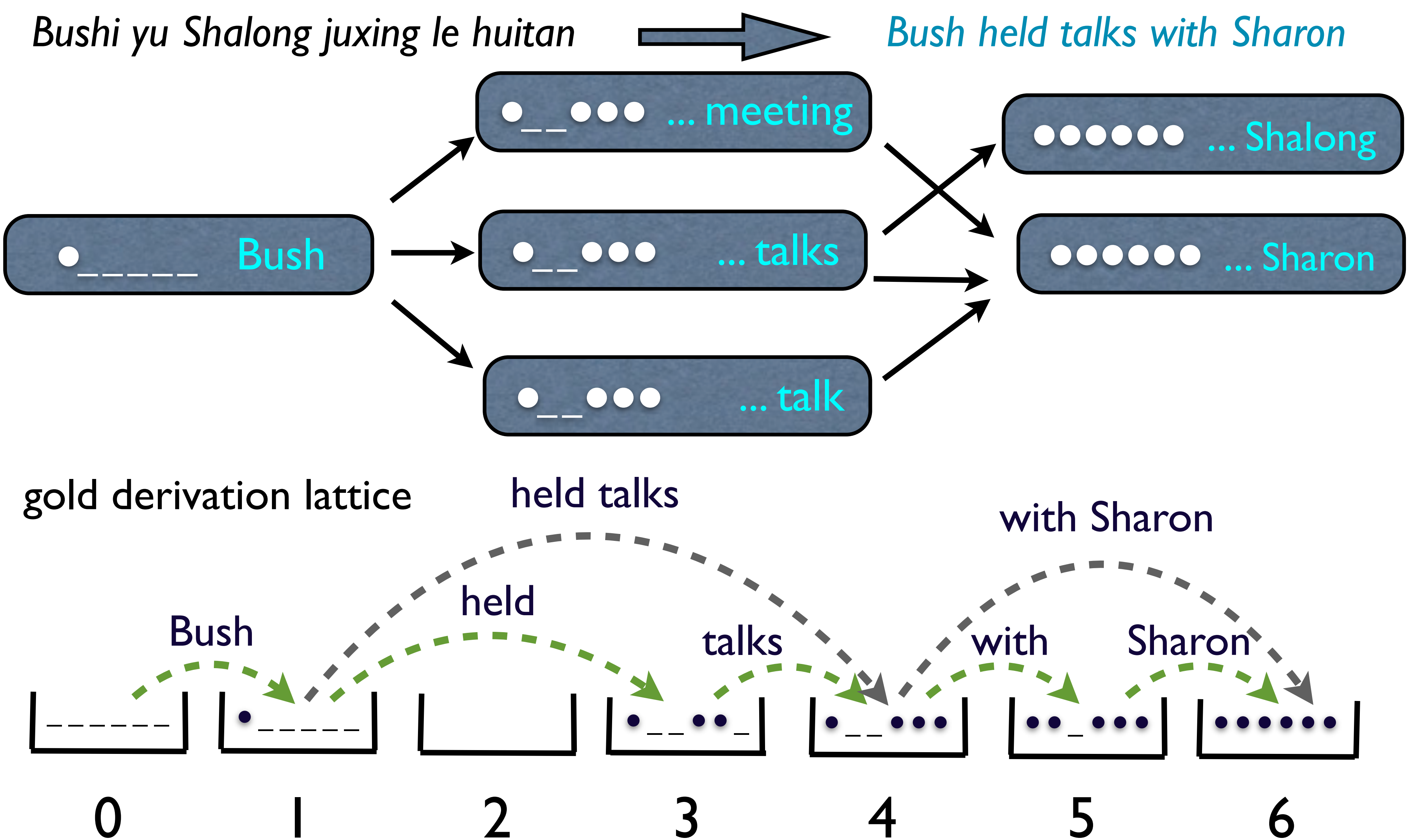
Forced Decoding

- both as data selection (more literal) and oracle derivations



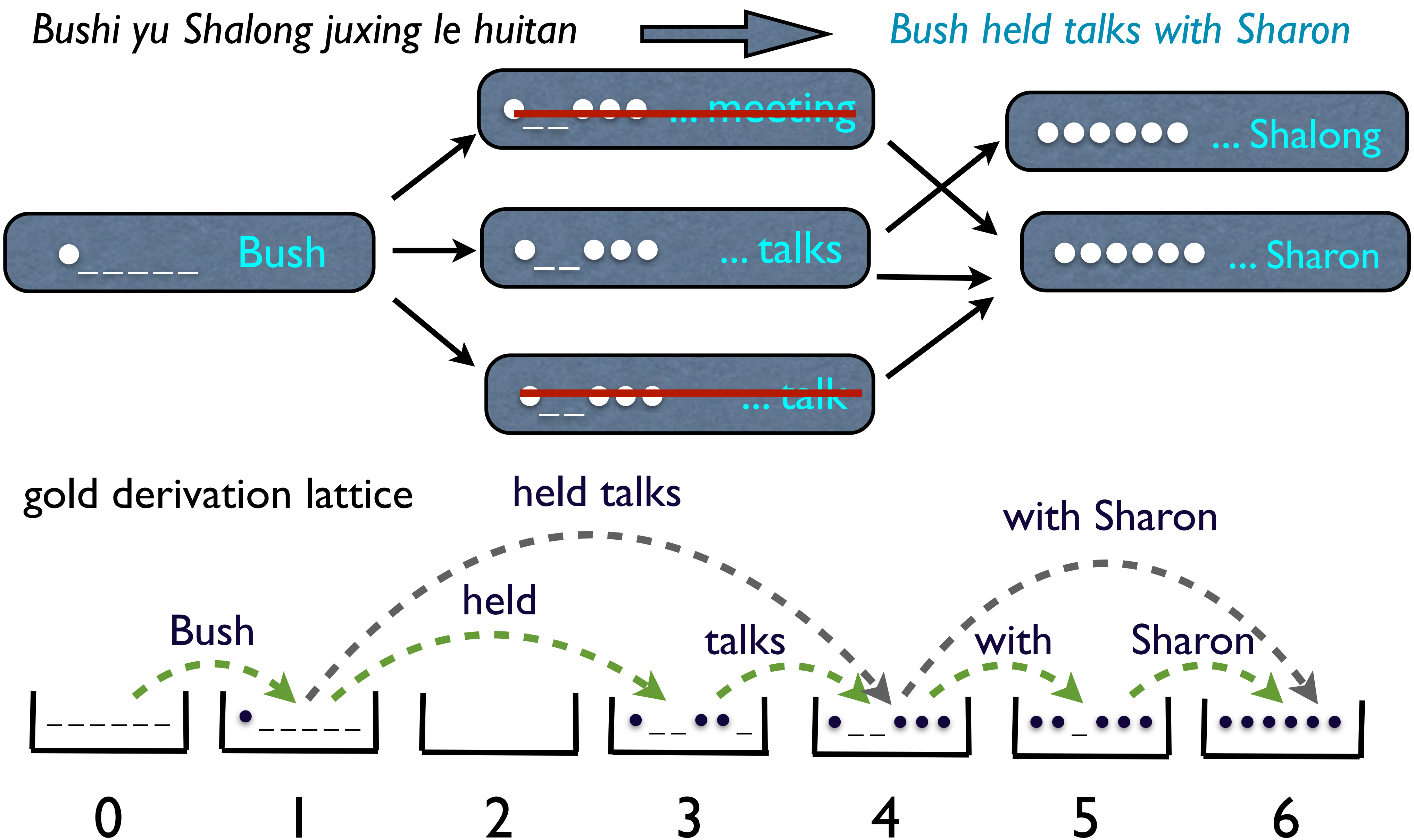
Forced Decoding

- both as data selection (more literal) and oracle derivations



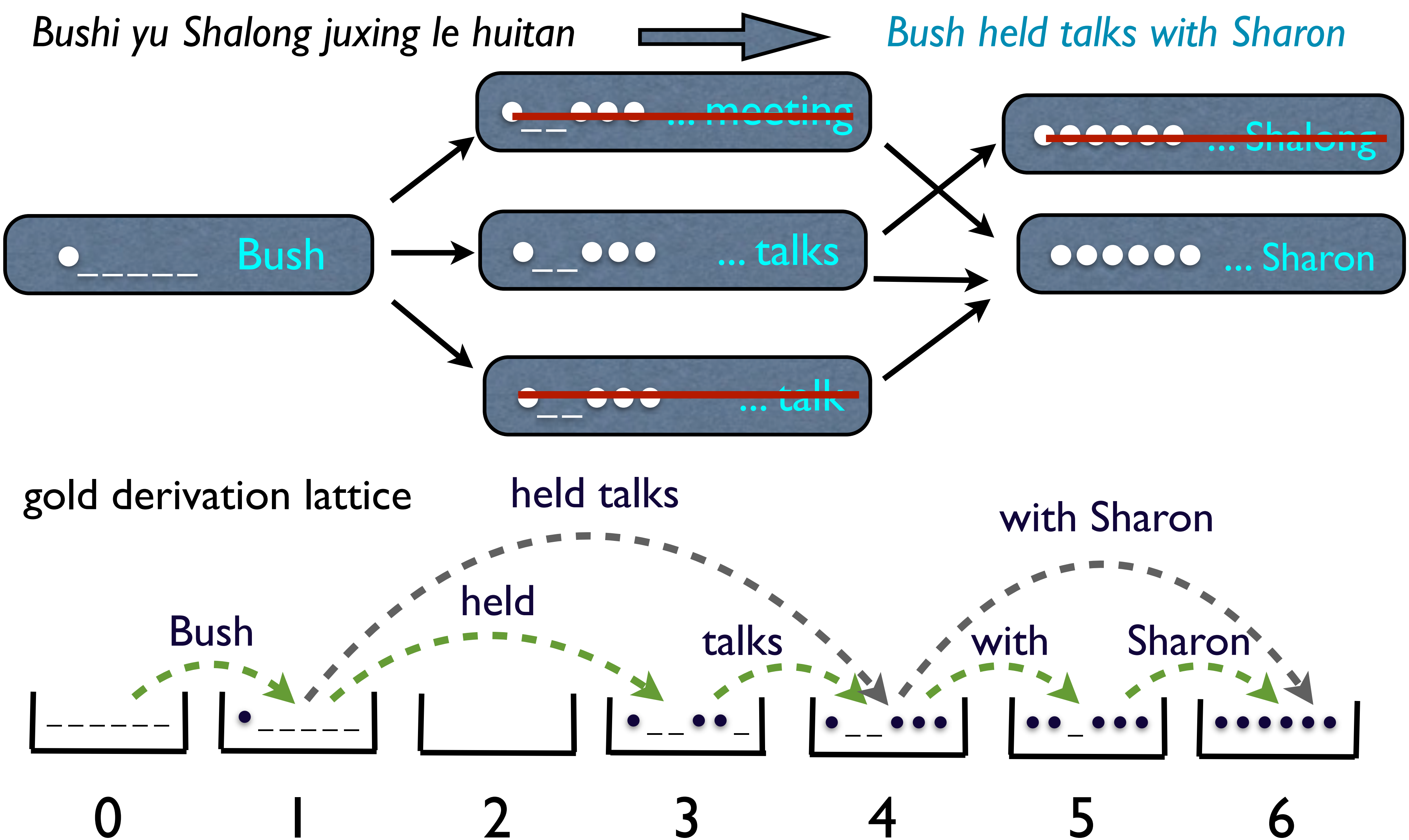
Forced Decoding

- both as data selection (more literal) and oracle derivations



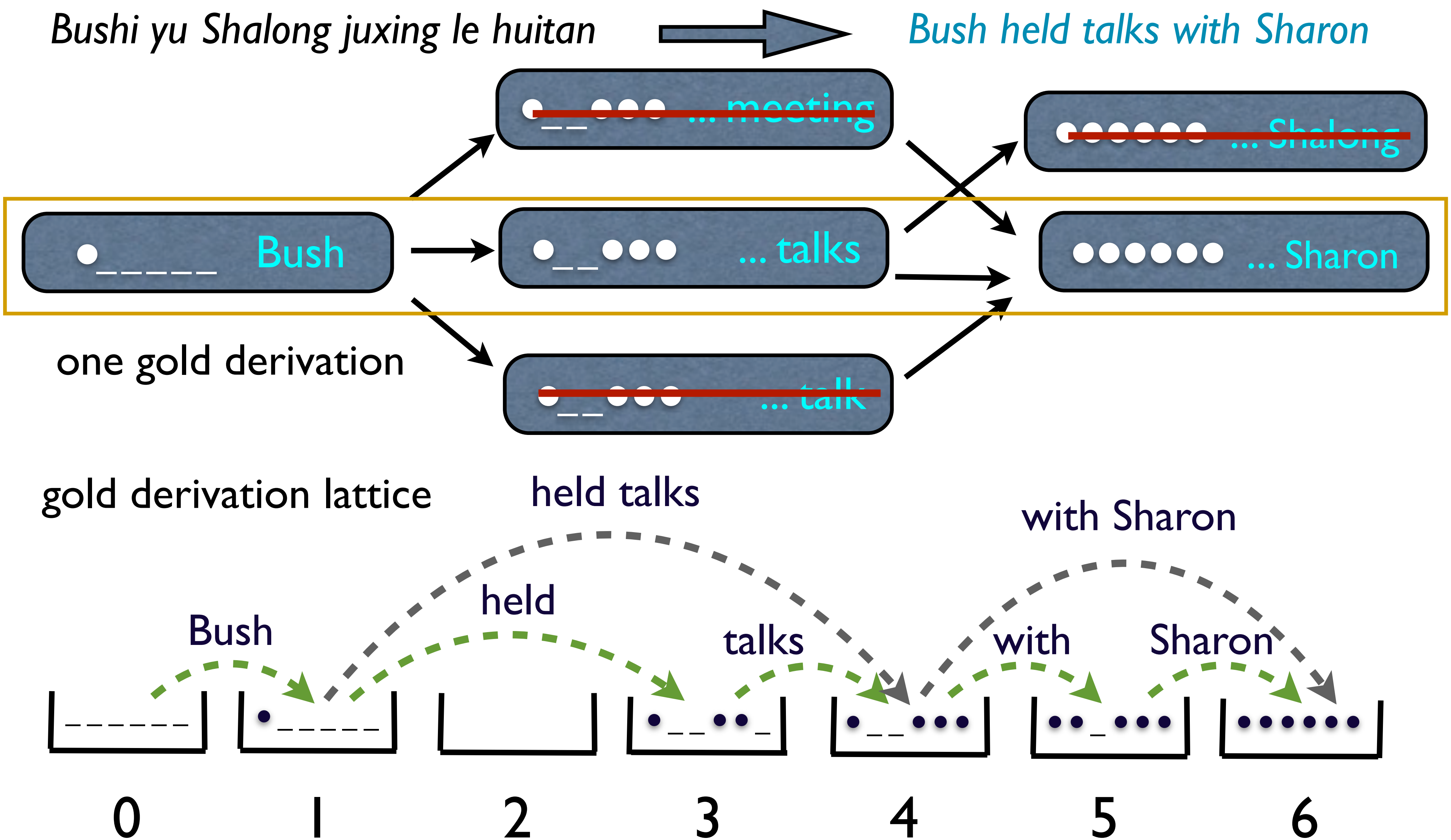
Forced Decoding

- both as data selection (more literal) and oracle derivations



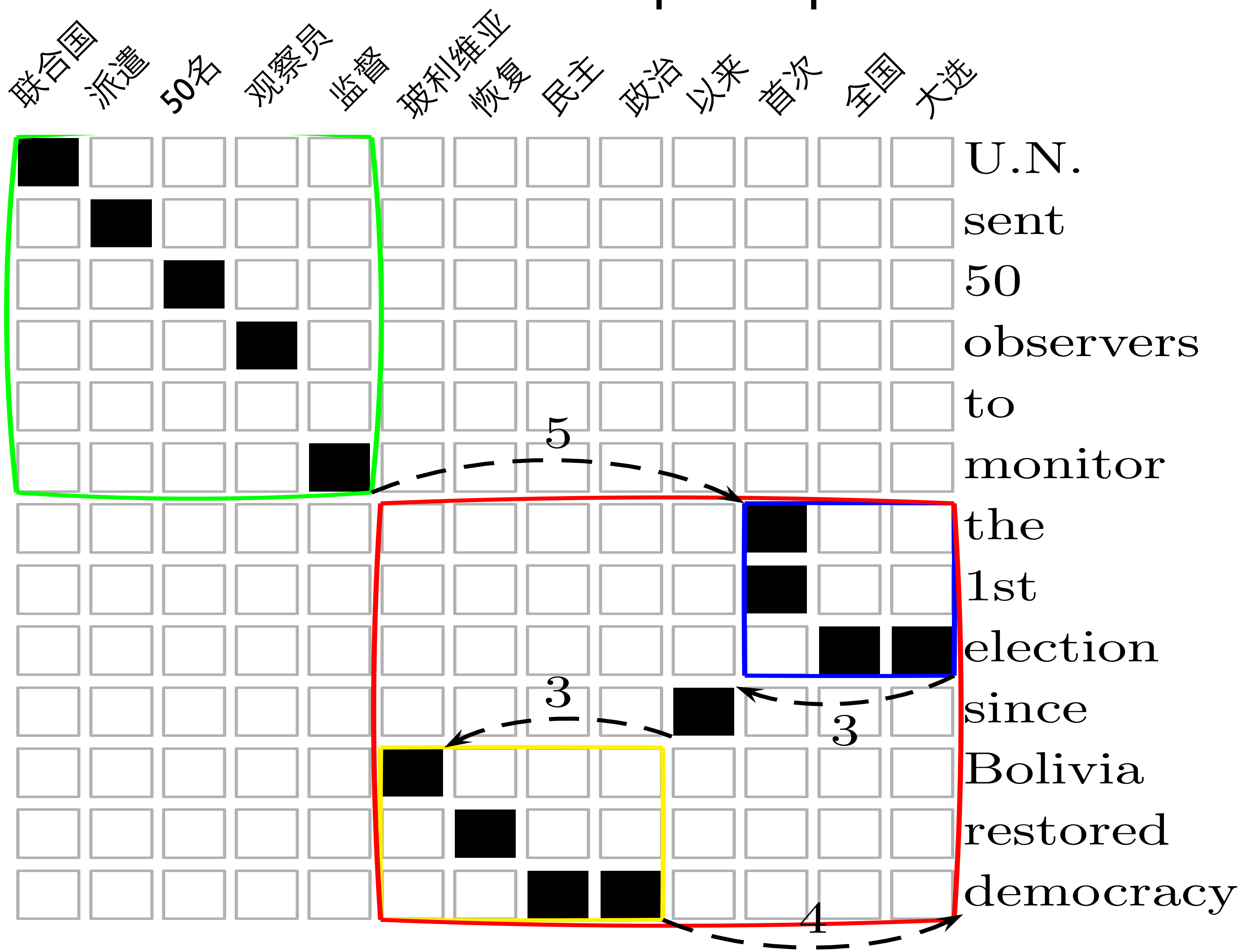
Forced Decoding

- both as data selection (more literal) and oracle derivations



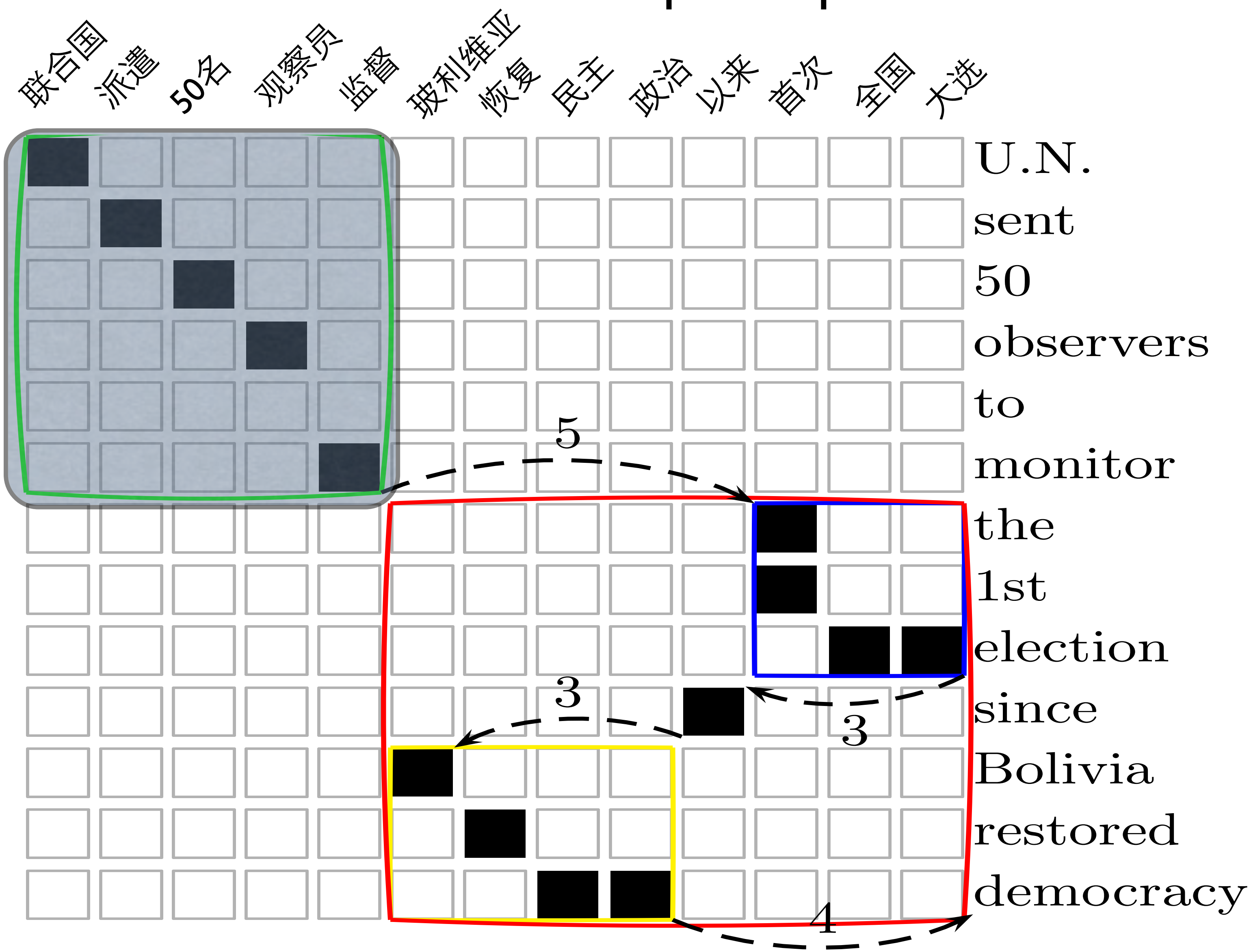
Unreachable Sentences and Prefix

- distortion limit causes unreachability (hiero would be better)
  - but we can still use reachable prefix-pairs of unreachable pairs



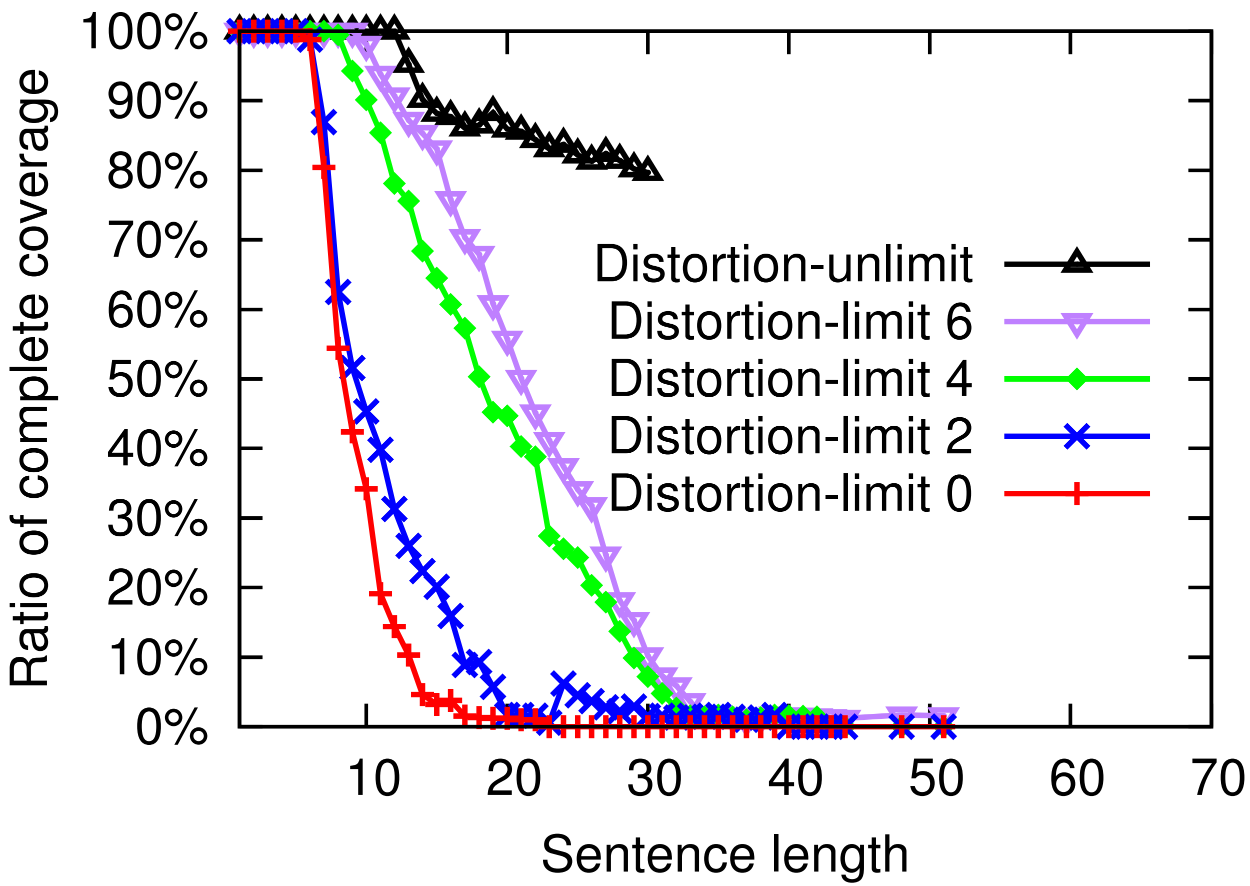
Unreachable Sentences and Prefix

- distortion limit causes unreachability (hiero would be better)
  - but we can still use reachable prefix-pairs of unreachable pairs



Sentence/Word Reachability Ratio

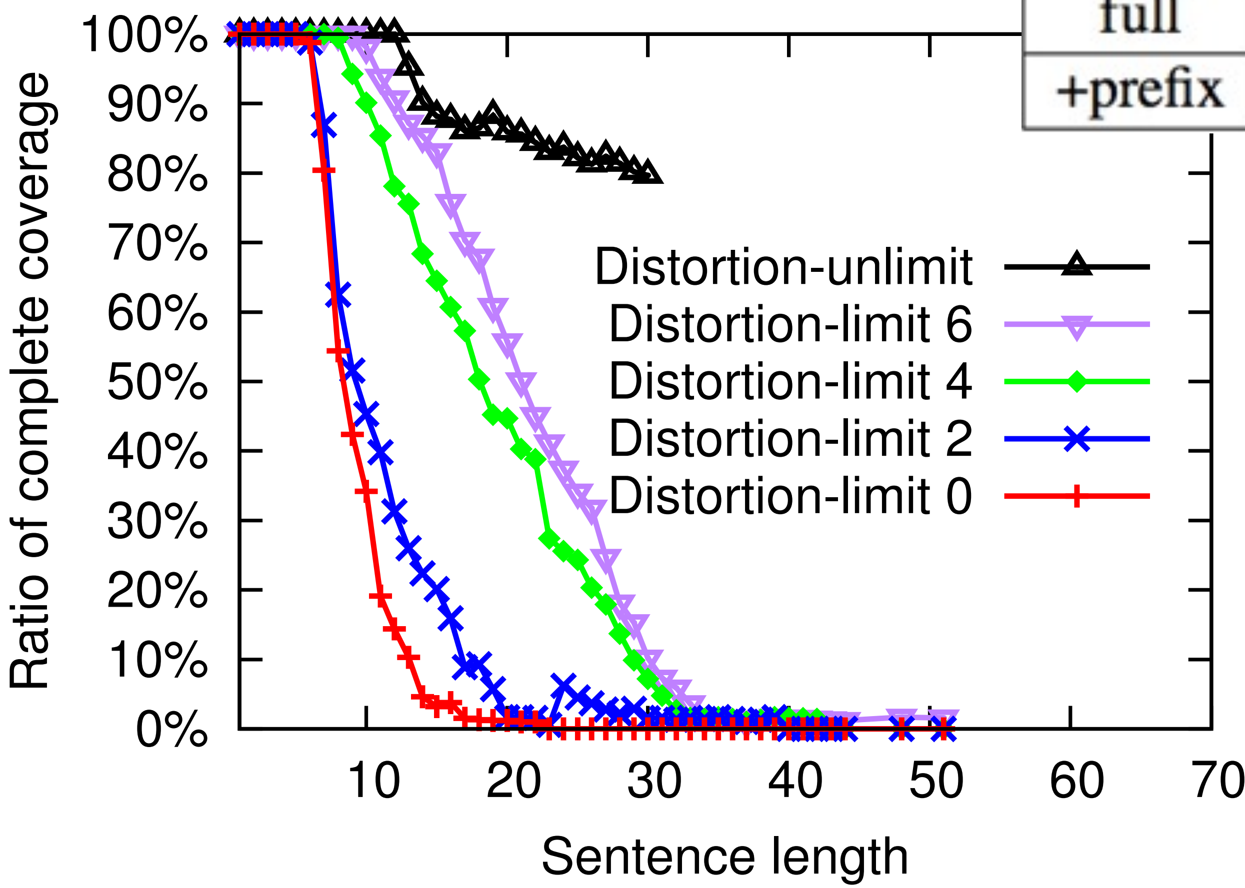
- how many sentences pairs pass forced decoding?
  - the ratio drops dramatically as sentences get longer
  - prefixes boost coverage



Sentence/Word Reachability Ratio

- how many sentences pairs pass forced decoding?
  - the ratio drops dramatically as sentences get longer
  - prefixes boost coverage

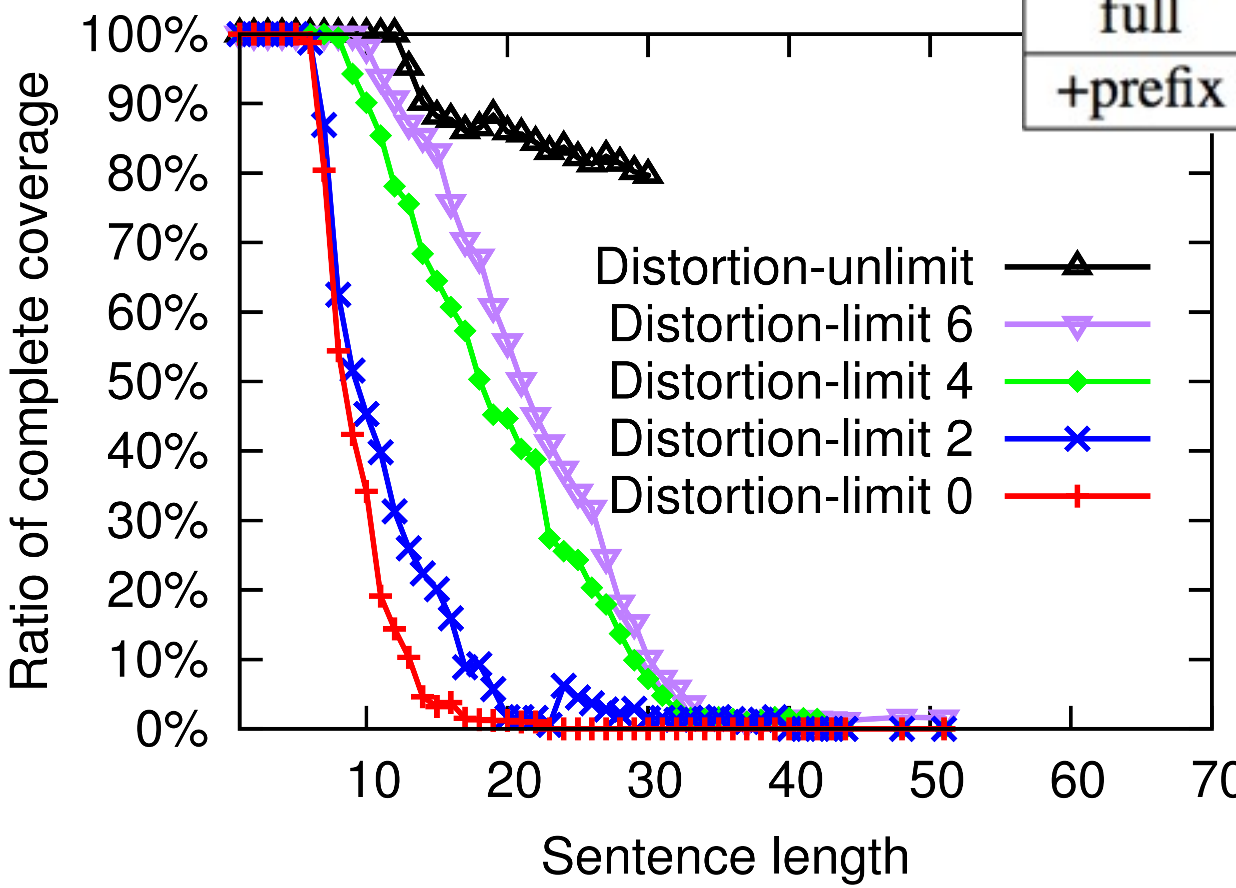
	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

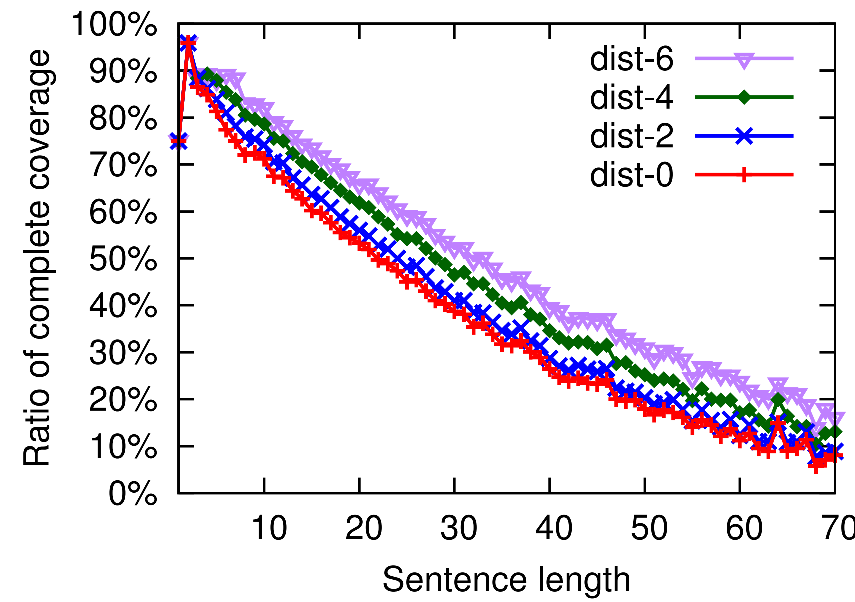


Sentence/Word Reachability Ratio

- how many sentences pairs pass forced decoding?
  - the ratio drops dramatically as sentences get longer
  - prefixes boost coverage

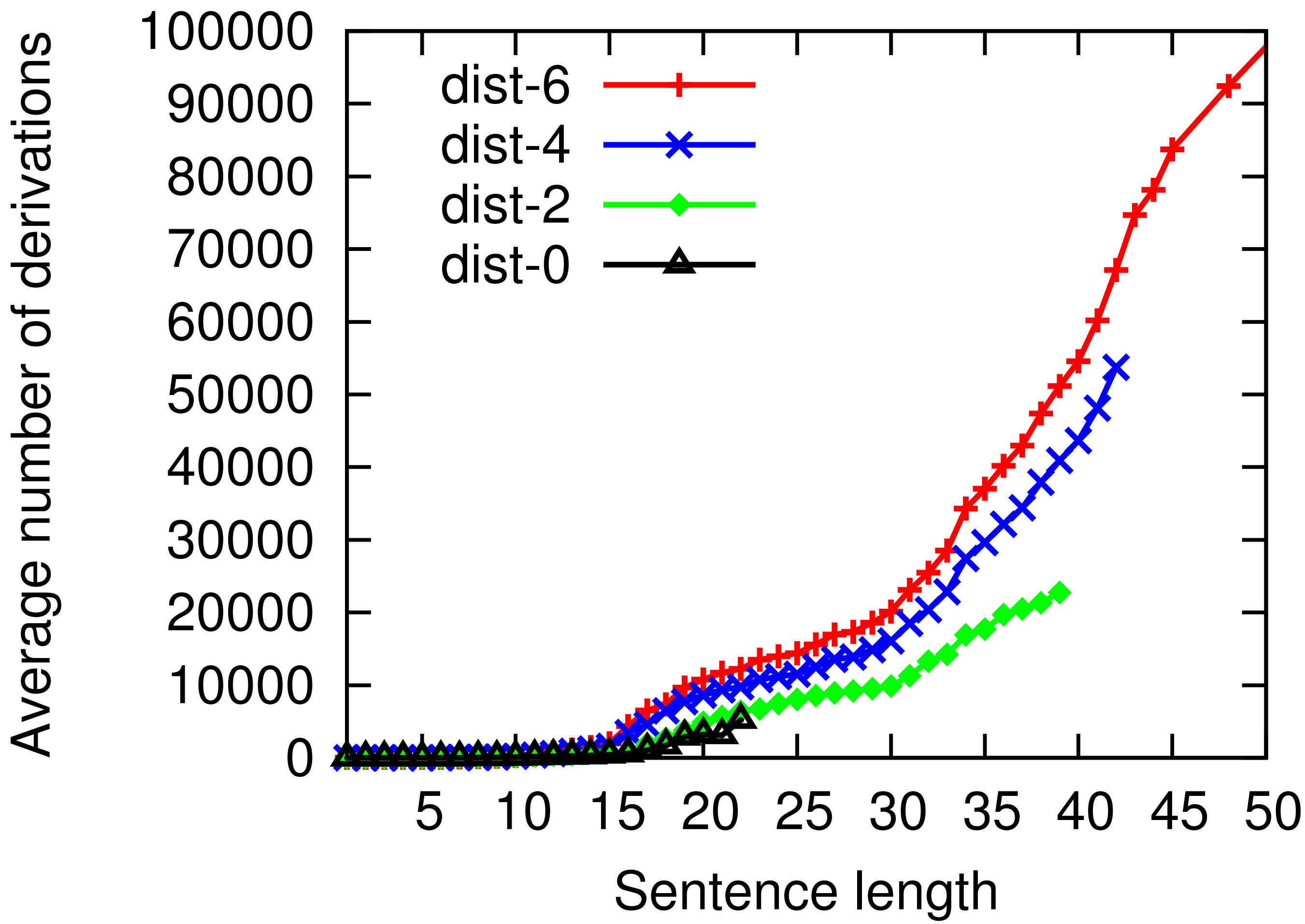
	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%





Number of Gold Derivations

- exponential in sentence length (on fully reachables)
  - these are the “latent variables” in learning

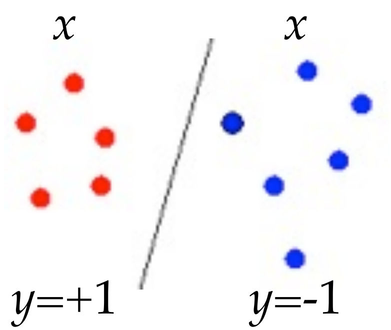


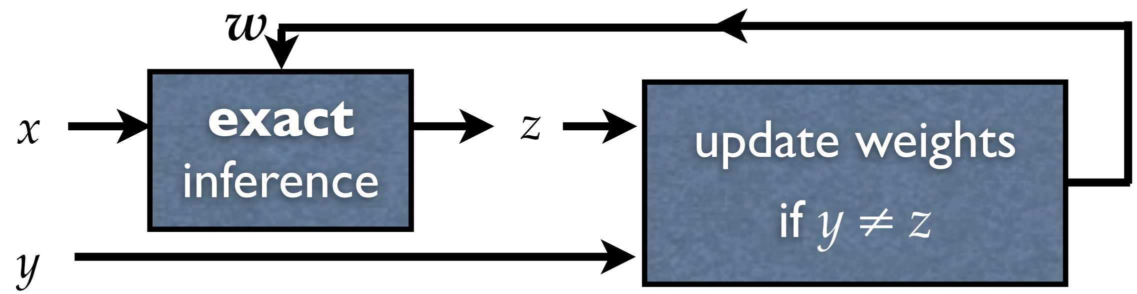
Outline

- Background: Phrase-based Translation (Koehn, 2004)
- Forced Decoding
- Violation-Fixing Perceptron for MT Training
  - Update strategy
  - Feature design
- Experiments

Structured Perceptron (Collins 02)

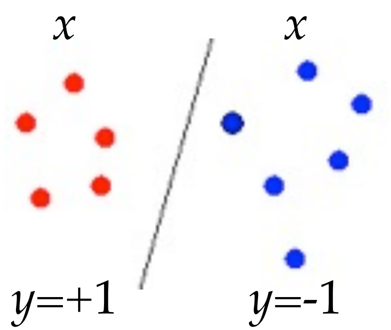
binary classification

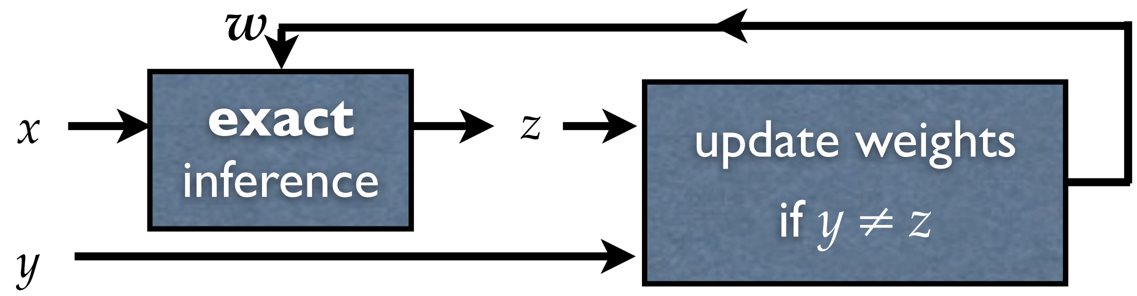




Structured Perceptron (Collins 02)

binary classification

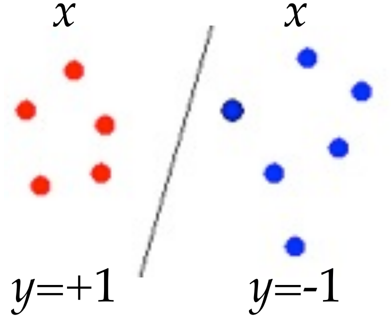




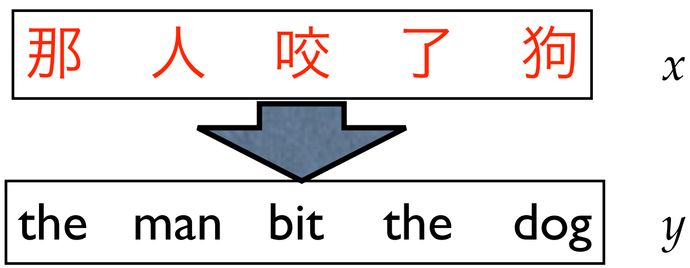
structured classification

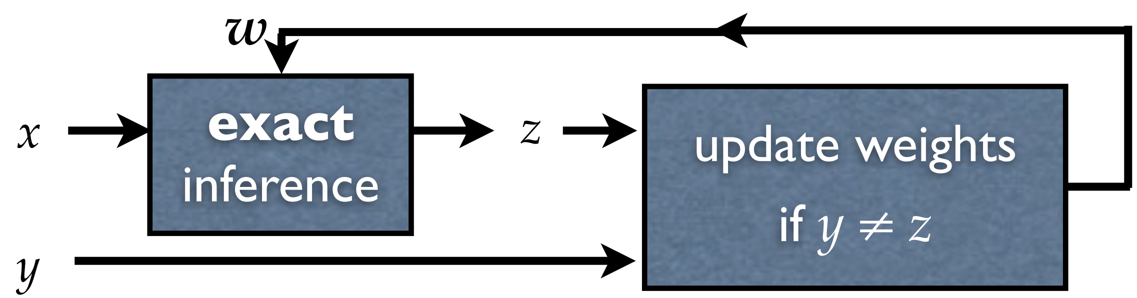
Structured Perceptron (Collins 02)

binary classification



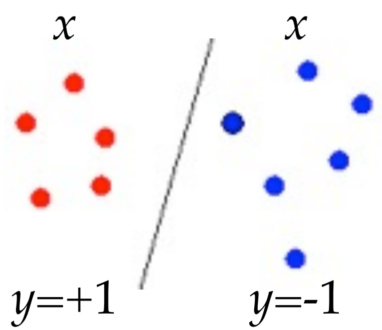
structured classification



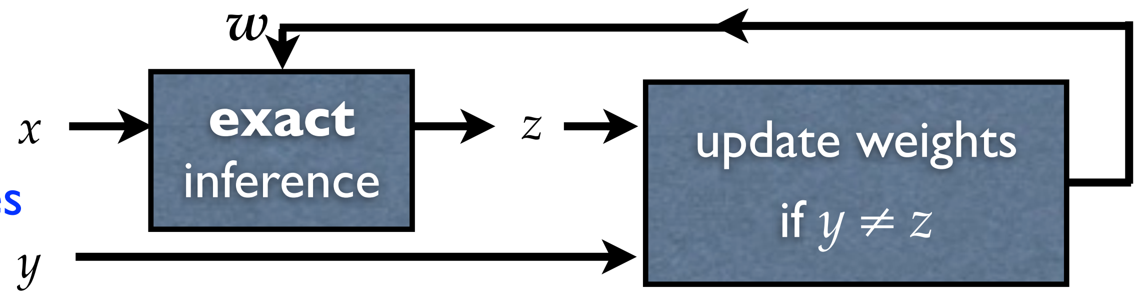


Structured Perceptron (Collins 02)

binary classification

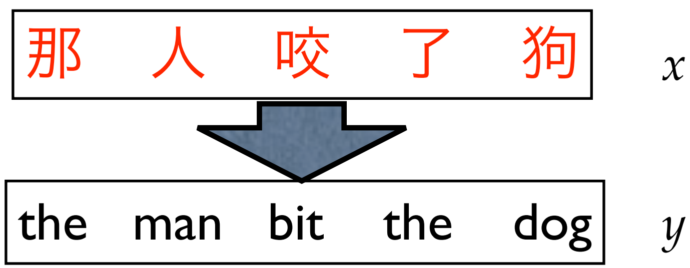


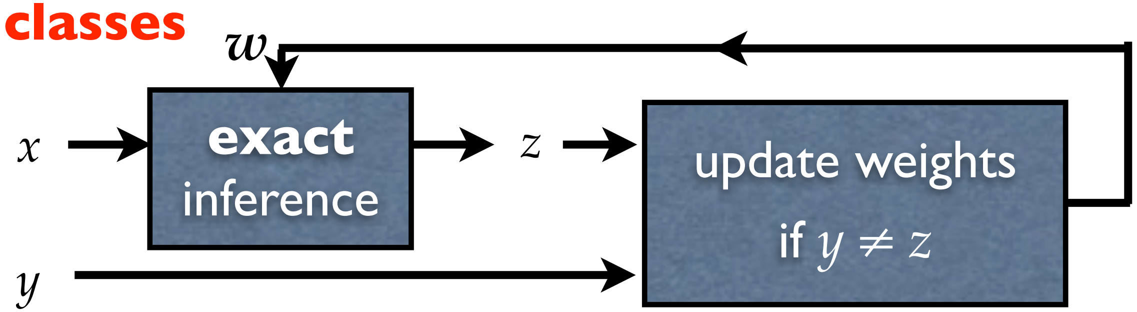
constant
# of classes



structured classification

exponential
# of classes

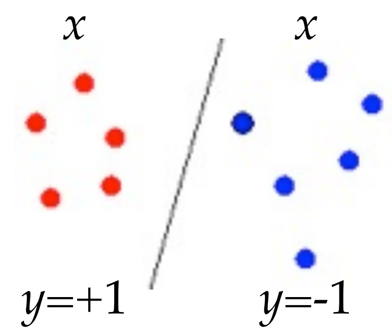




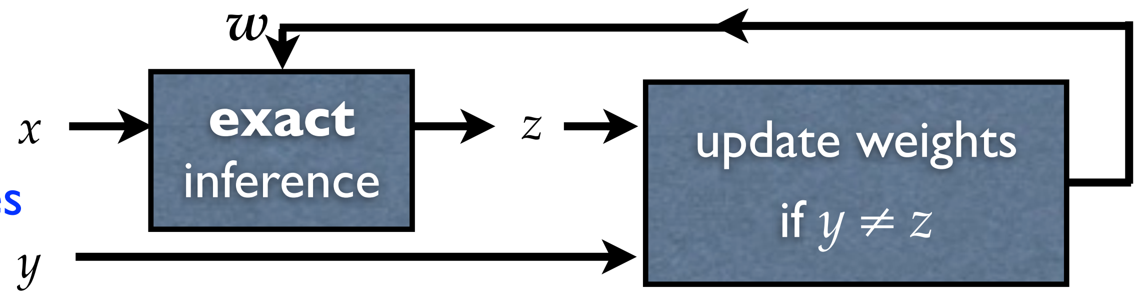
- challenges in applying perceptron for MT
  - the inference (decoding) is vastly inexact (beam search)
  - we know standard perceptron doesn't work for MT
- intuition: the learner should fix the search error first

Structured Perceptron (Collins 02)

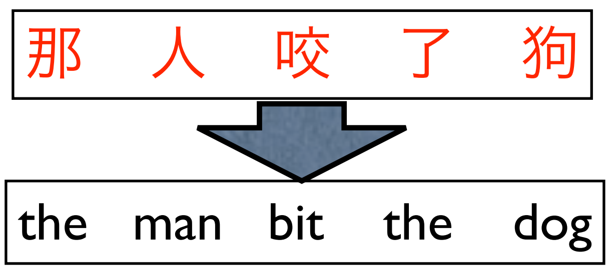
binary classification



constant
# of classes



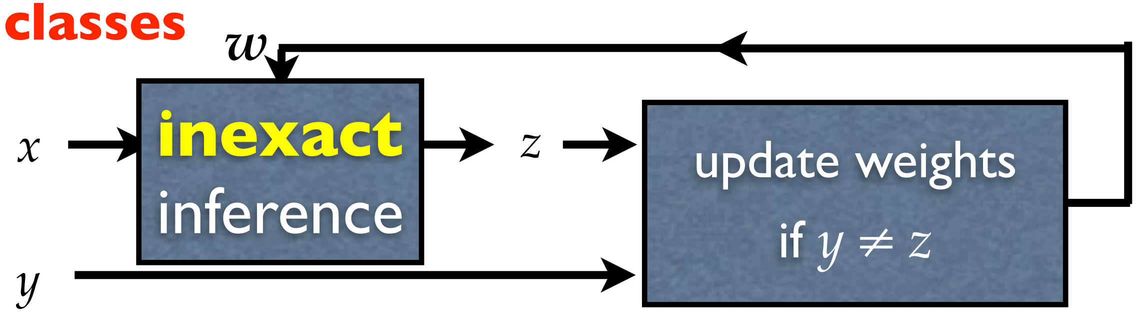
structured classification



x

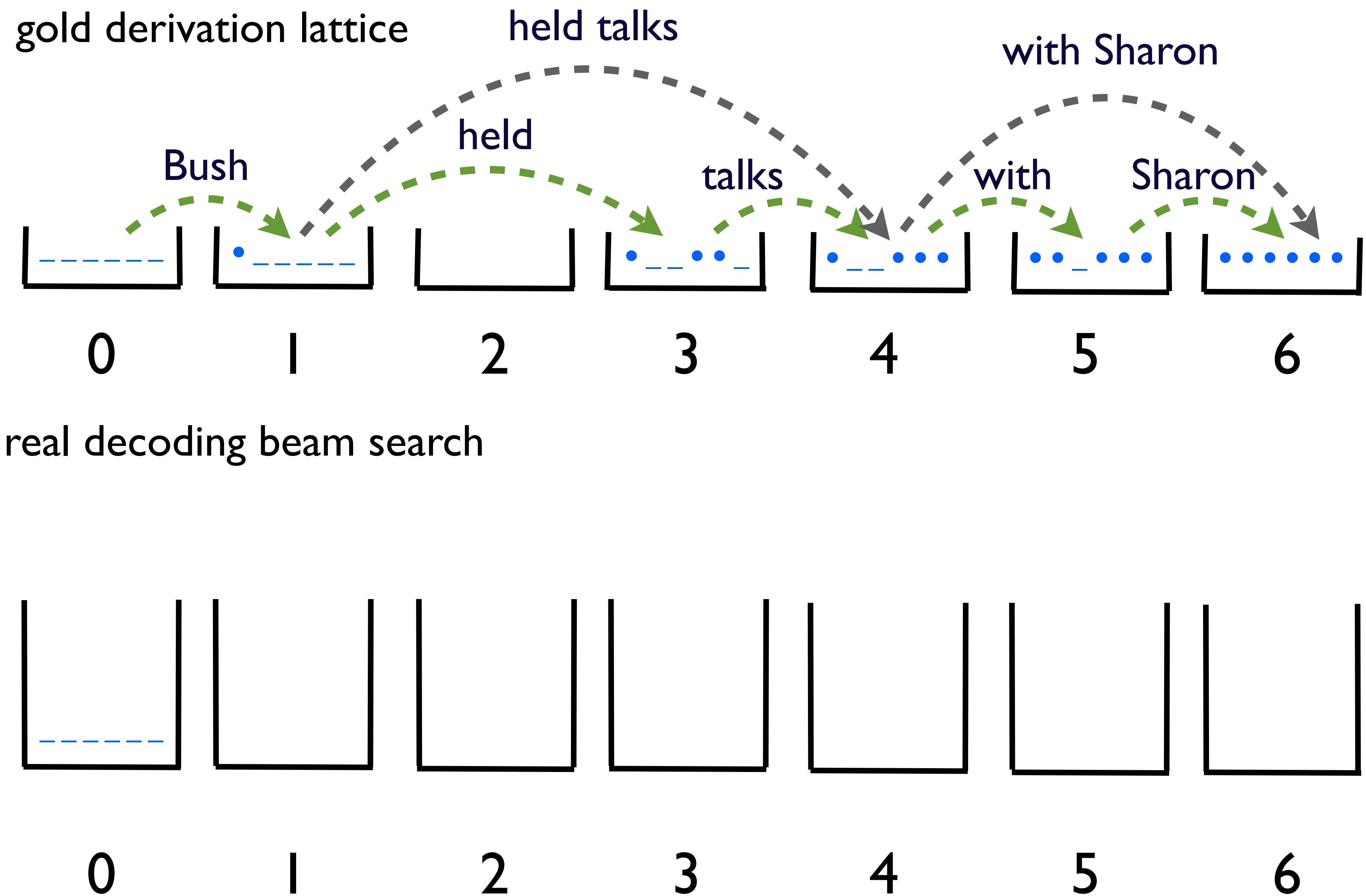
y

exponential
# of classes

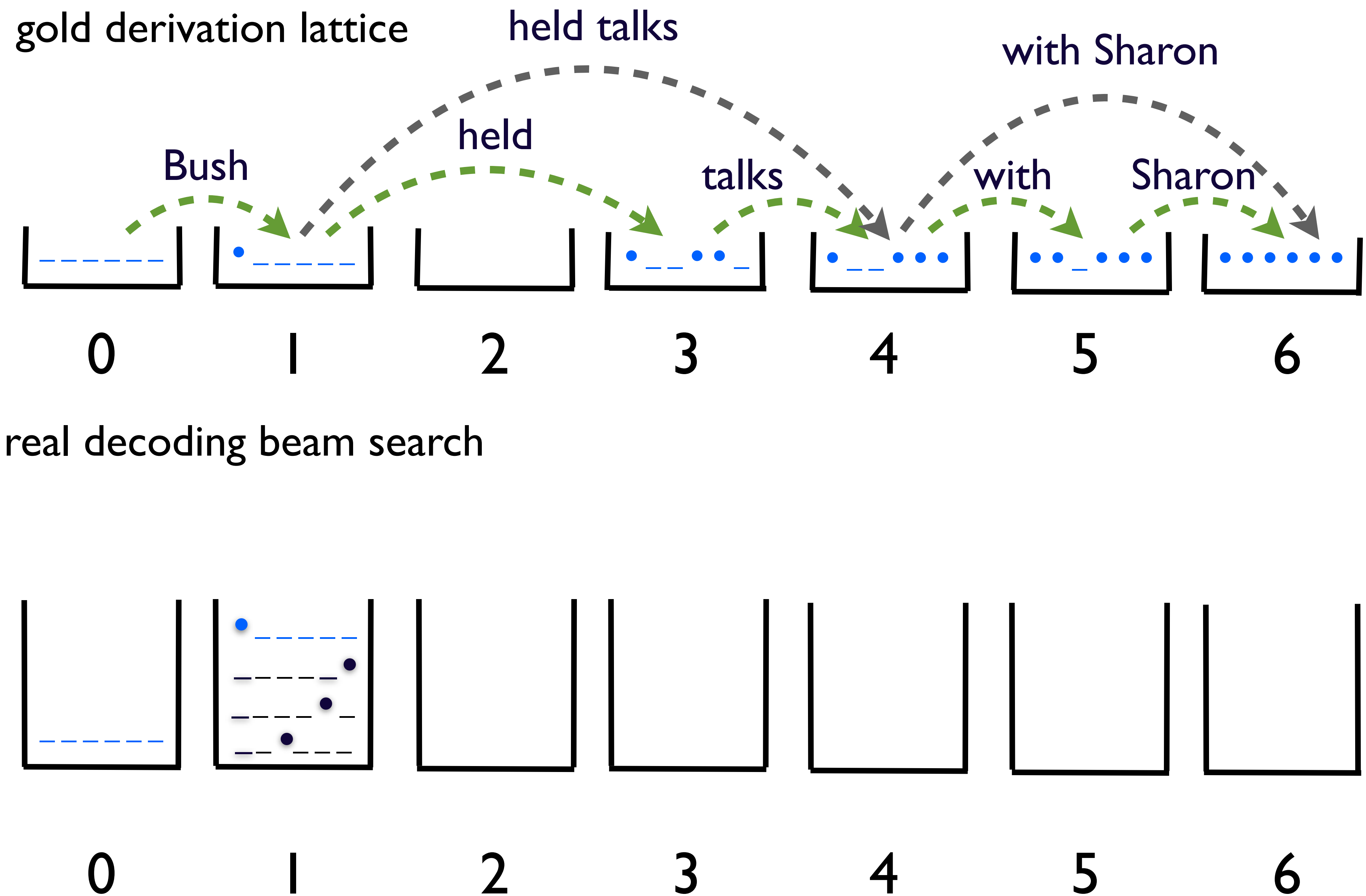


- challenges in applying perceptron for MT
  - the inference (decoding) is vastly inexact (beam search)
  - we know standard perceptron doesn't work for MT
- intuition: the learner should fix the search error first

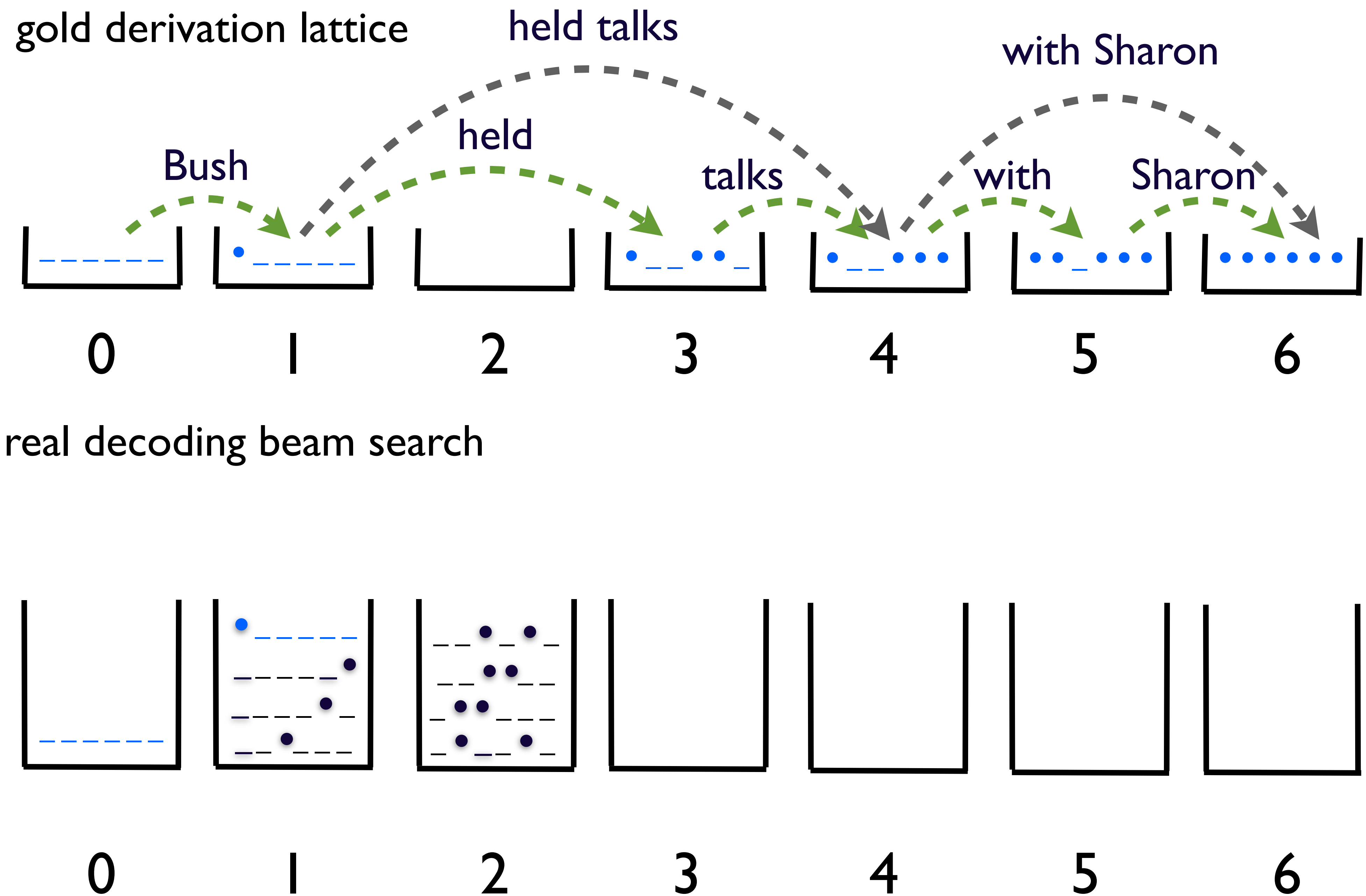
Search Error: Gold Derivations Pruned



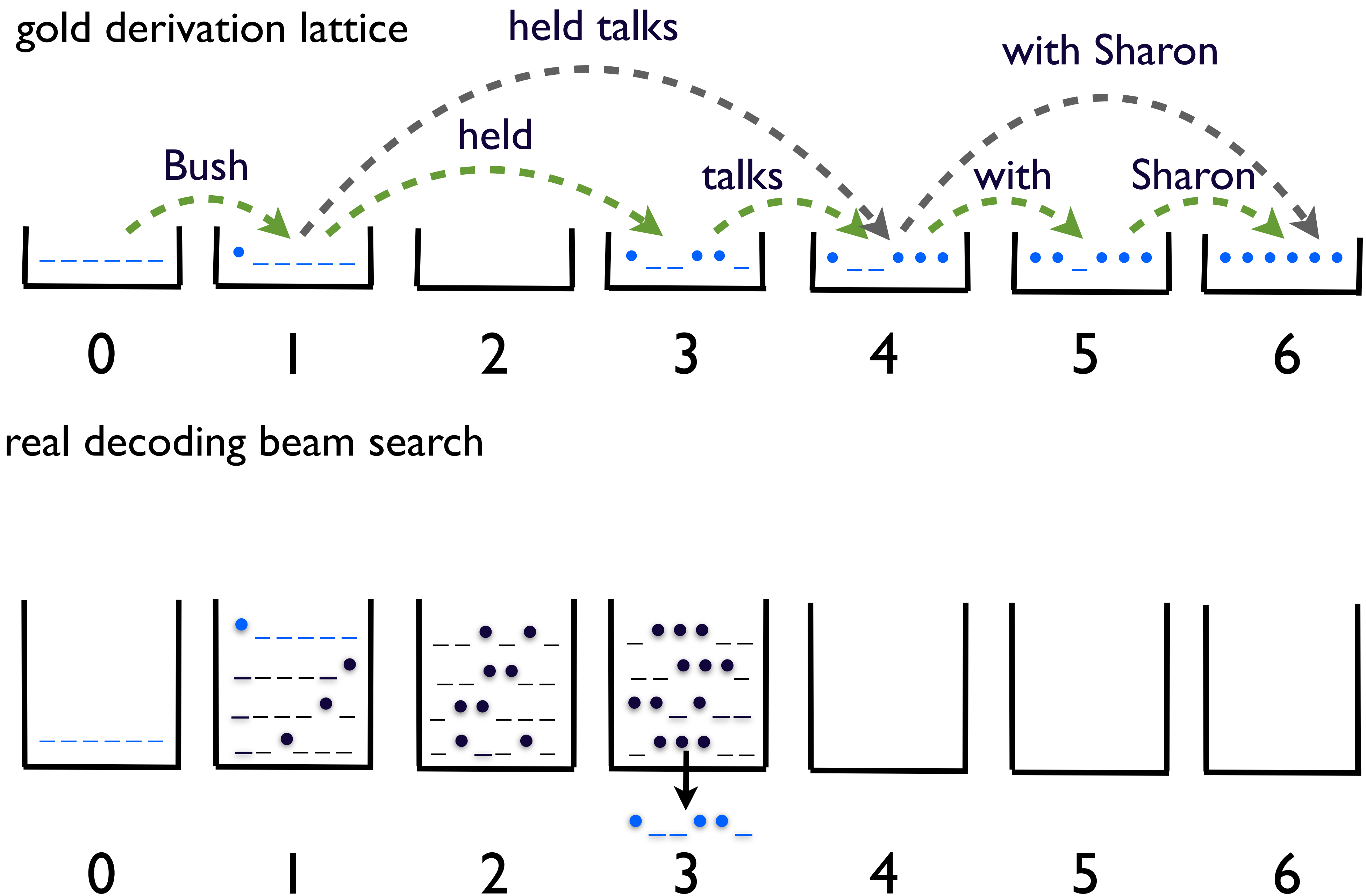
Search Error: Gold Derivations Pruned



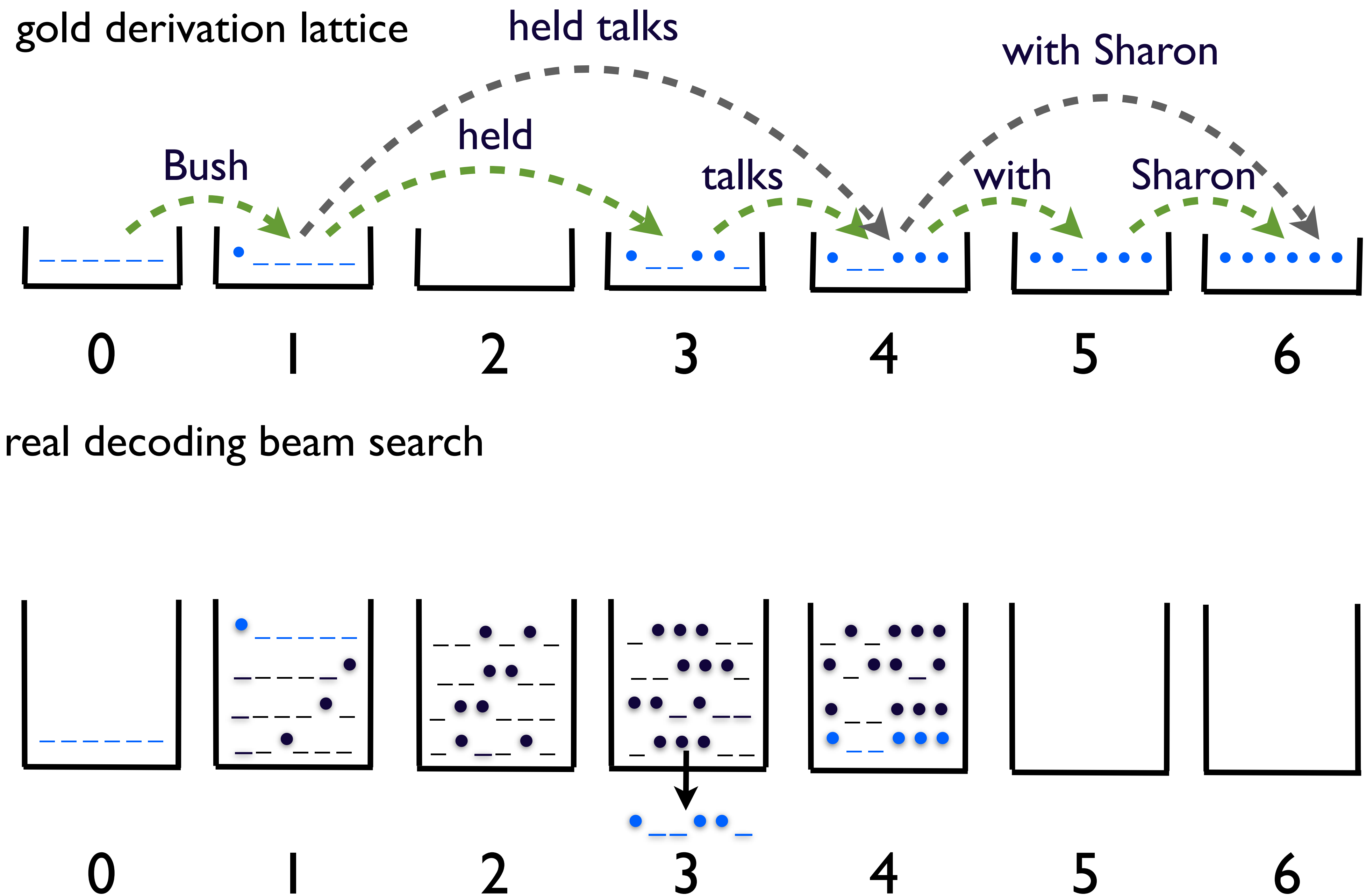
Search Error: Gold Derivations Pruned



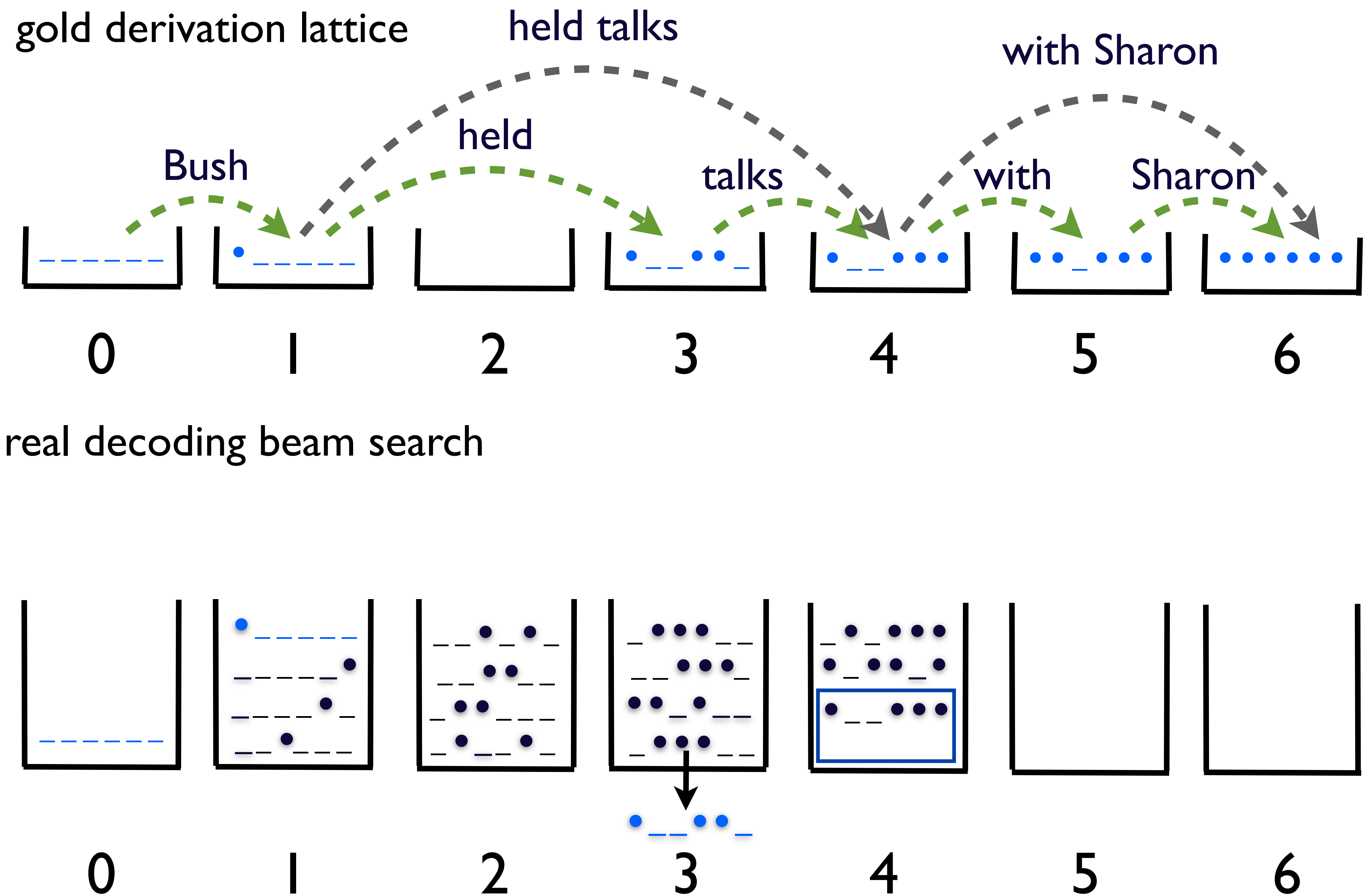
Search Error: Gold Derivations Pruned



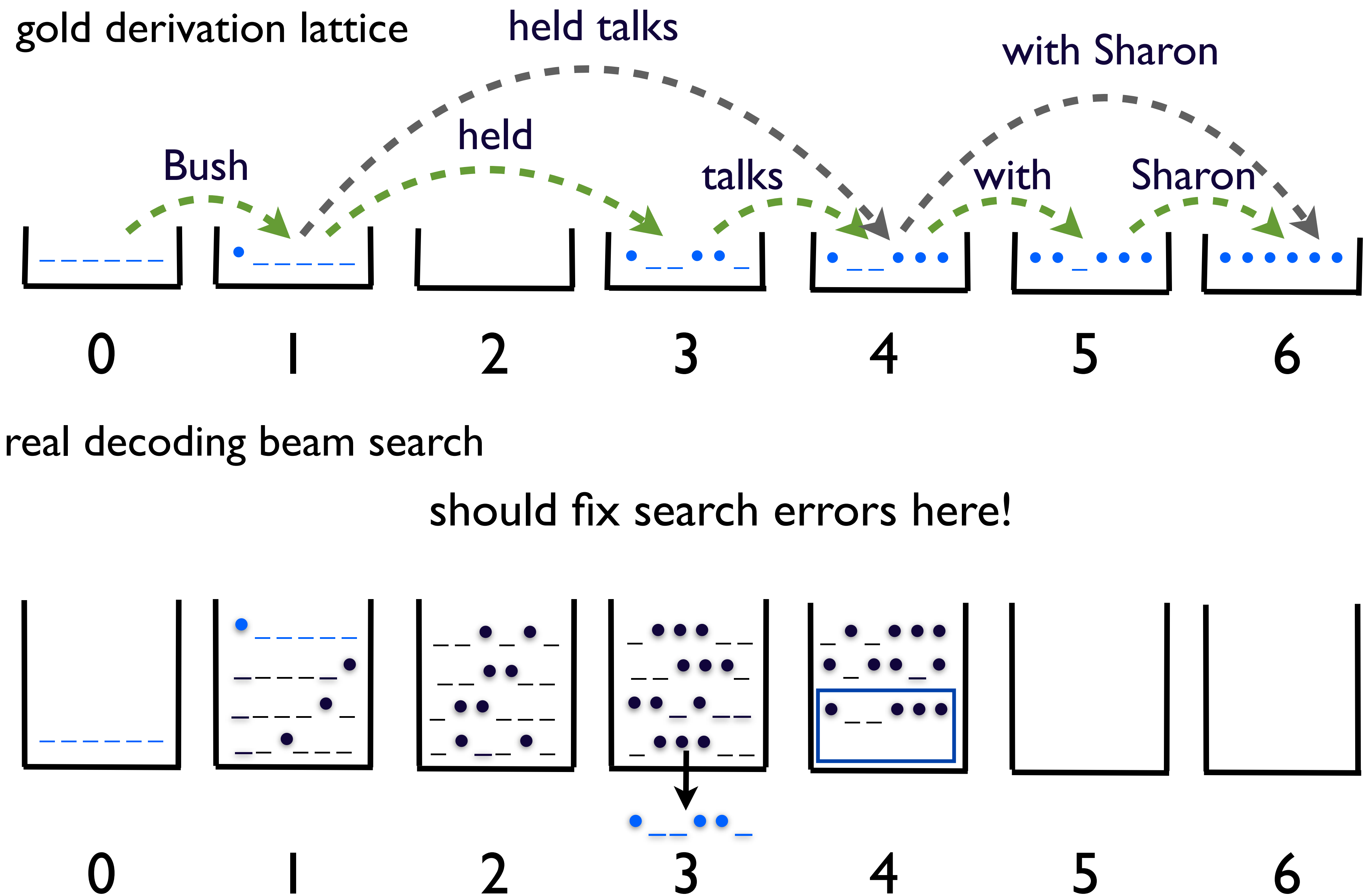
Search Error: Gold Derivations Pruned



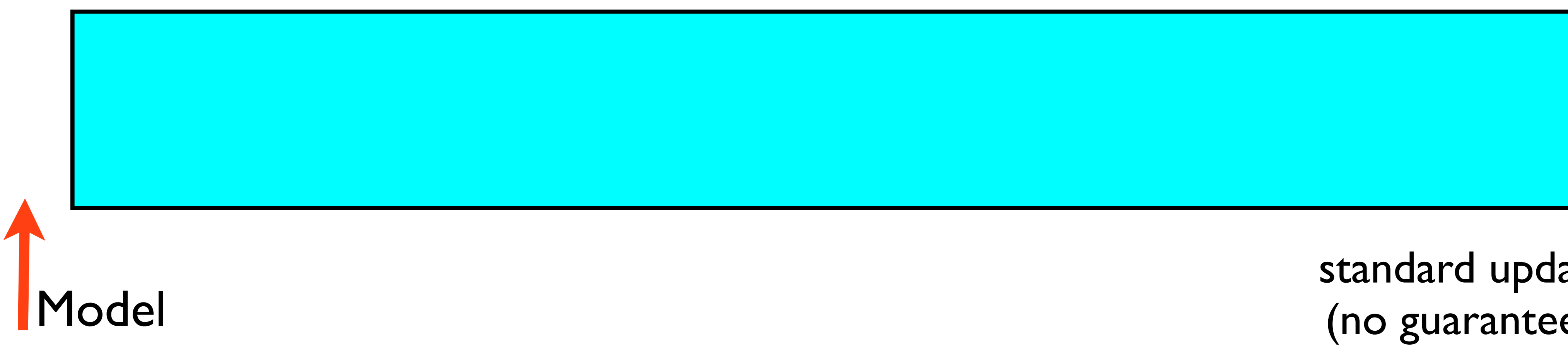
Search Error: Gold Derivations Pruned



Search Error: Gold Derivations Pruned



Fixing Search Error 1: Early Update



standard update
(no guarantee!)

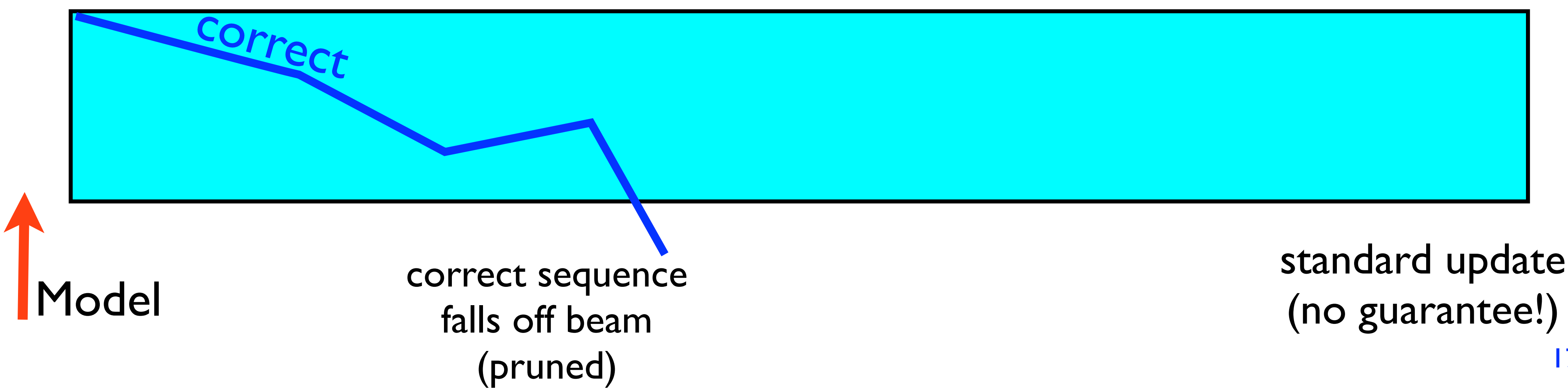
Fixing Search Error 1: Early Update

- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" which we want to fix

standard update
(no guarantee!)

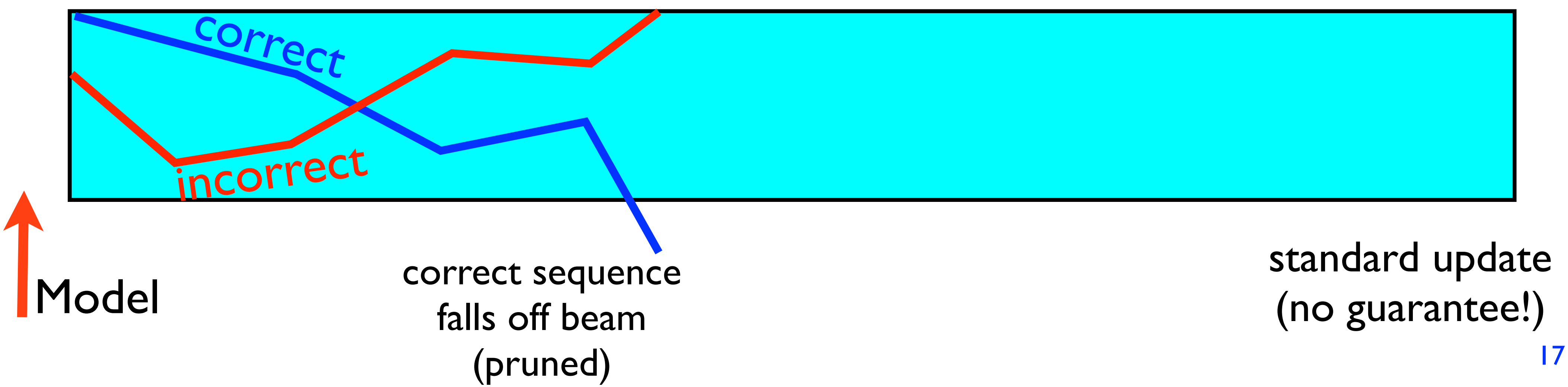
Fixing Search Error 1: Early Update

- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" which we want to fix



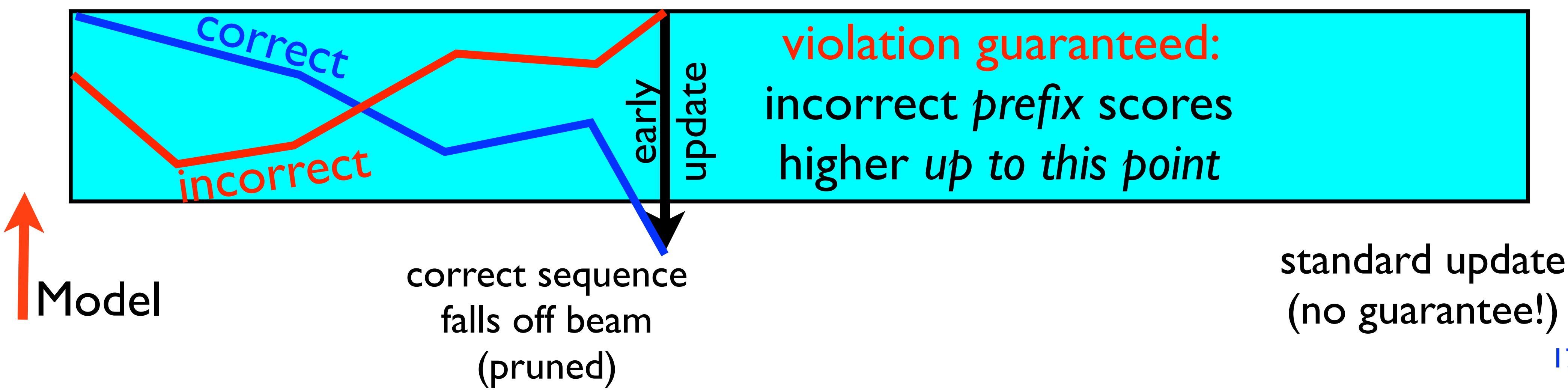
Fixing Search Error 1: Early Update

- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" which we want to fix



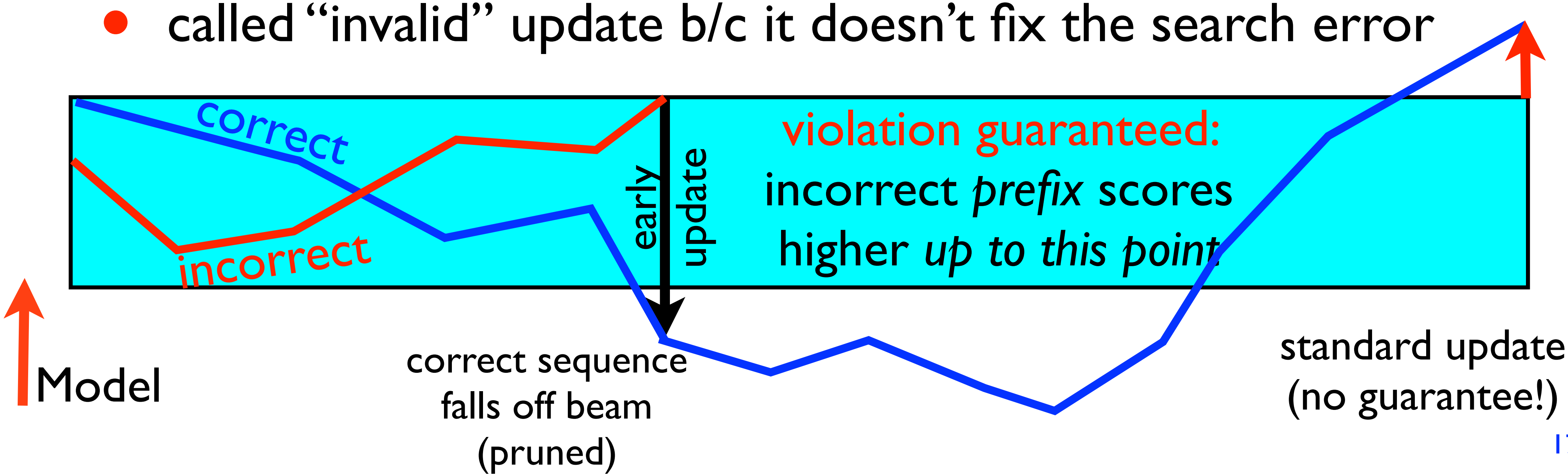
Fixing Search Error 1: Early Update

- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" which we want to fix



Fixing Search Error 1: Early Update

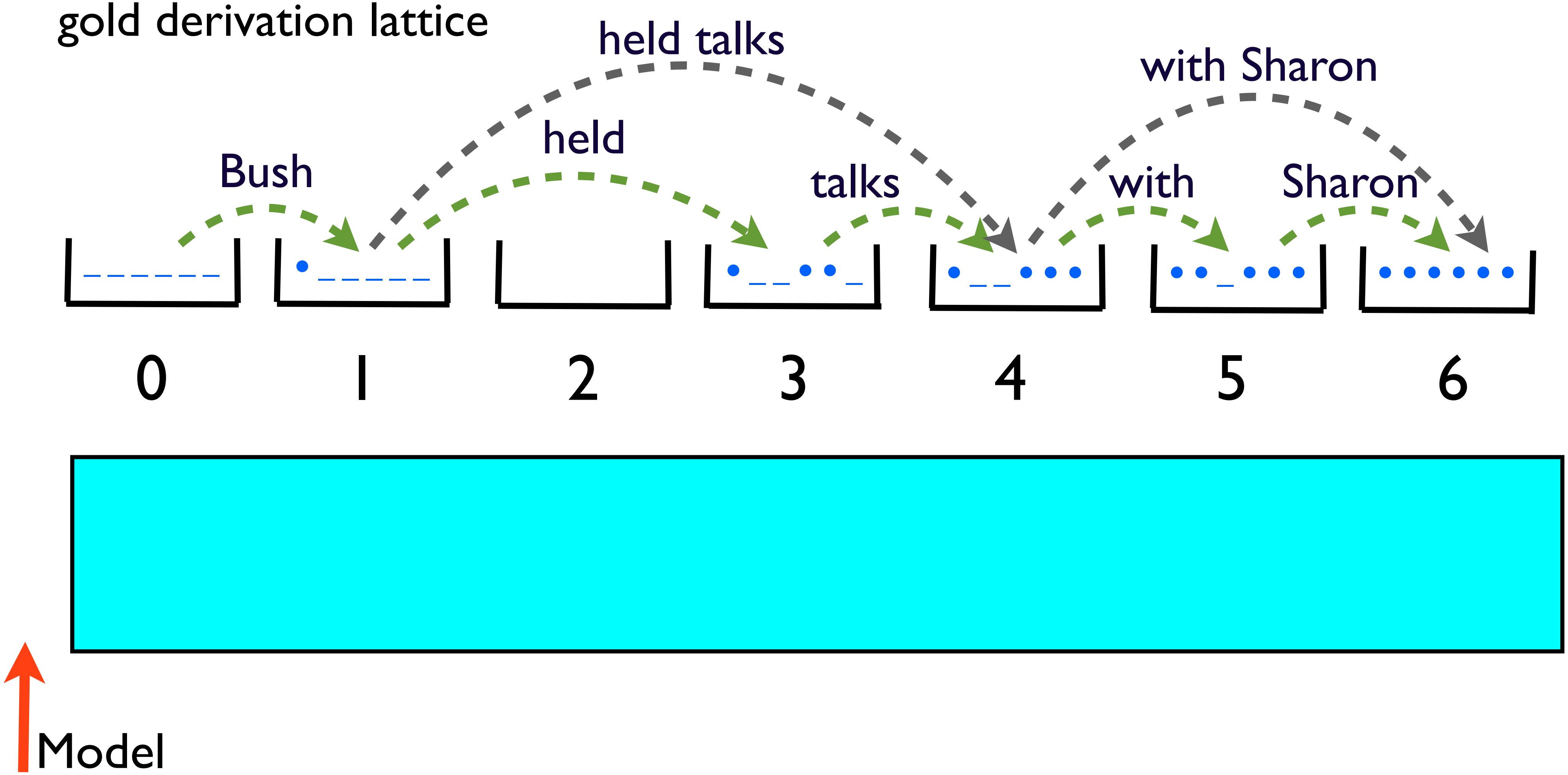
- early update (Collins/Roark'04) when the correct falls off beam
  - up to this point the incorrect prefix should score higher
  - that's a "violation" which we want to fix
- standard perceptron does not guarantee violation
  - w/ pruning, the correct seq. might score higher at the end!
  - called "invalid" update b/c it doesn't fix the search error



Early Update w/ Latent Variable

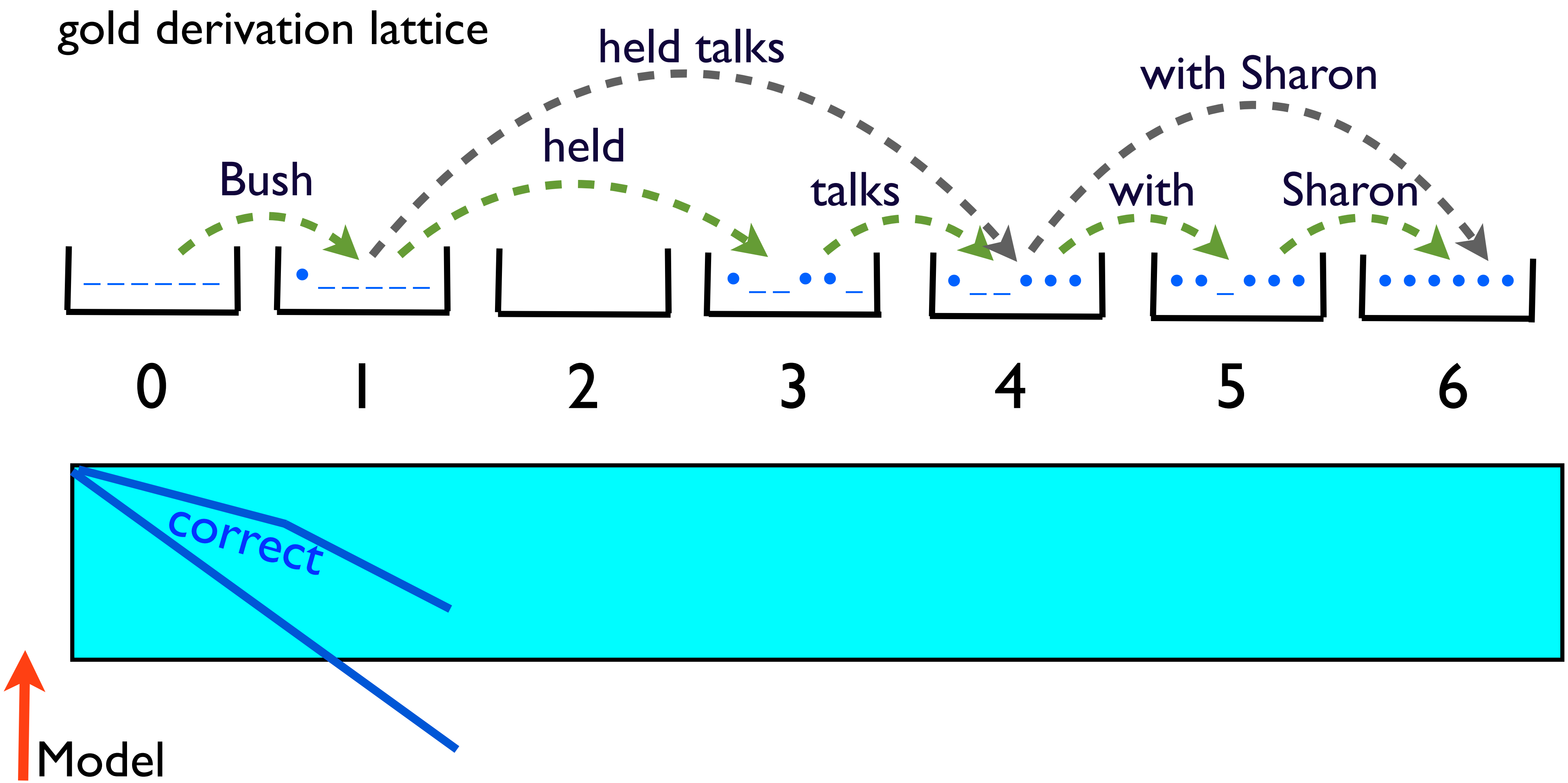
- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good

gold derivation lattice



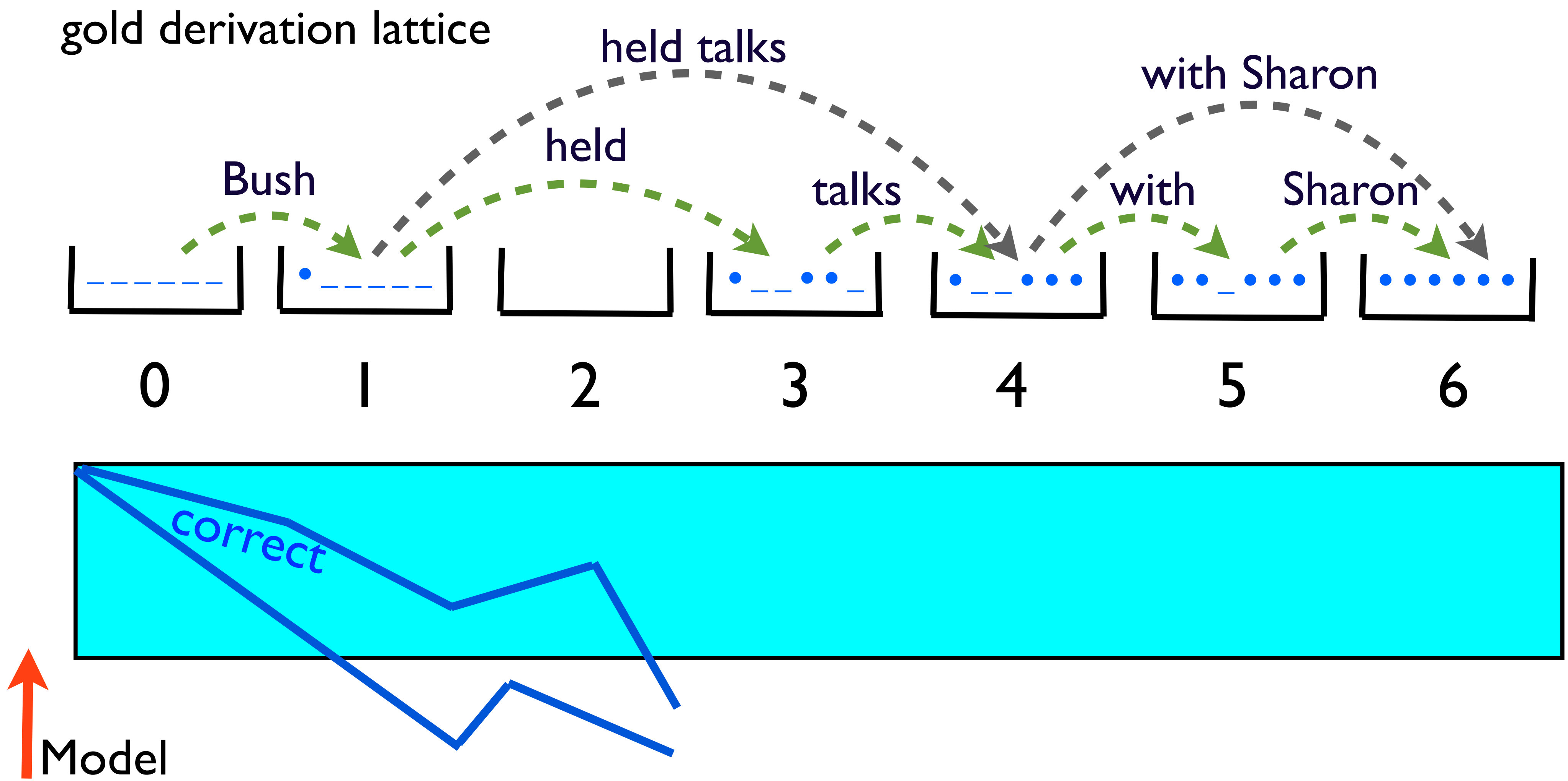
Early Update w/ Latent Variable

- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good



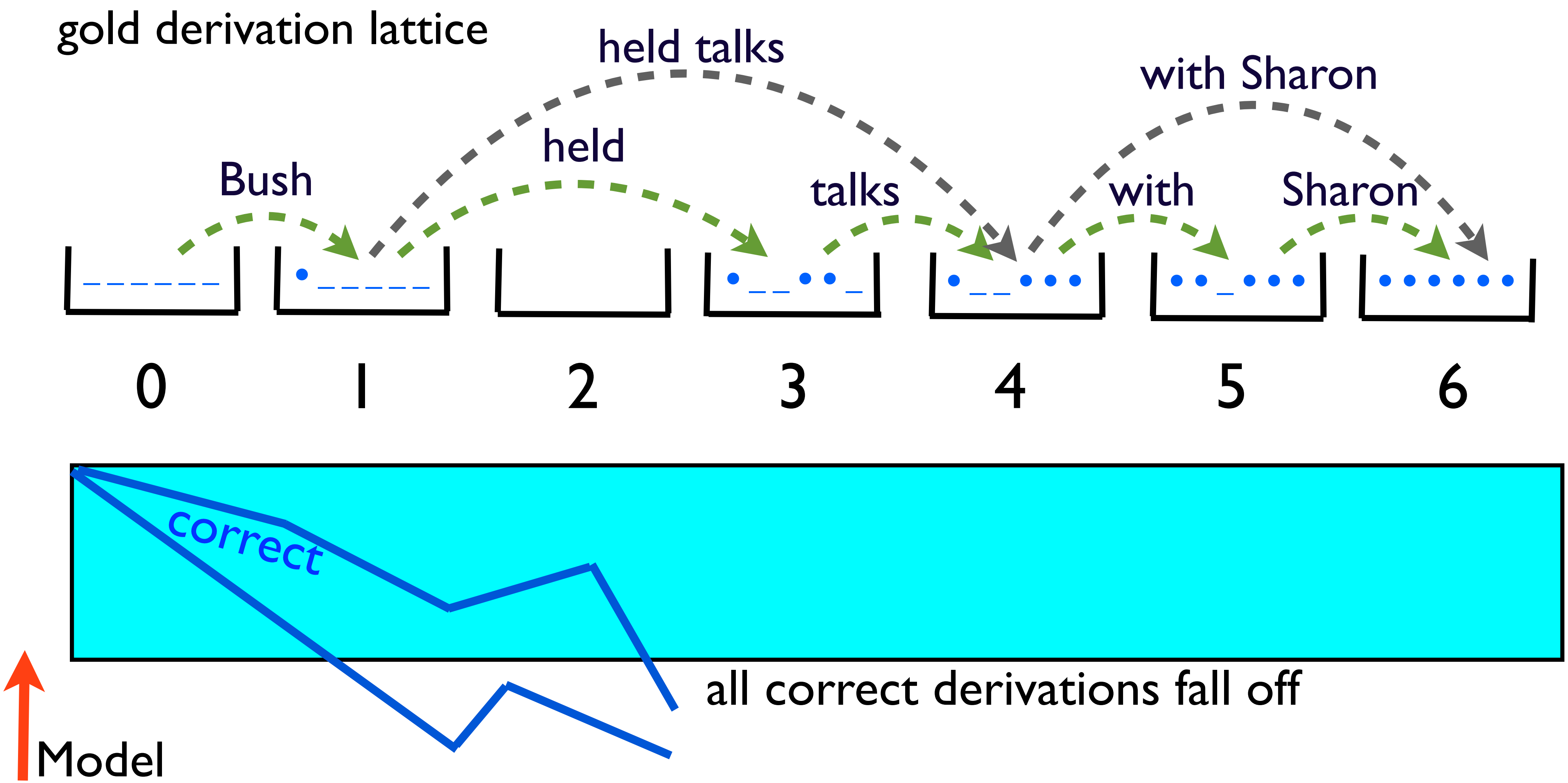
Early Update w/ Latent Variable

- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good



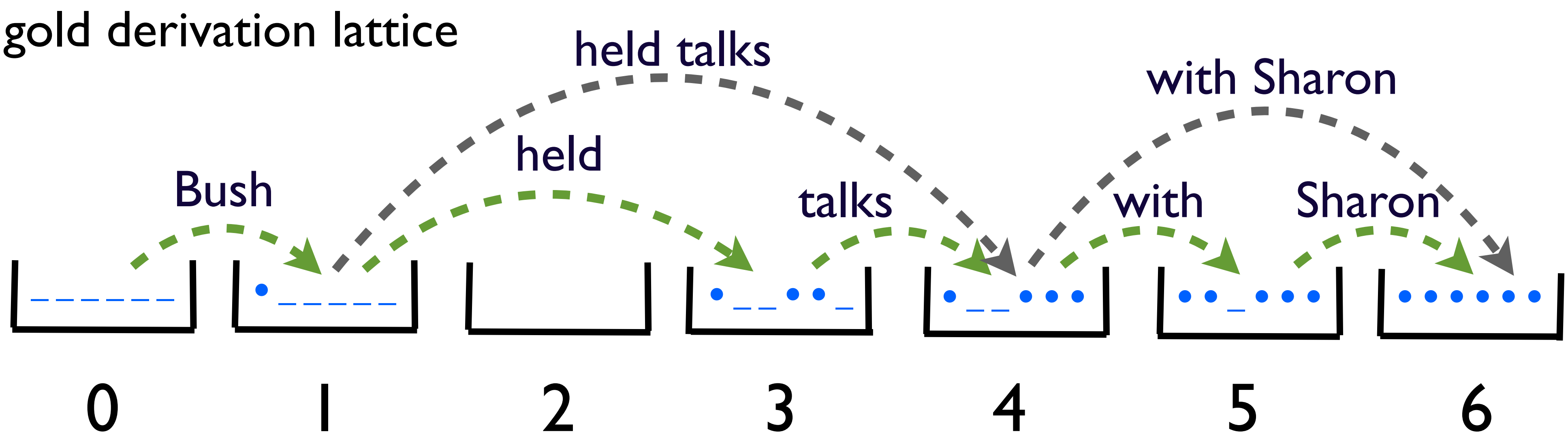
Early Update w/ Latent Variable

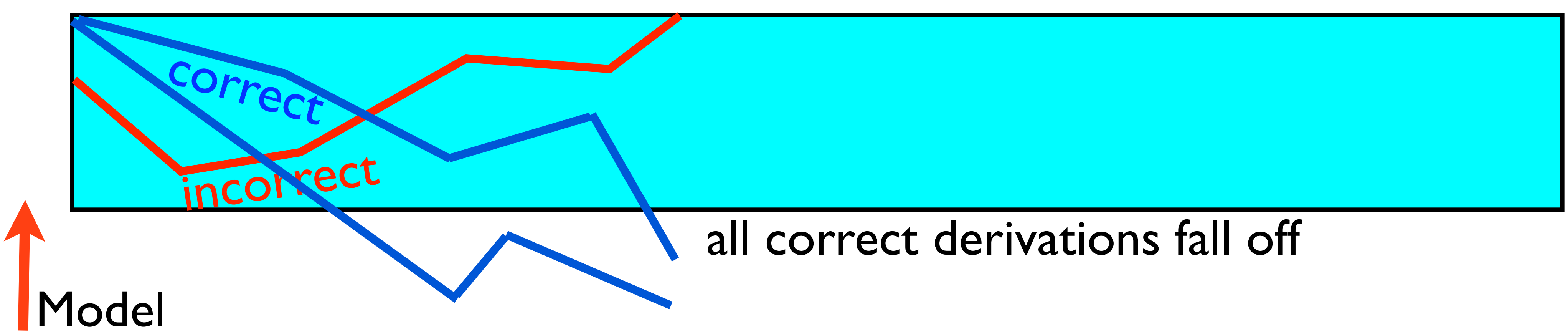
- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good



Early Update w/ Latent Variable

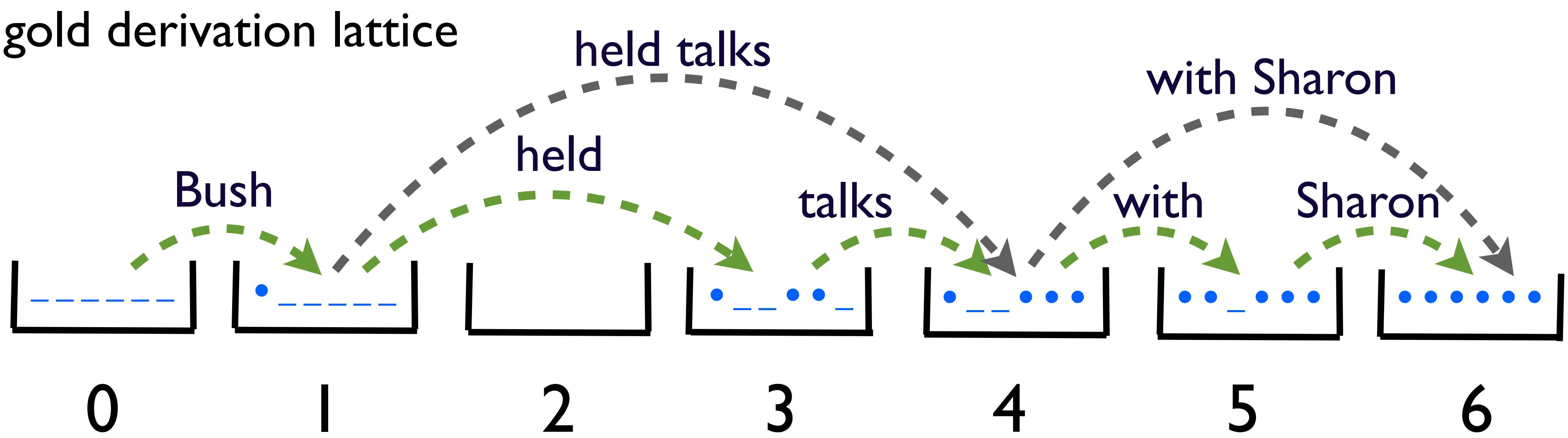
- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good

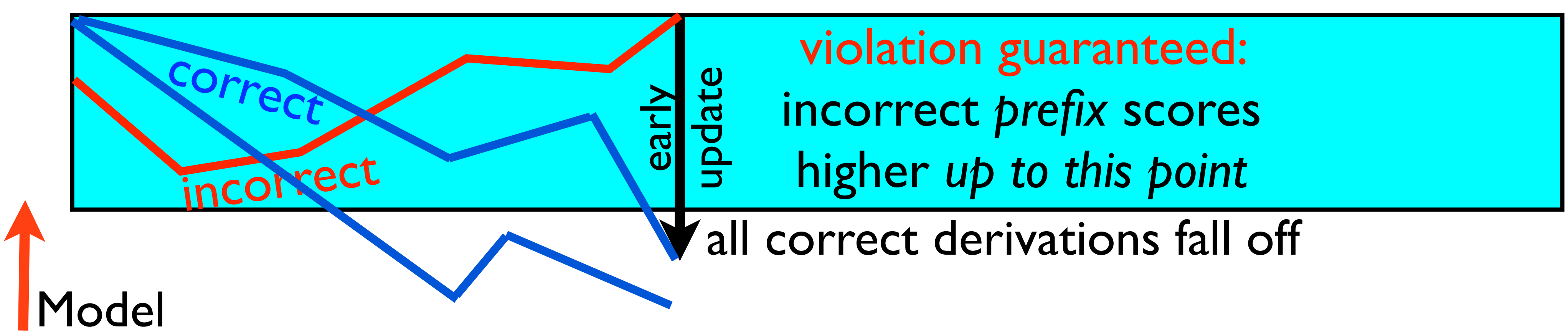




Early Update w/ Latent Variable

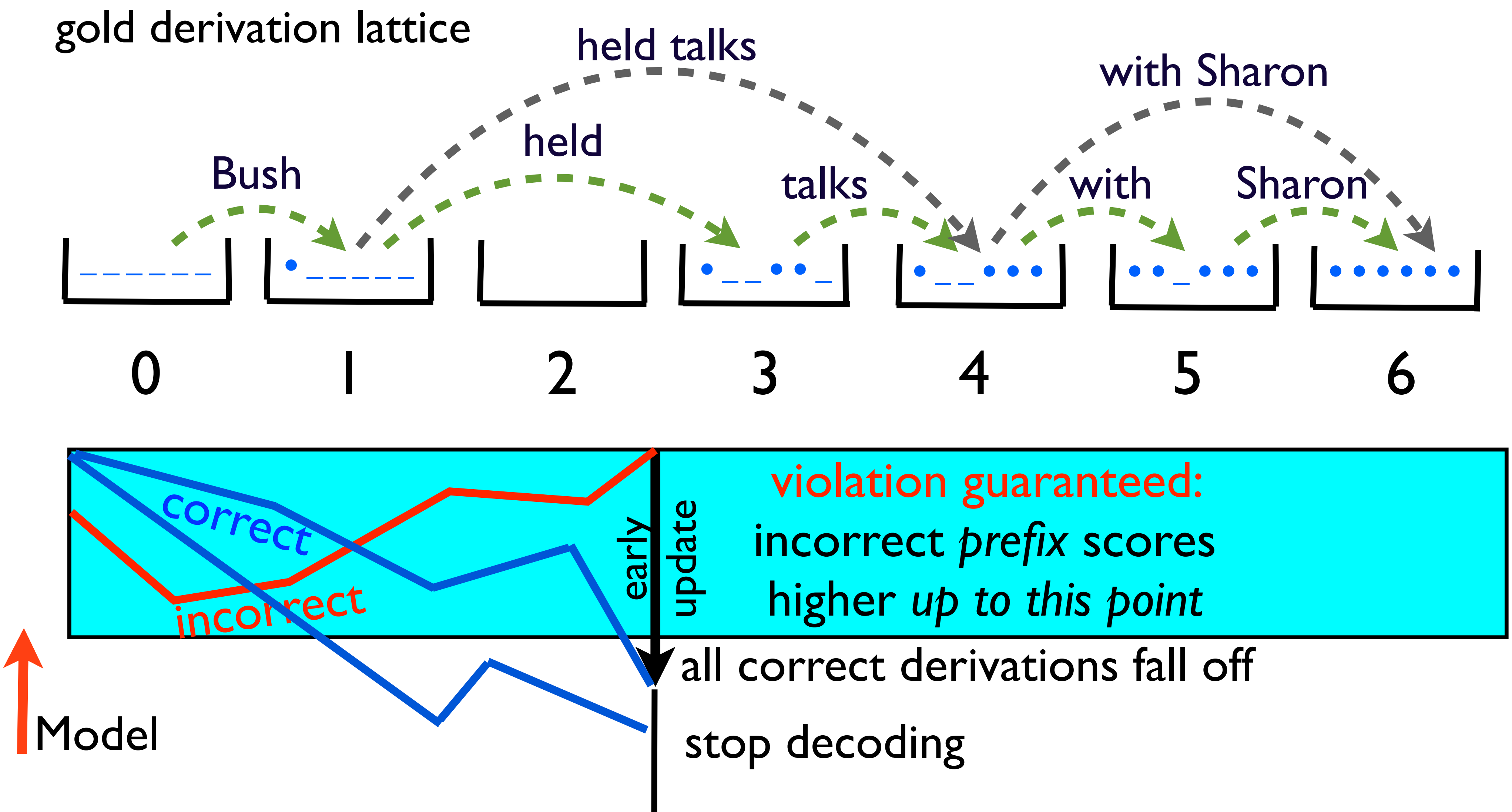
- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good



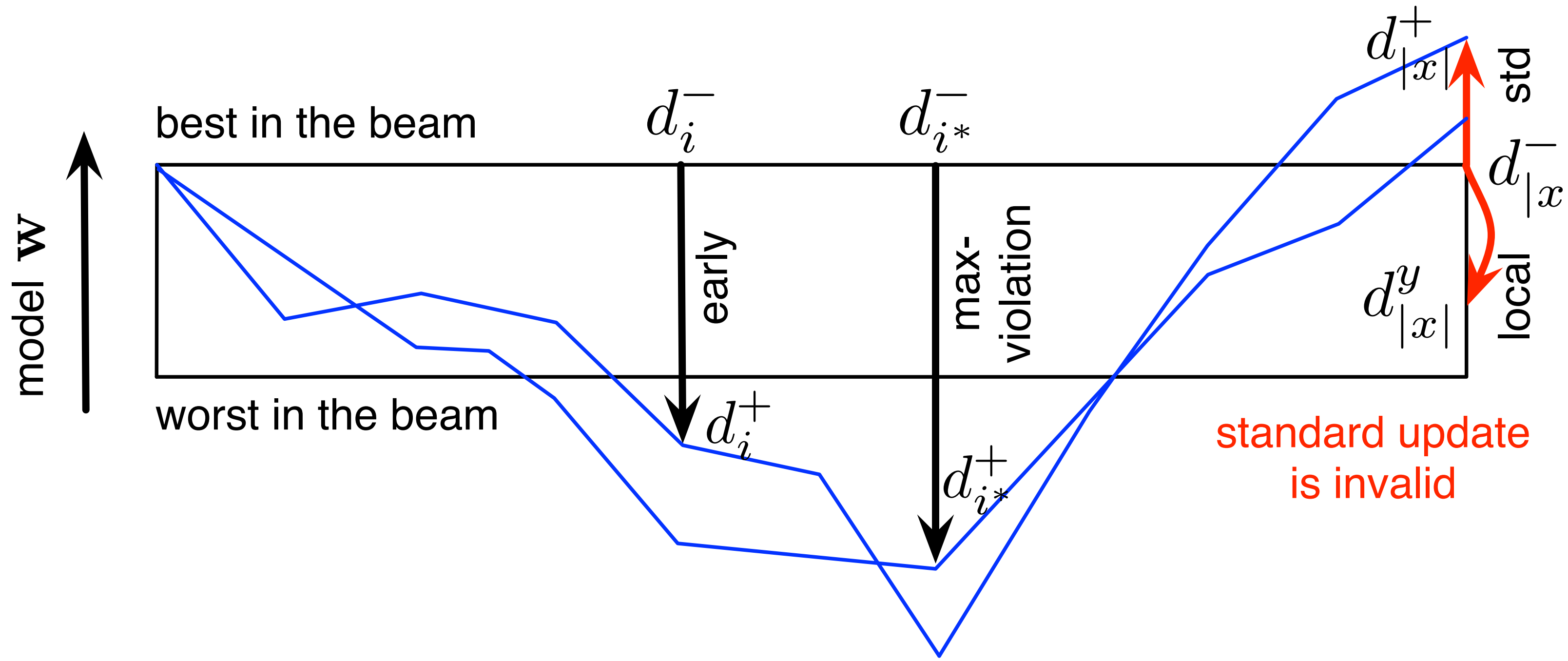


Early Update w/ Latent Variable

- the gold-standard derivations are ***not*** annotated
  - we treat any reference-producing derivation as good

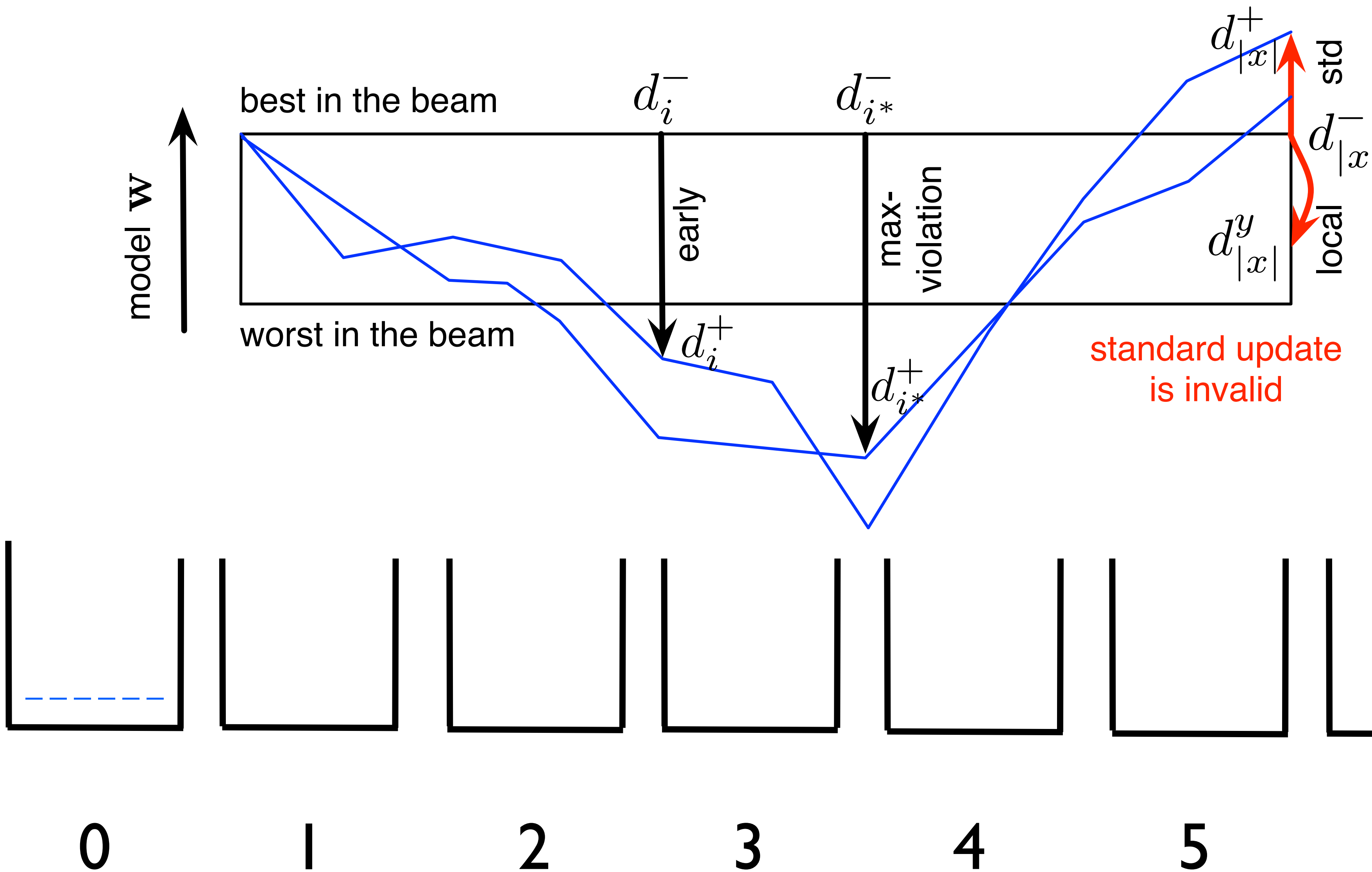


Fixing Search Error 2: Max-Violation

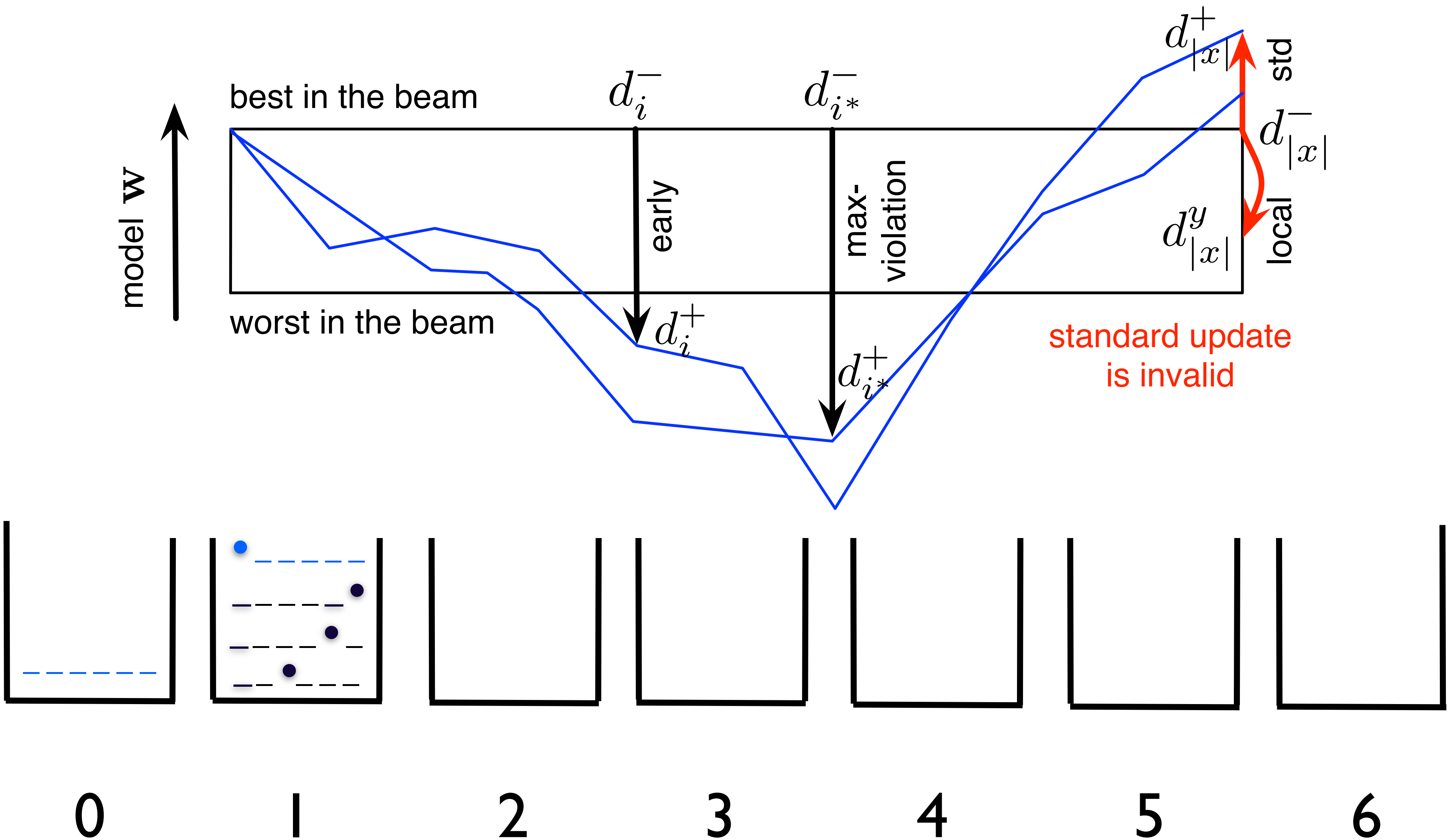


- early update works but learns slowly due to partial updates
- max-violation: use the prefix where violation is maximum
  - "worst-mistake" in the search space
- we call these methods "violation-fixing perceptrons" (Huang et al 2012)

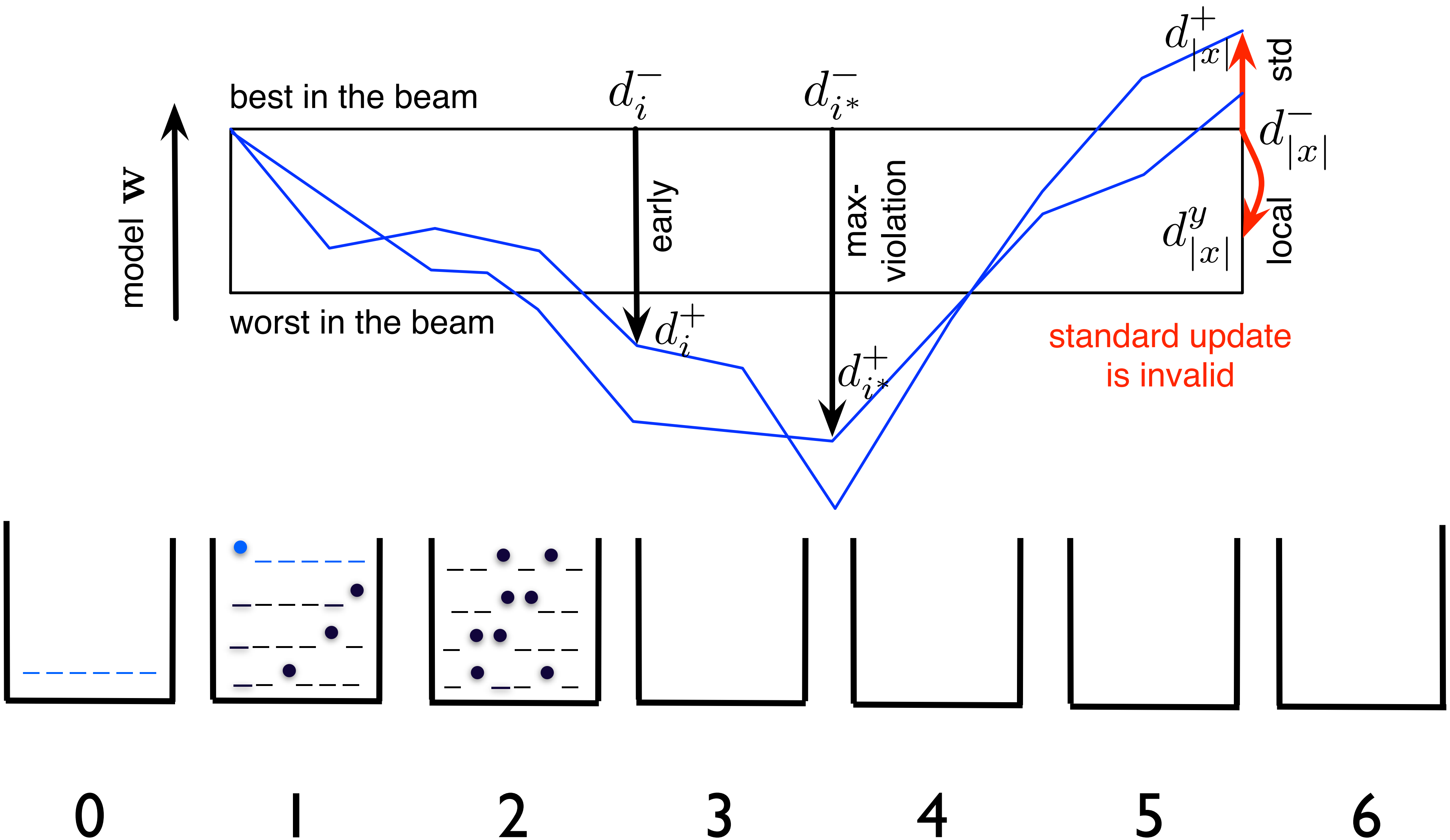
Early Update vs. Max-Violation



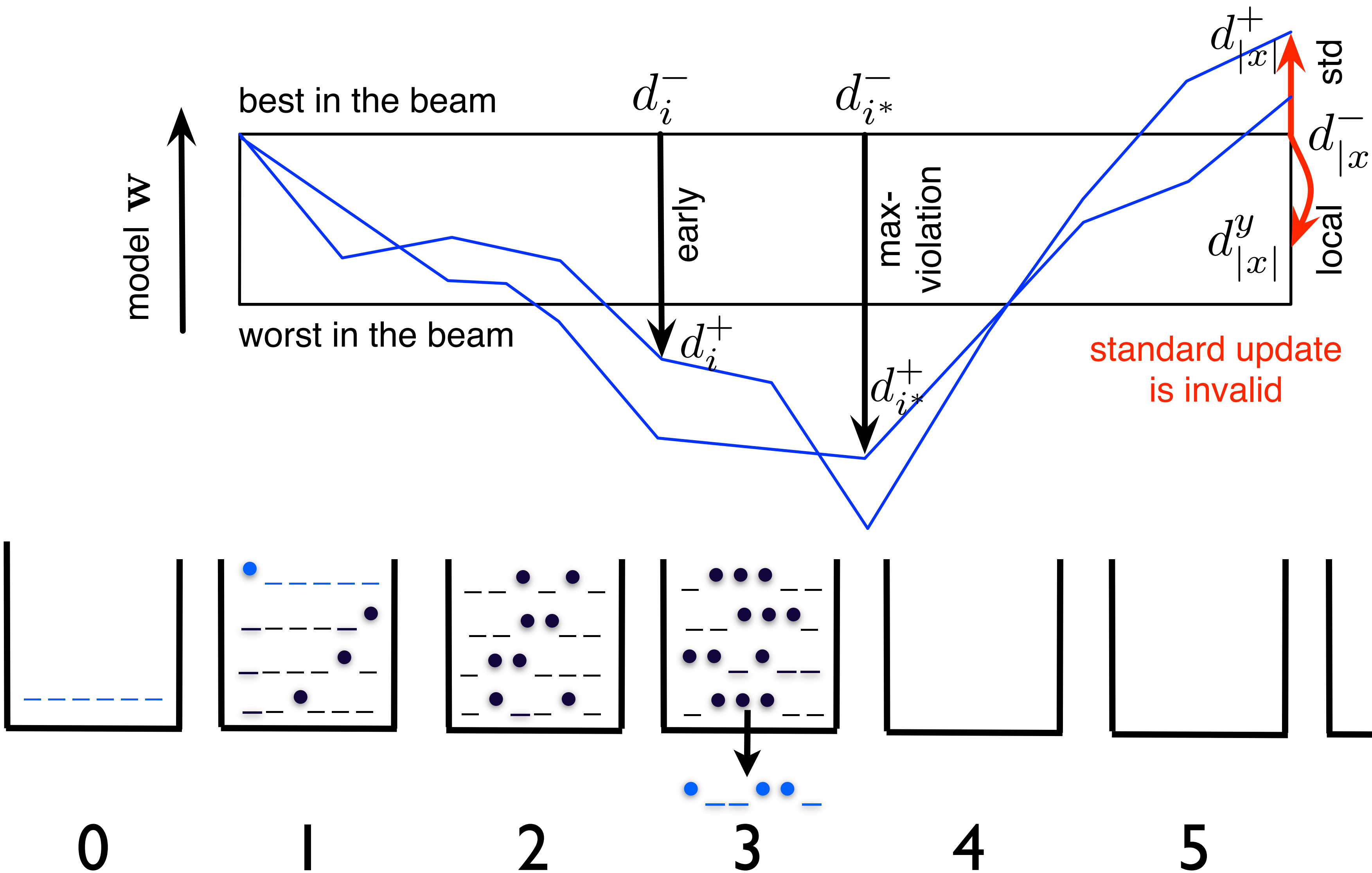
Early Update vs. Max-Violation



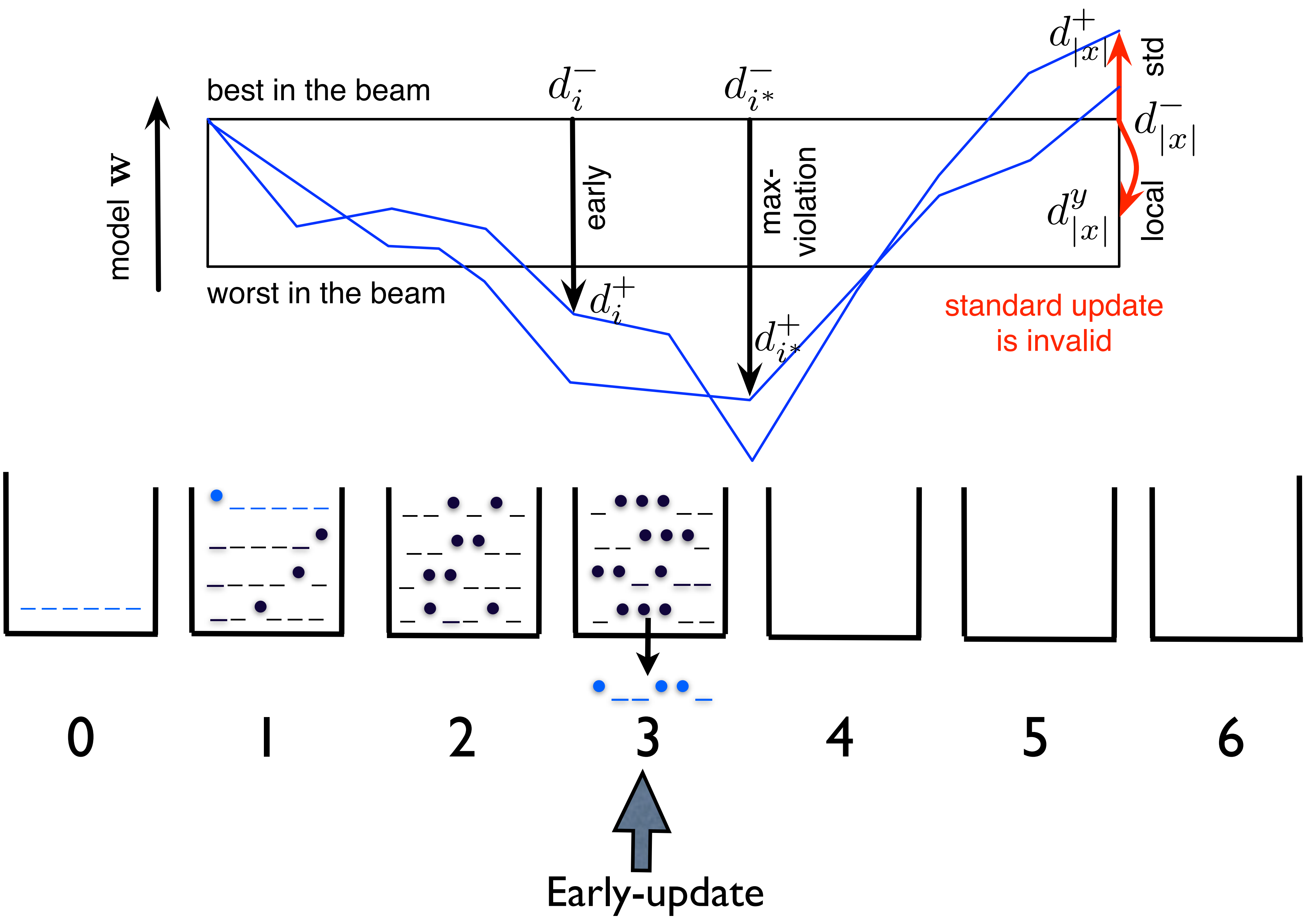
Early Update vs. Max-Violation



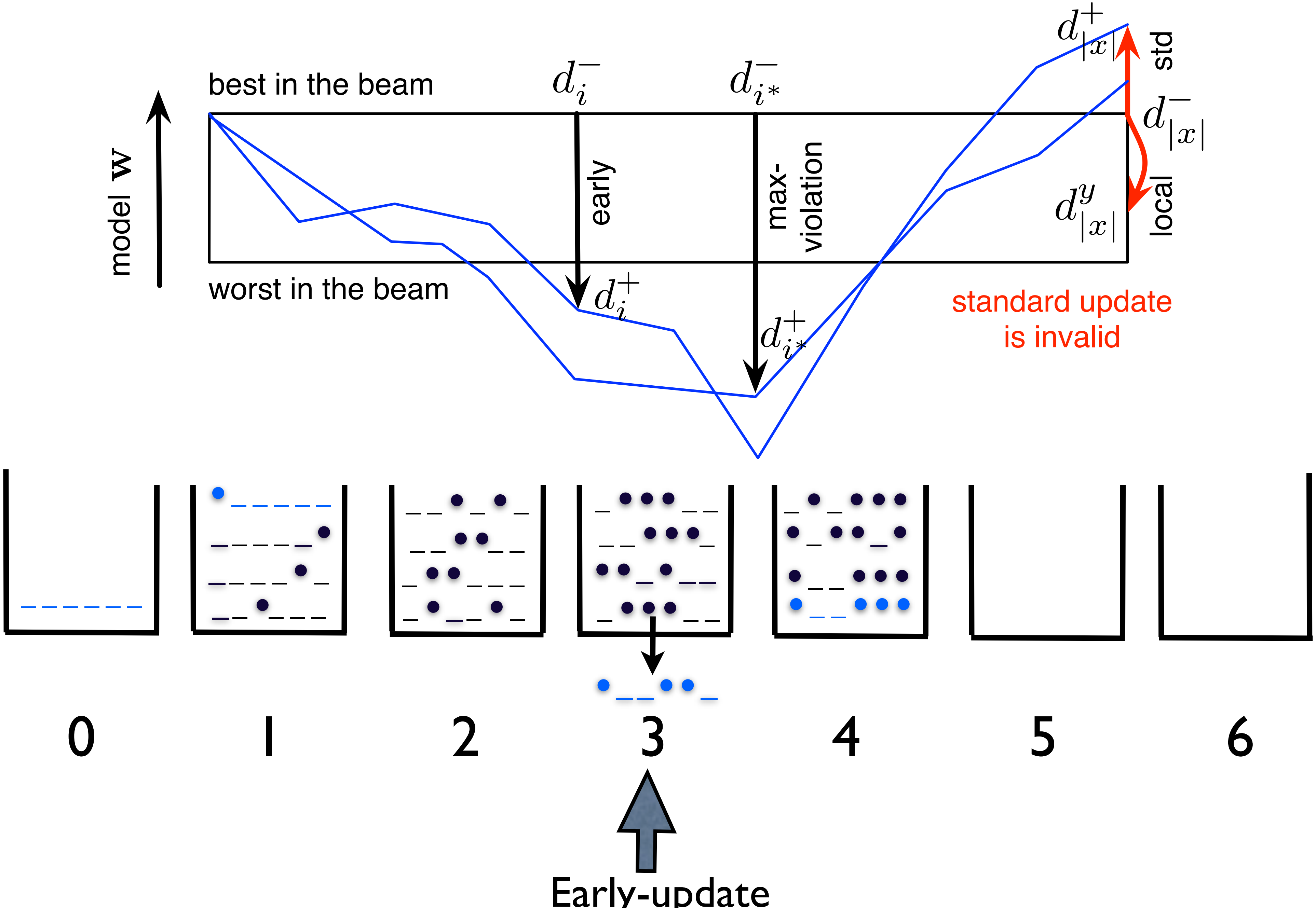
Early Update vs. Max-Violation



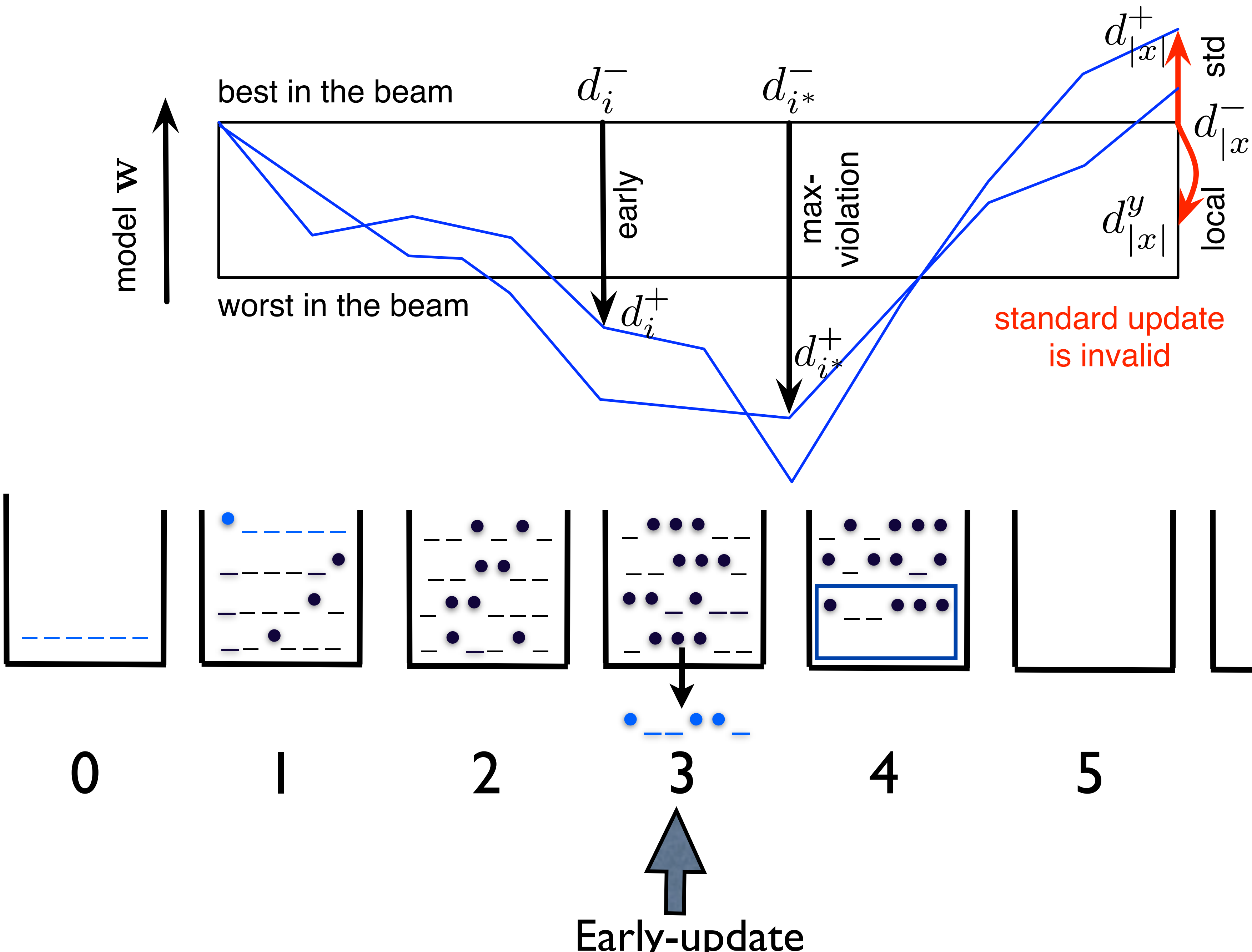
Early Update vs. Max-Violation



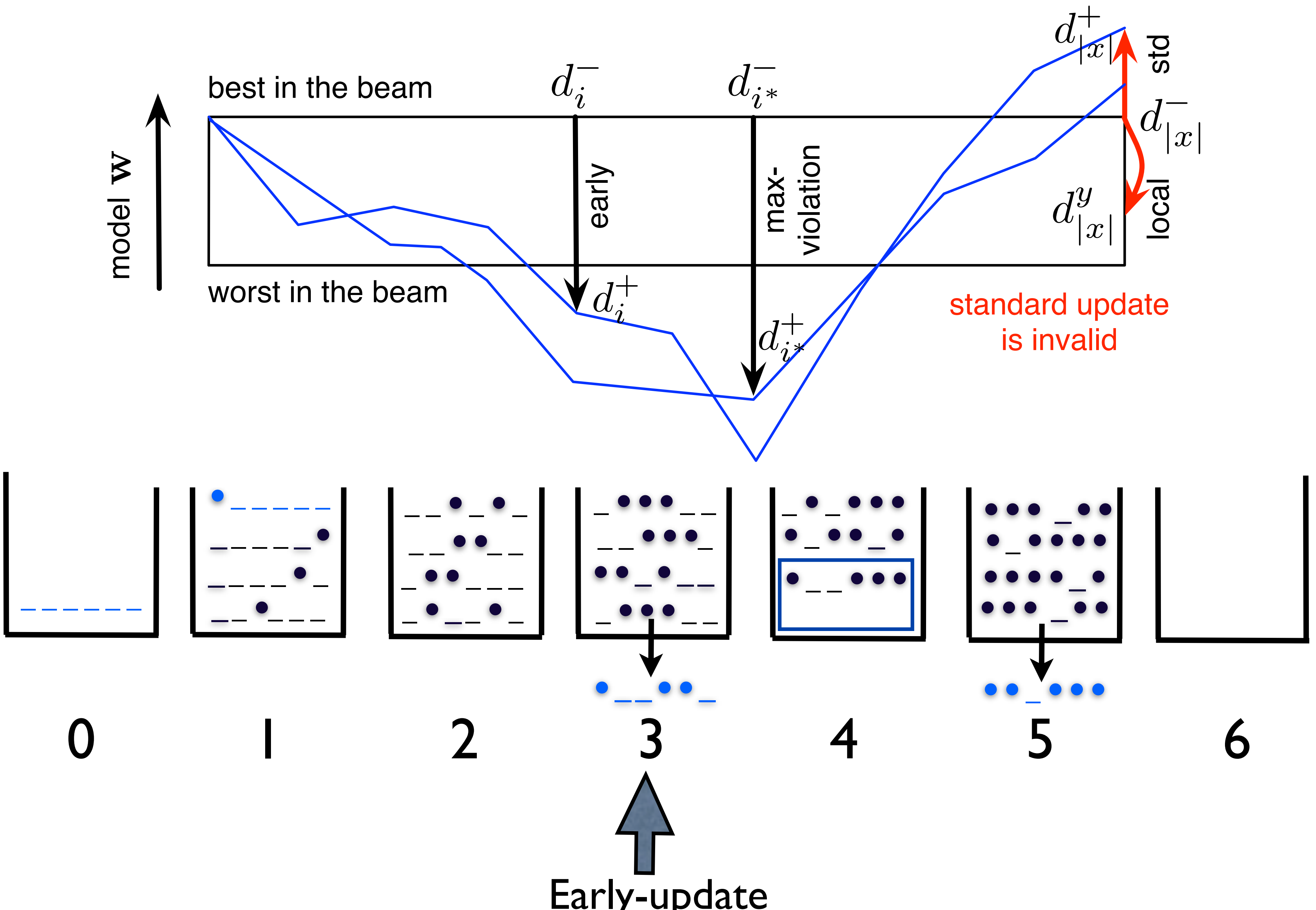
Early Update vs. Max-Violation



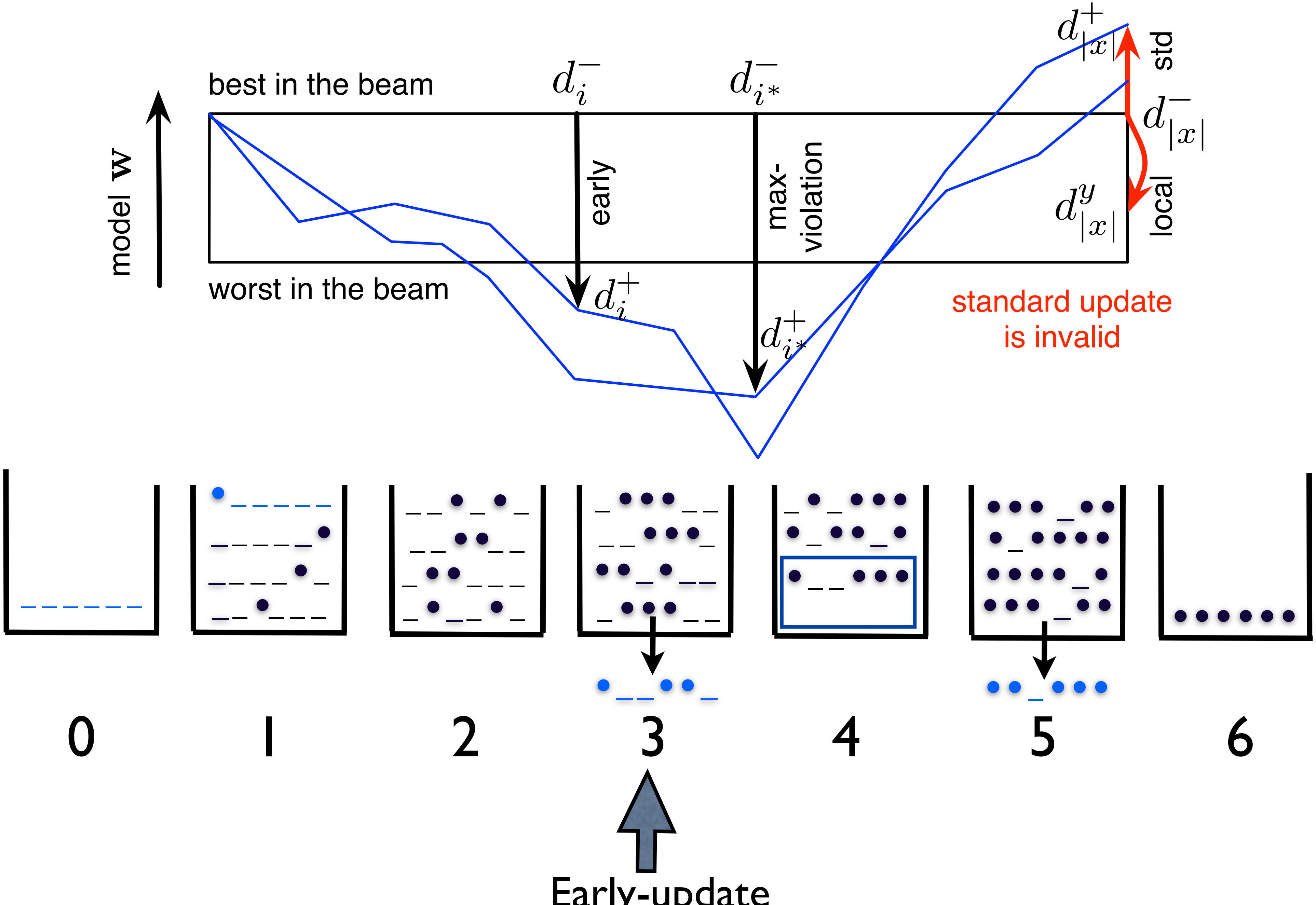
Early Update vs. Max-Violation



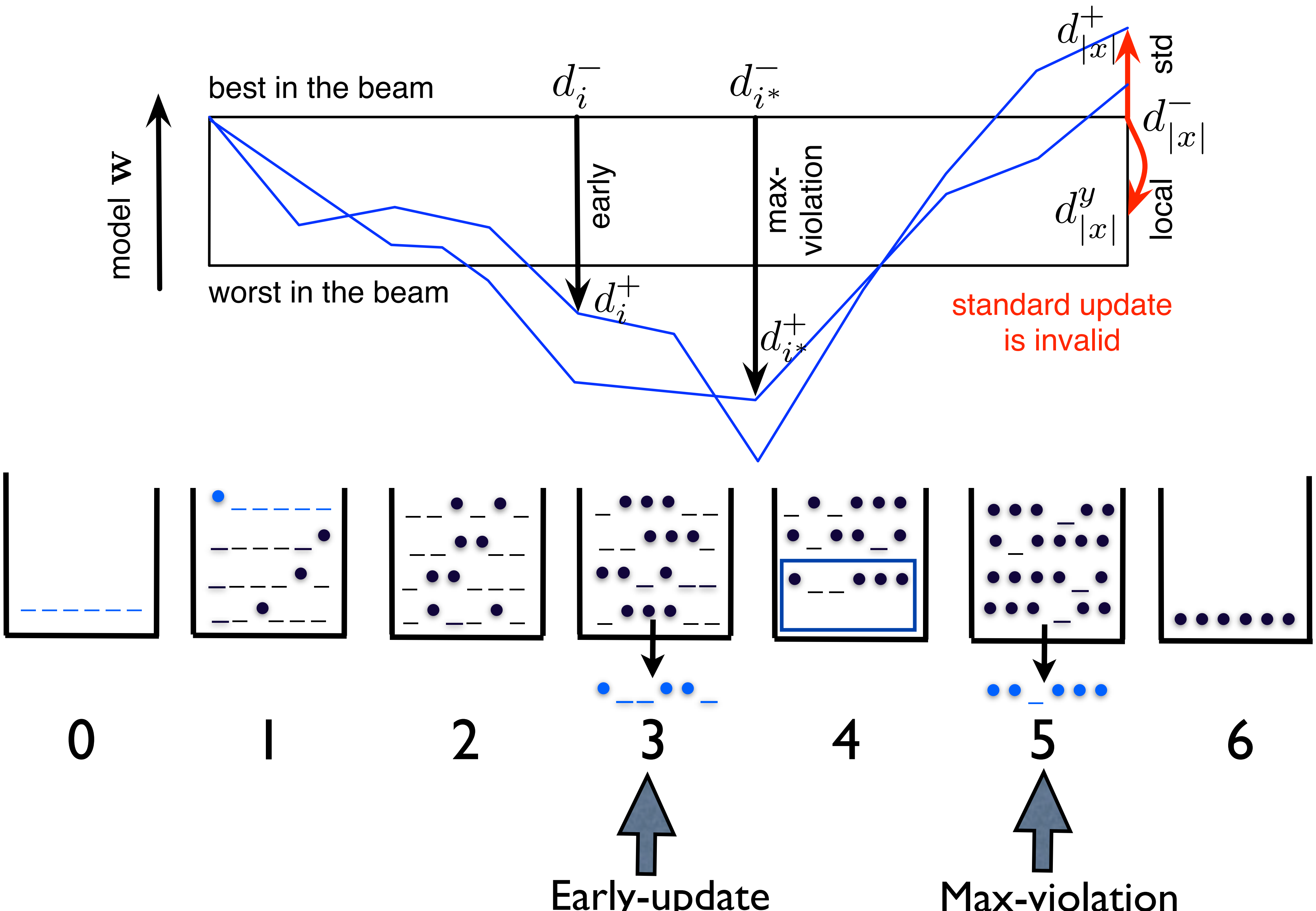
Early Update vs. Max-Violation



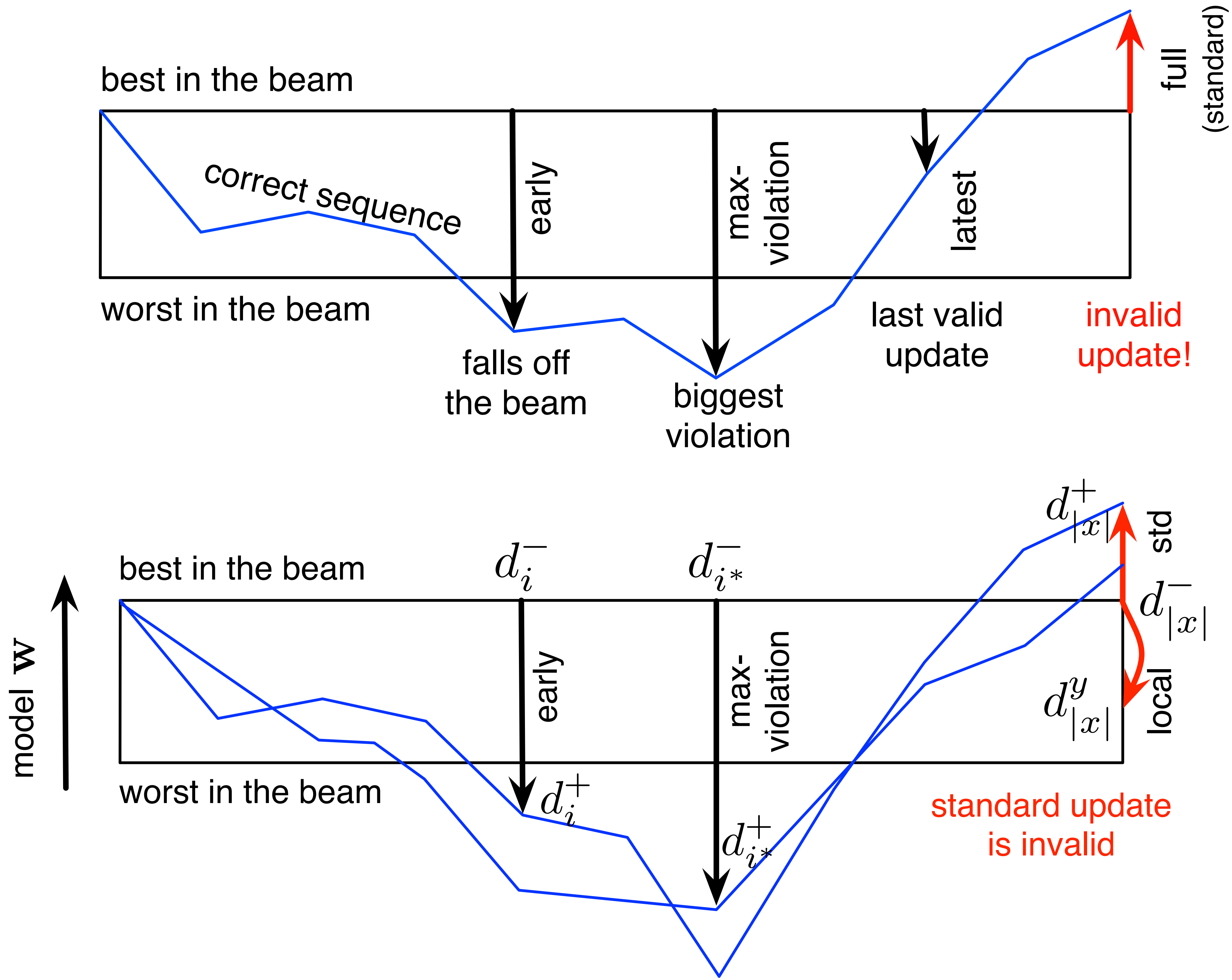
Early Update vs. Max-Violation



Early Update vs. Max-Violation



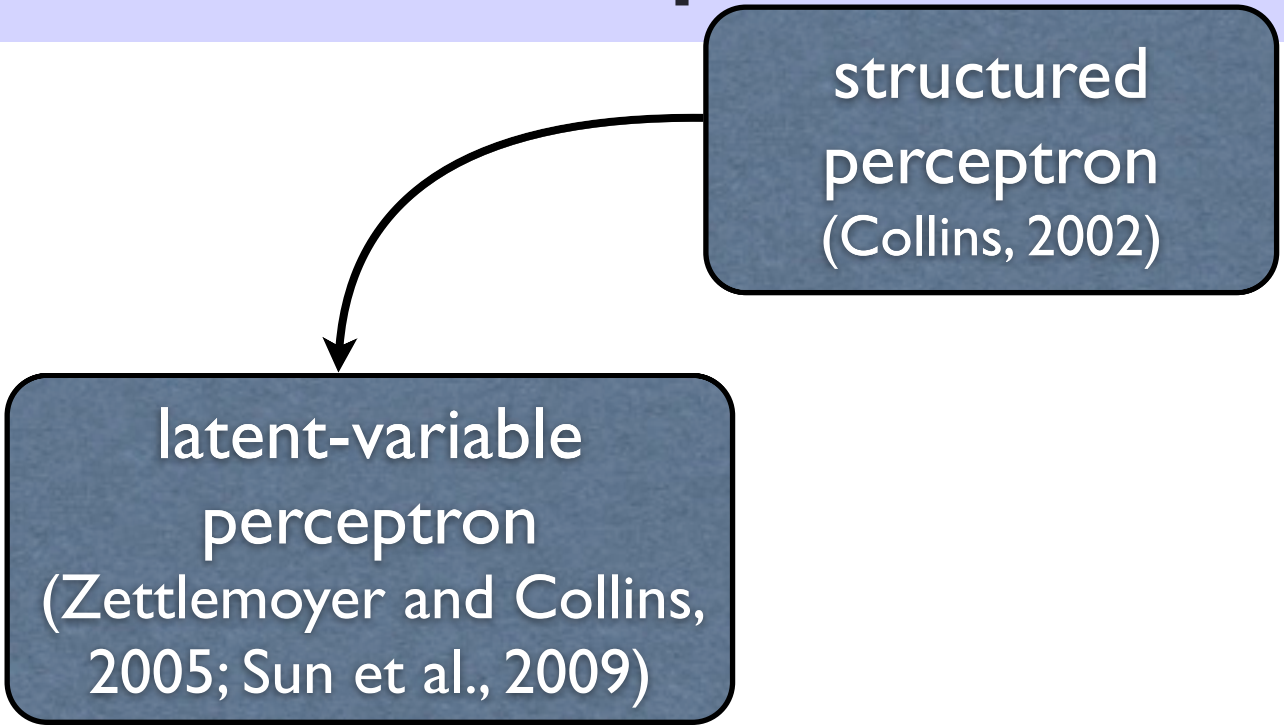
Latent-Variable Perceptron



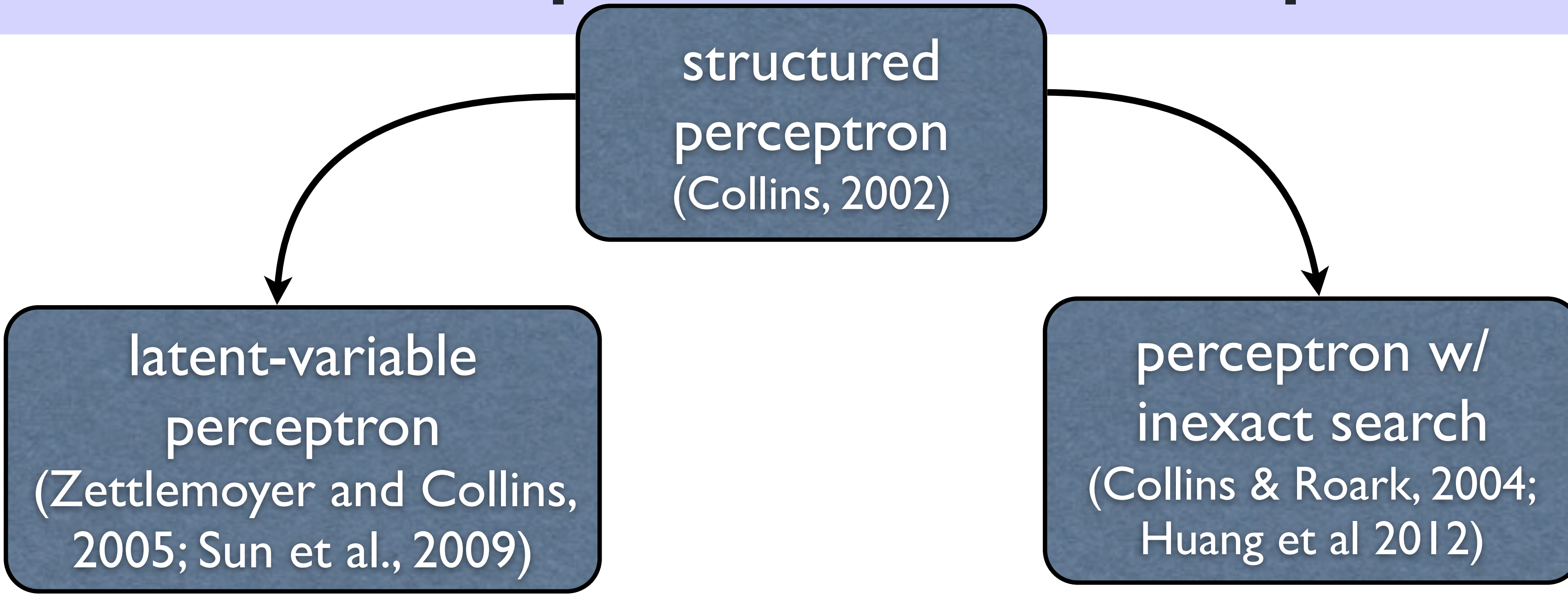
Roadmap of the techniques

structured perceptron
(Collins, 2002)

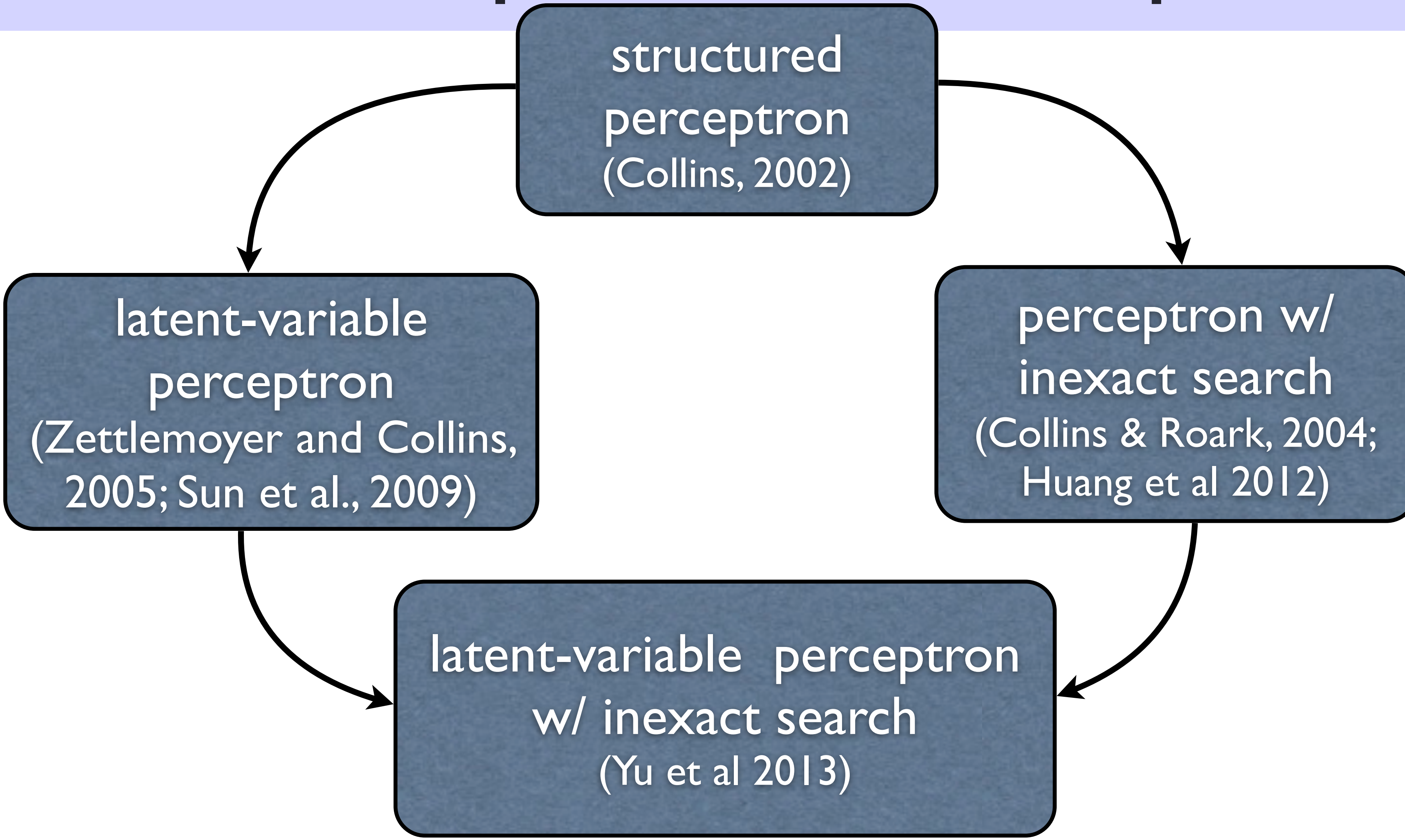
Roadmap of the techniques



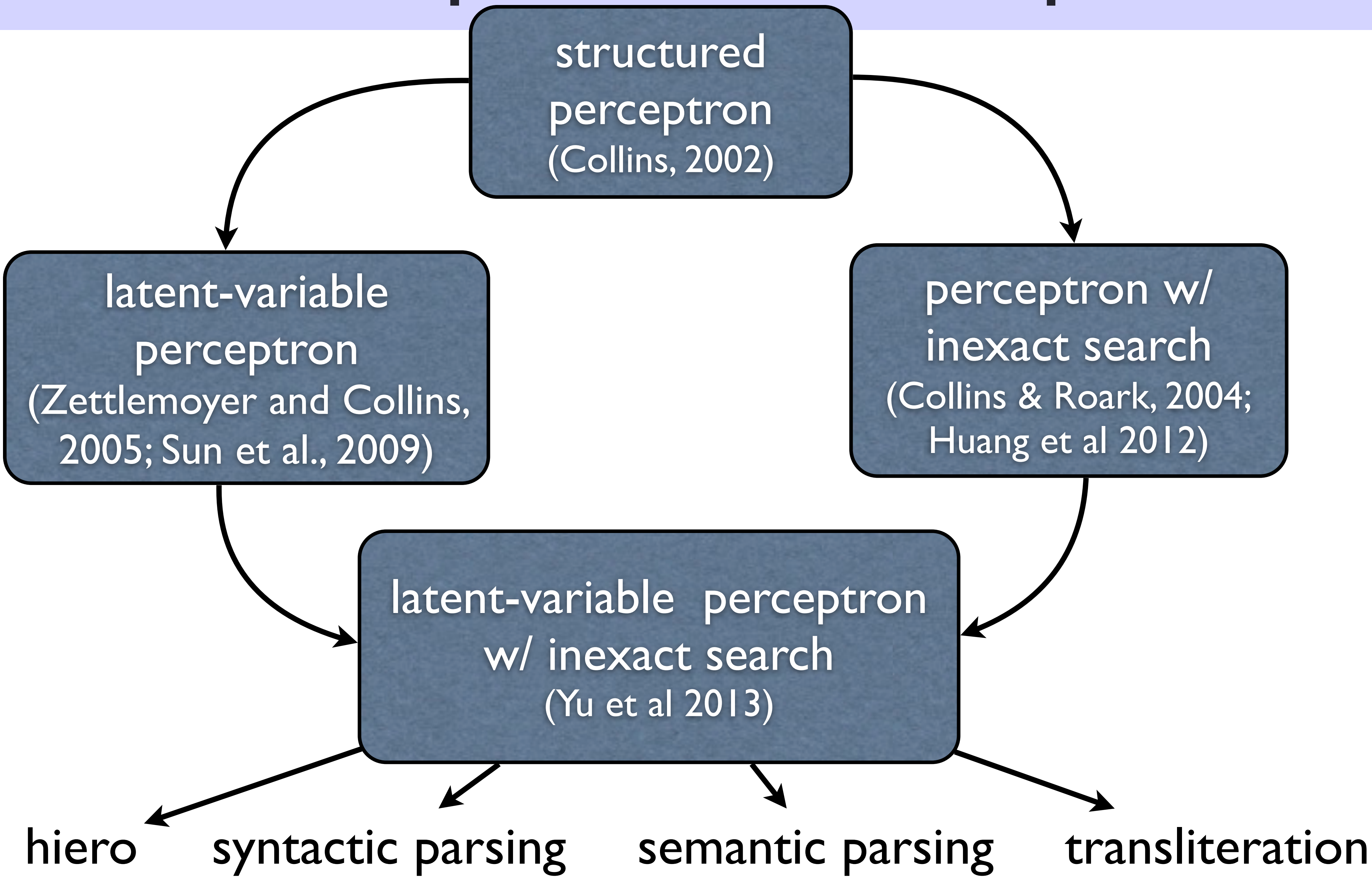
Roadmap of the techniques



Roadmap of the techniques



Roadmap of the techniques



Feature Design

- Dense features:
  - standard phrase-based features (Koehn, 2004)
- Sparse Features:
  - rule-identification features (unique id for each rule)
  - word-edges features
    - lexicalized local translation context within a rule
  - non-local features
    - dependency between consecutive rules

WordEdges Features (local)

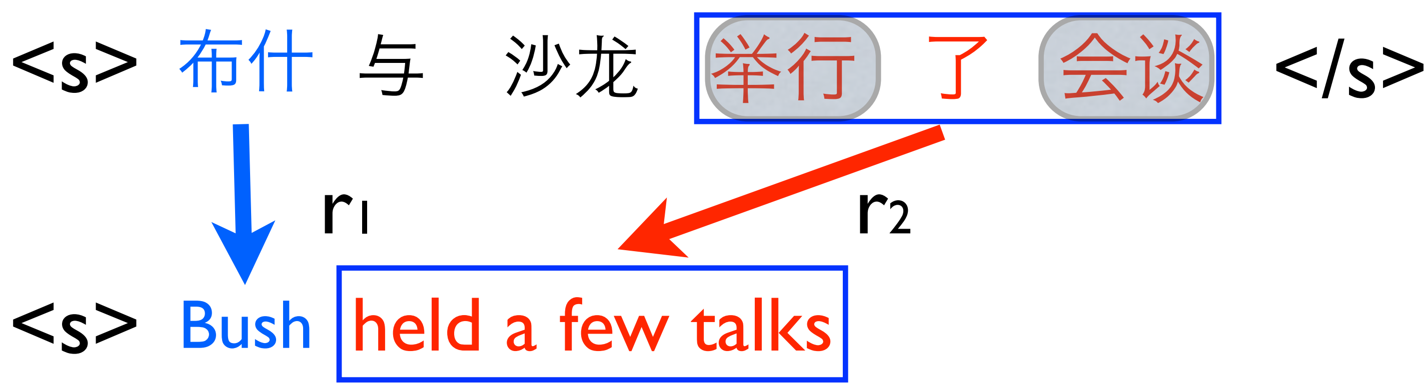
<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

<s> Bush held a few talks

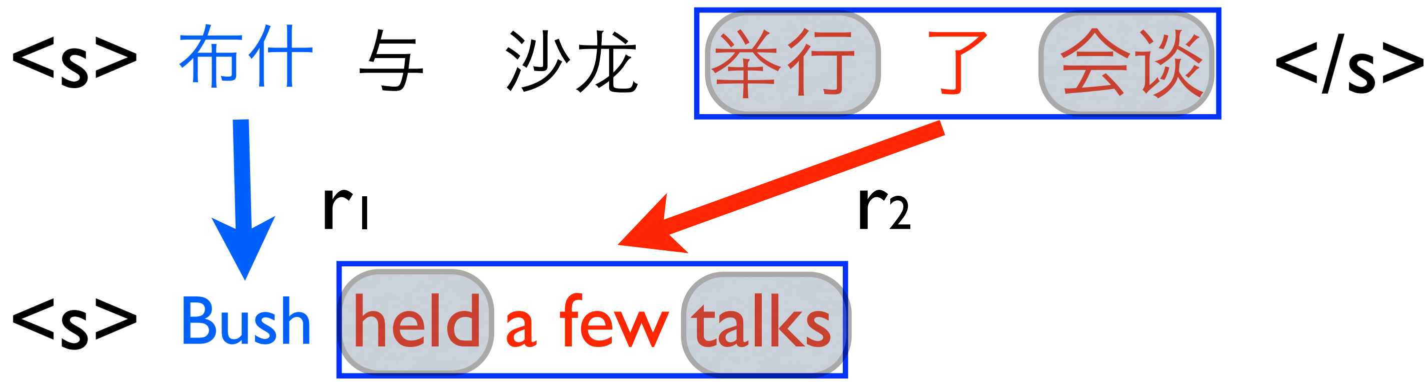
- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

WordEdges Features (local)



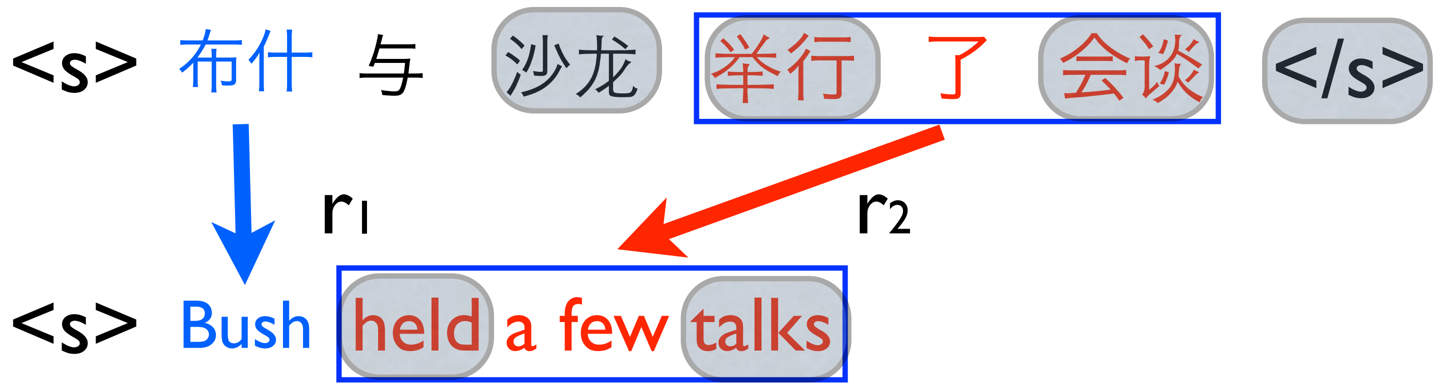
- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

WordEdges Features (local)

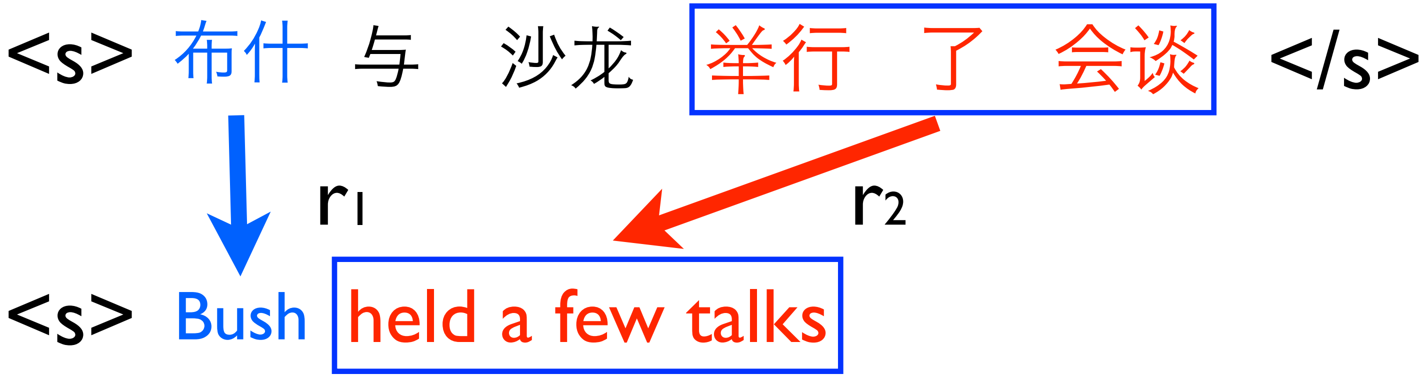
<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

<s> Bush held a few talks

- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

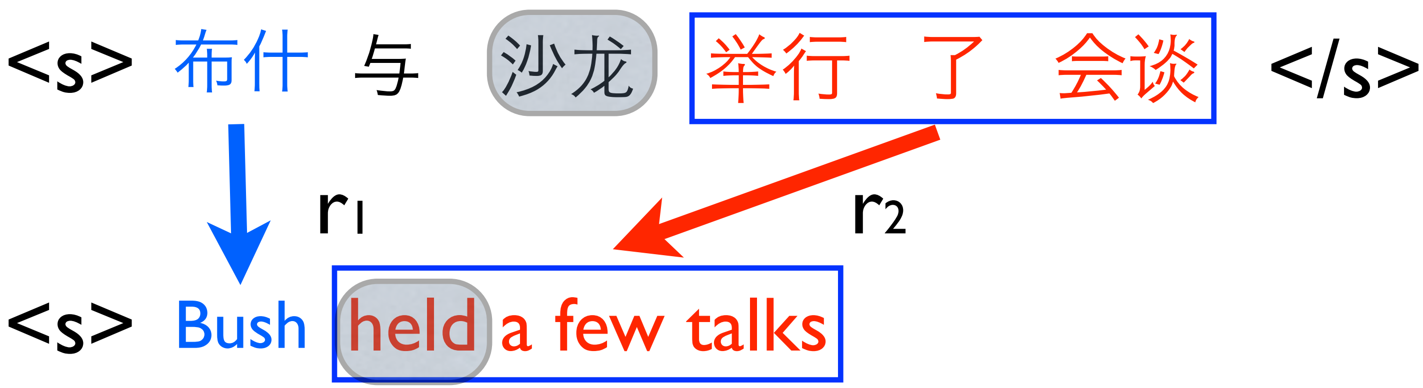
WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

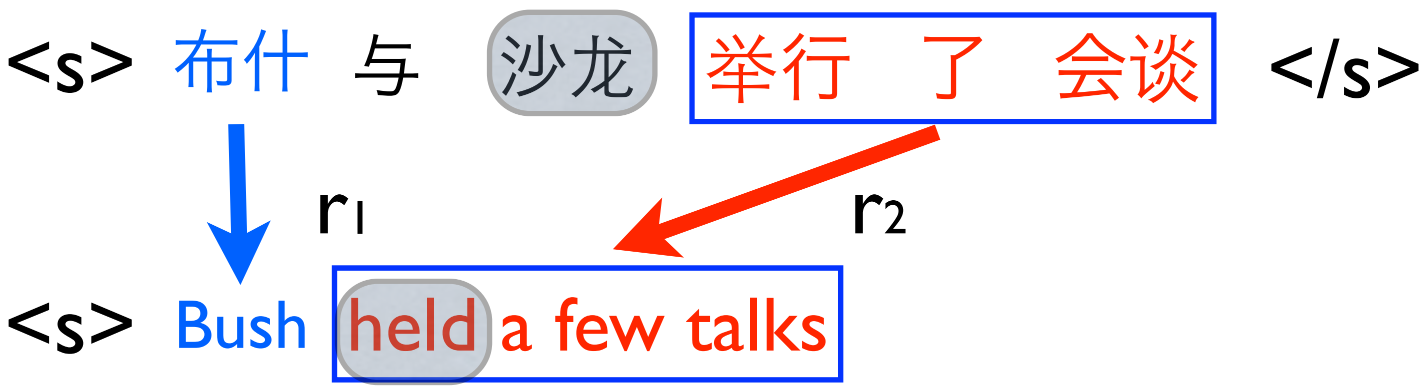
WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

WordEdges Features (local)

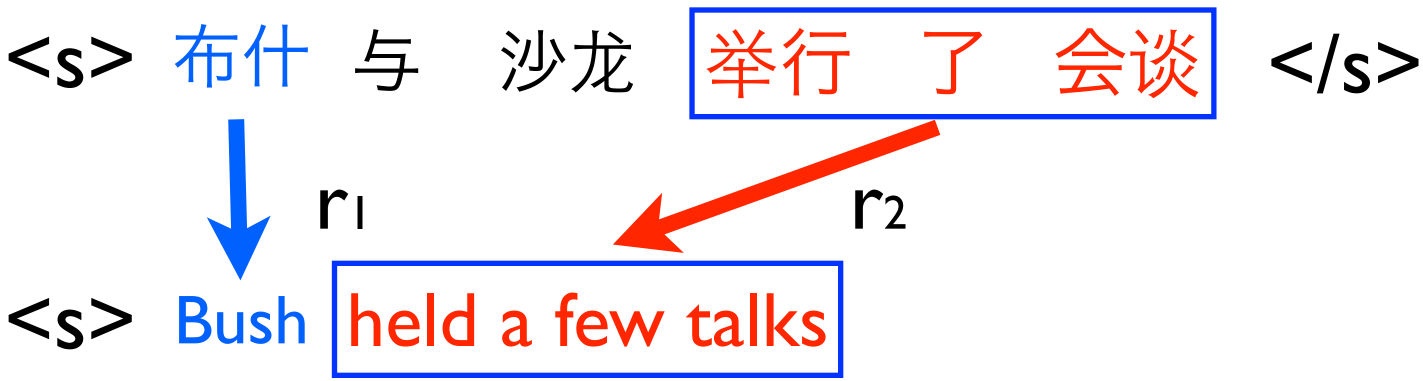


- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

100010=沙龙|held

WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

100010=沙龙|held

WordEdges Features (local)

<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

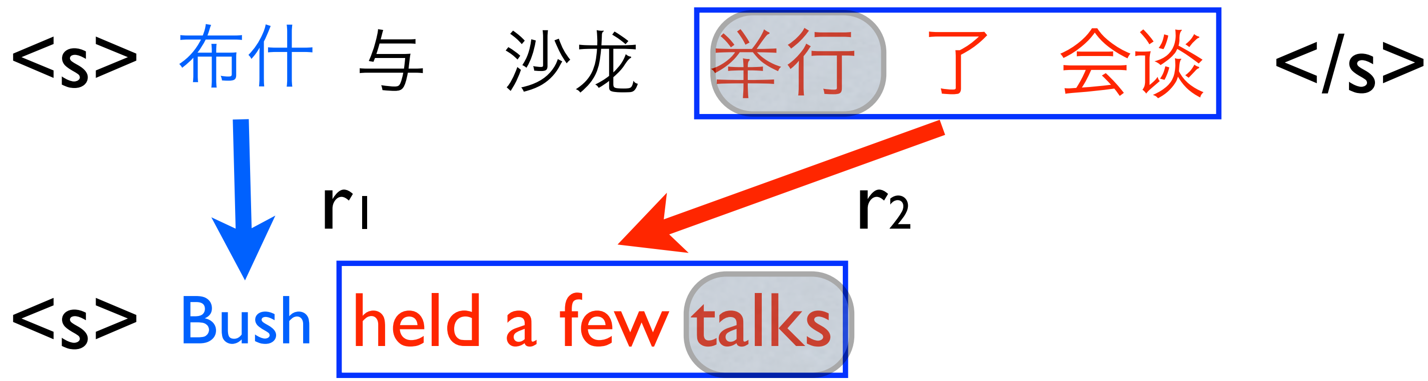
<s> Bush held a few talks

- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

100010=沙龙|held

WordEdges Features (local)



- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

100010=沙龙|held

010001=举行|talks

WordEdges Features (local)

<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

<s> Bush held a few talks

- the first and last Chinese words in the rule
- the first and last English words in the rule
- the two Chinese words surrounding the rule

Combo Features:

100010=沙龙|held

010001=举行|talks

Lexical backoffs and combos

<s> 布什 与 沙龙 举行 了 会谈 </s>

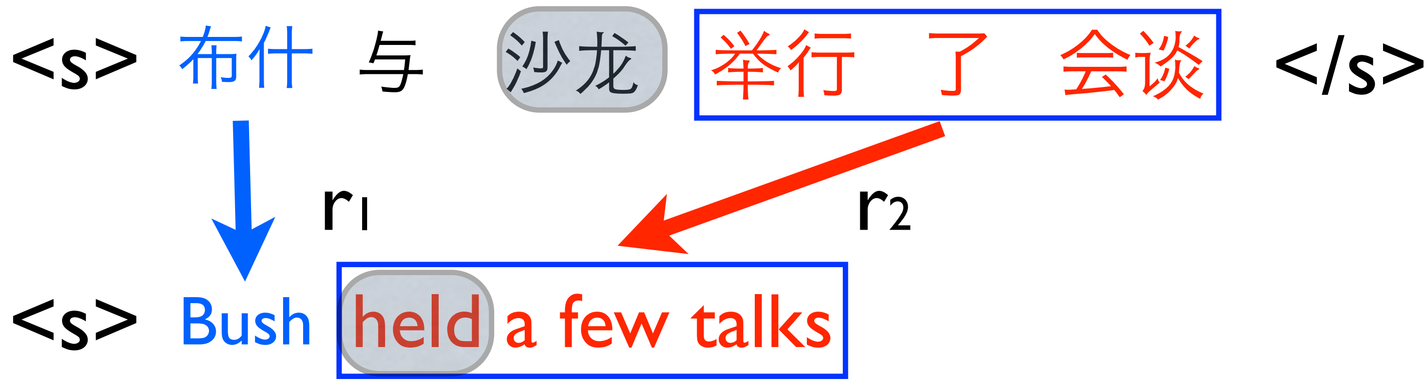
r_1 r_2

<s> Bush held a few talks

- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
    - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

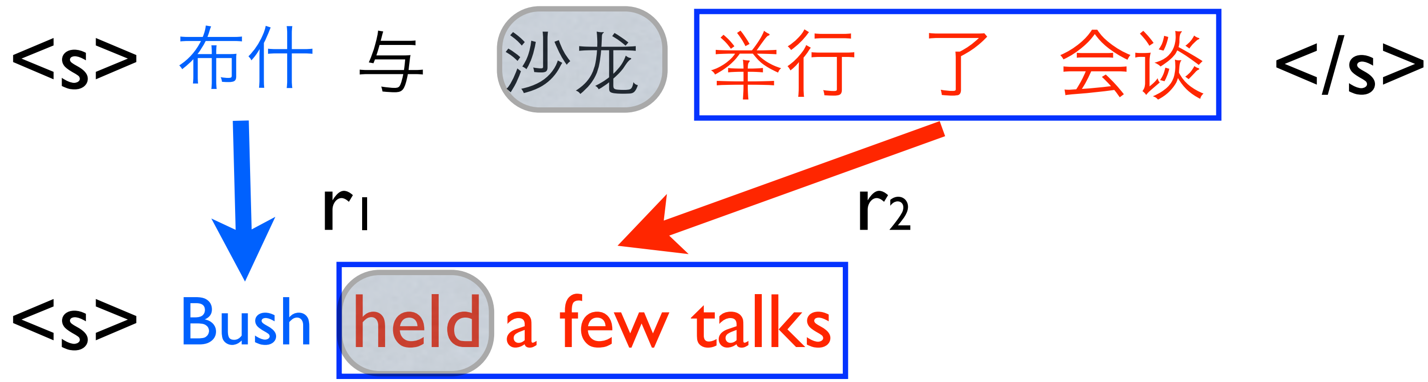
Lexical backoffs and combos



- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

Lexical backoffs and combos



- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

100010=沙龙|held

Lexical backoffs and combos

<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

<s> Bush held a few talks

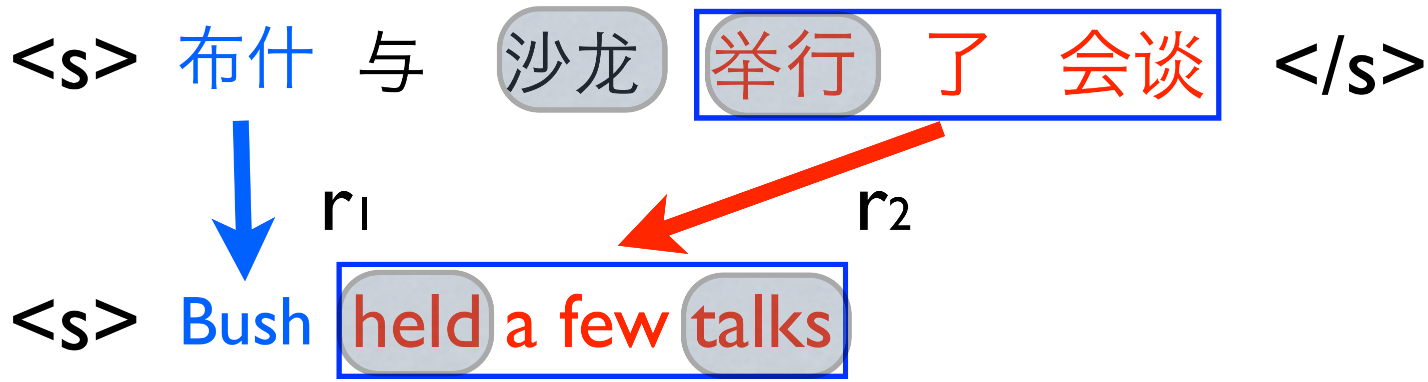
- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

100010=沙龙|held

P00010=NN|held

Lexical backoffs and combos



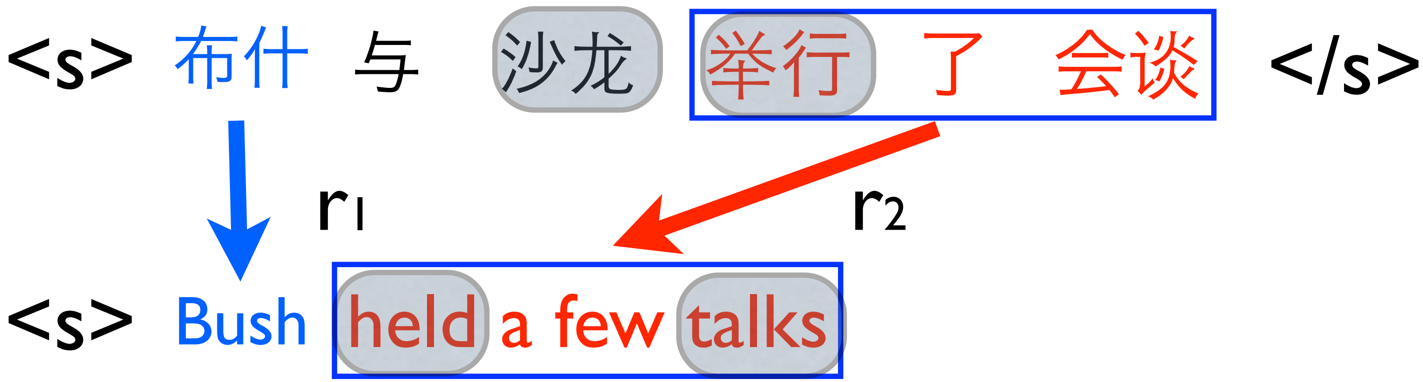
- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

100010=沙龙|held

P00010=NN|held

Lexical backoffs and combos



- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

100010=沙龙|held

P00010=NN|held

010001=举行|talks

Lexical backoffs and combos

<s> 布什 与 沙龙 举行 了 会谈 </s>

r_1 r_2

<s> Bush held a few talks

- Lexical features are often too sparse
- 6 kinds of lexical backoffs with various budgets
  - total budget can't exceed 10 (bilexical)

Chinese	English	class size		budget
word		52.9k	64.2k	5
characters	-	3.7k	-	3
Brown cluster, full string		200		3
Brown cluster, prefix 6		6	8	2
Brown cluster, prefix 4		4	4	2
POS tag		52	36	2
word type	-	4	-	1

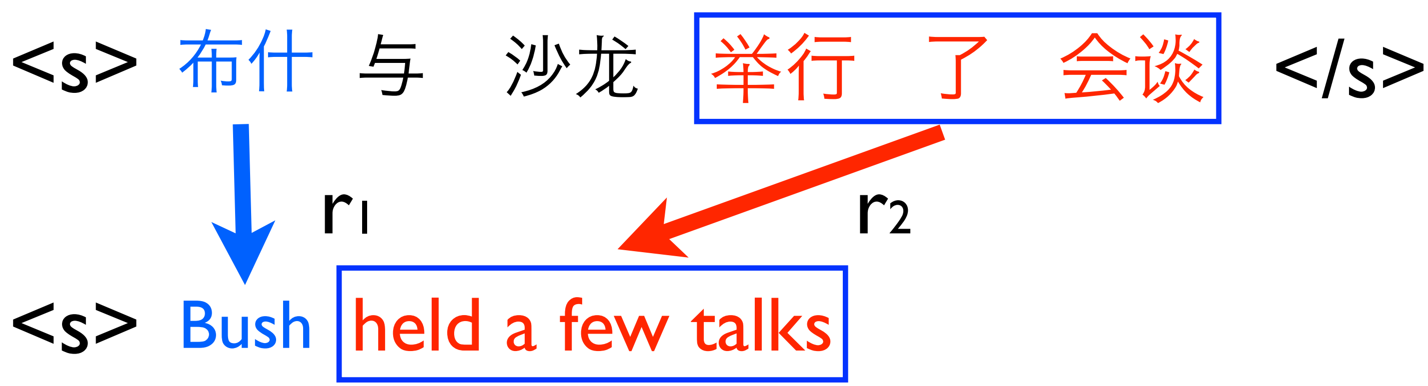
100010=沙龙|held

P00010=NN|held

010001=举行|talks

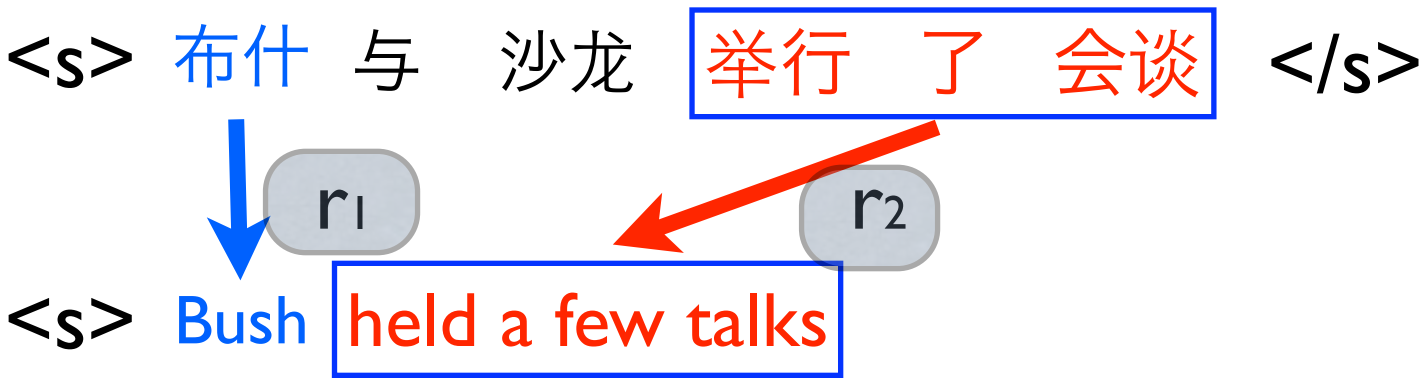
0c0001=举|talks

Non-Local Features (trivial)



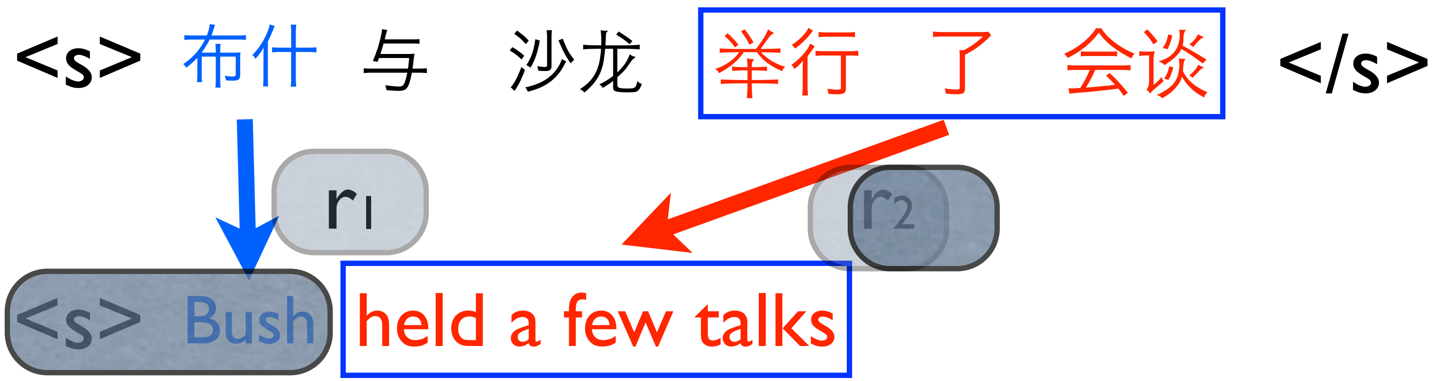
- two consecutive rule ids (rule bigram model)
- the last two English words and the current rule
- should explore a lot more!

Non-Local Features (trivial)



- two consecutive rule ids (rule bigram model)
- the last two English words and the current rule
- should explore a lot more!

Non-Local Features (trivial)



- two consecutive rule ids (rule bigram model)
- the last two English words and the current rule
- should explore a lot more!

Experiments

- Date sets

Experiments

- Date sets

Scale	Language	sent.	dev	tst

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

Table 3: The ratio of sentence and word coverage on small and large training sets.

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

10x dev

Table 3: The ratio of sentence and word coverage on small and large training sets.

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

10x dev 120x dev

Table 3: The ratio of sentence and word coverage on small and large training sets.

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		
Large	Sp-En	170k	newstest2012	newtest2013

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

10x dev   120x dev

Table 3: The ratio of sentence and word coverage on small and large training sets.

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		
Large	Sp-En	170k	newstest2012	newtest2013

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%
		10x dev		120x dev

Table 3: The ratio of sentence and word coverage on small and large training sets.

Sp-En	sent.	word.
ratio	55%	43.9%

Experiments

- Date sets

Scale	Language	sent.	dev	tst
Small	Ch-En	30k	nist06 news	nist08 news
Large		240k		
Large	Sp-En	170k	newstest2012	newtest2013

	small		large	
	sent.	words	sent.	words
full	21.4%	8.8%	32.1%	12.7%
+prefix	61.3%	24.6%	67.3%	32.8%

10x dev **120x dev**

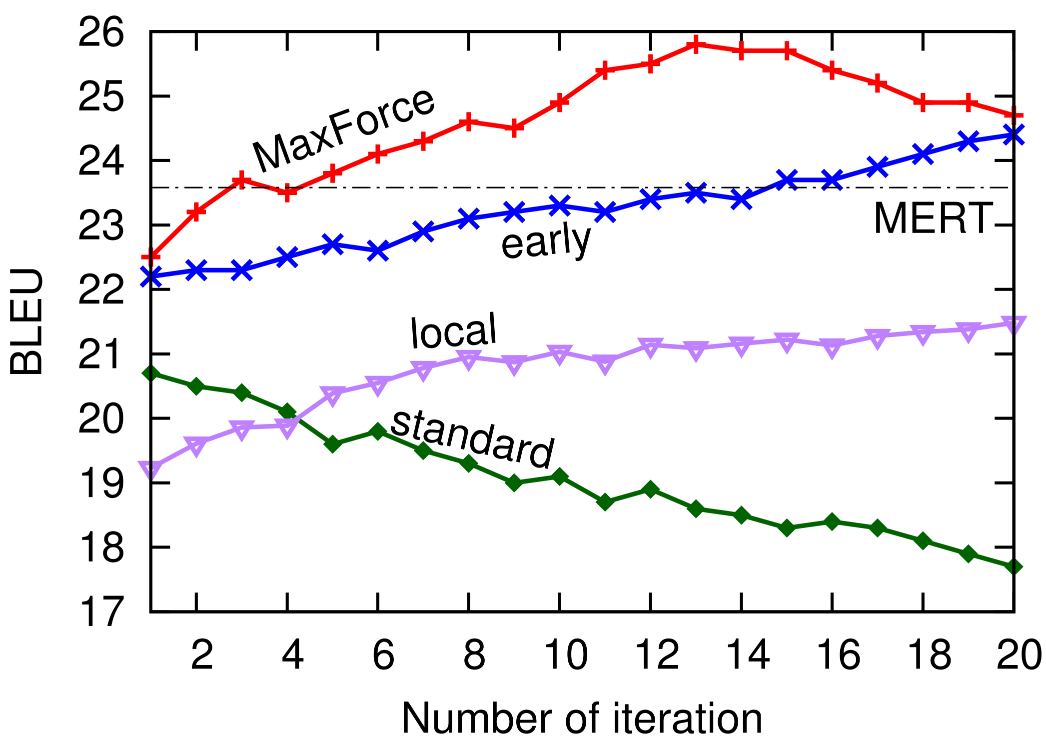
Table 3: The ratio of sentence and word coverage on small and large training sets.

Sp-En	sent.	word.
ratio	55%	43.9%

31x dev

Perceptron: std, early, and max-violation

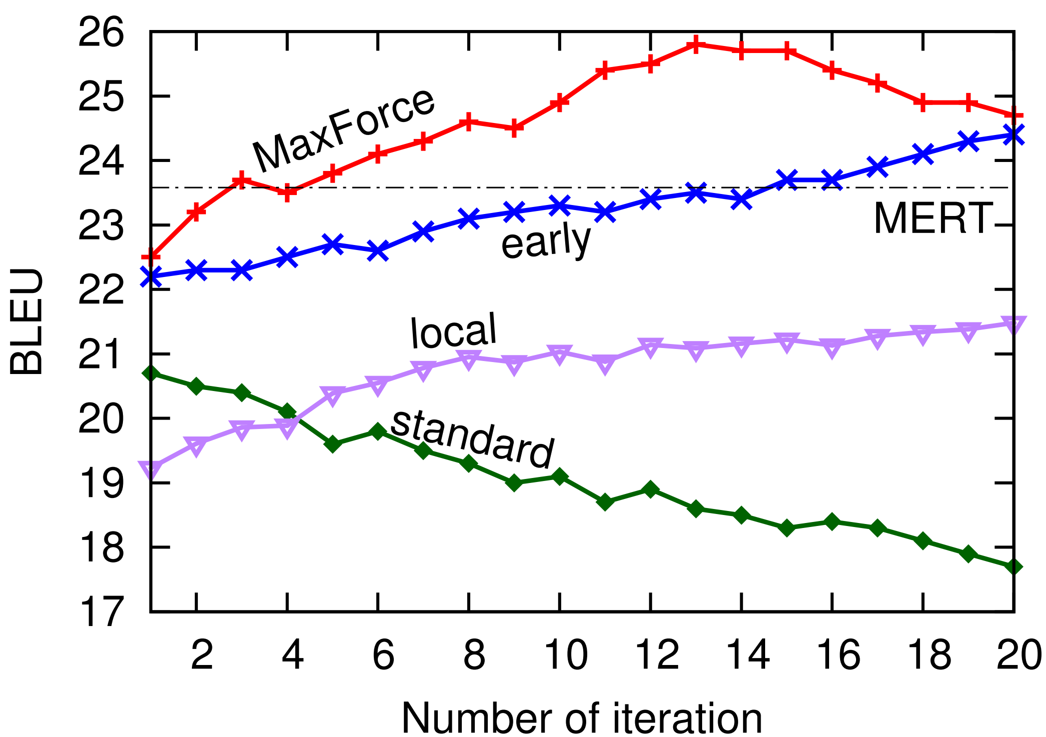
- standard perceptron (Liang et al's "bold") works poorly
  - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update



Perceptron: std, early, and max-violation

- standard perceptron (Liang et al's "bold") works poorly
  - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update

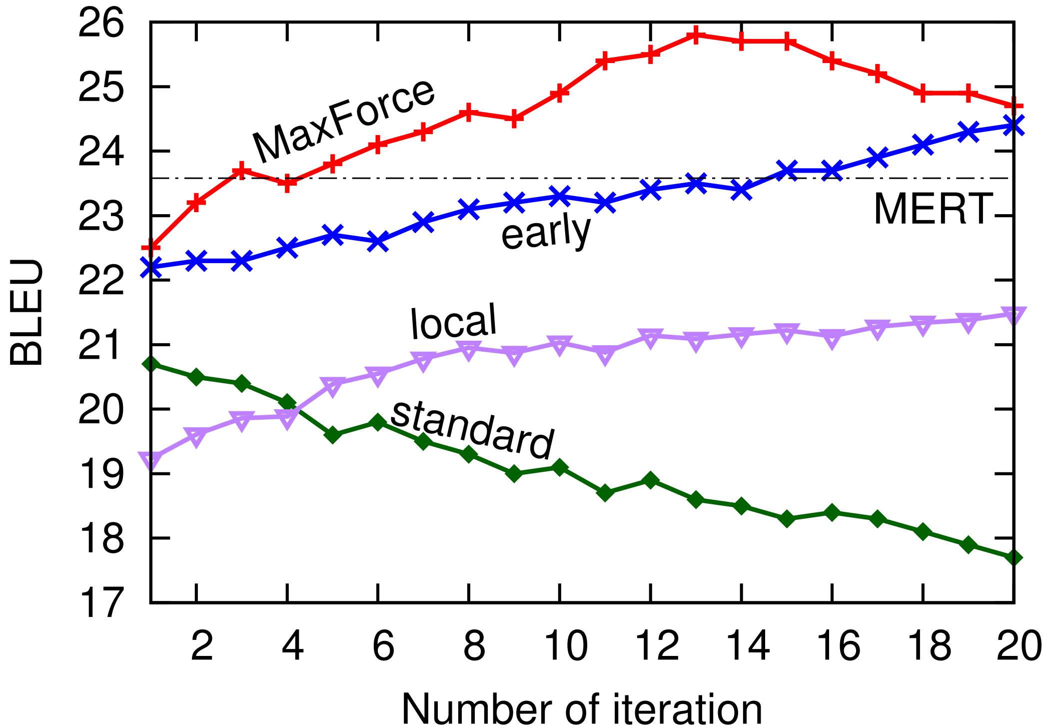
this explains why Liang et al '06 failed
std ~ "bold"; local ~ "local"

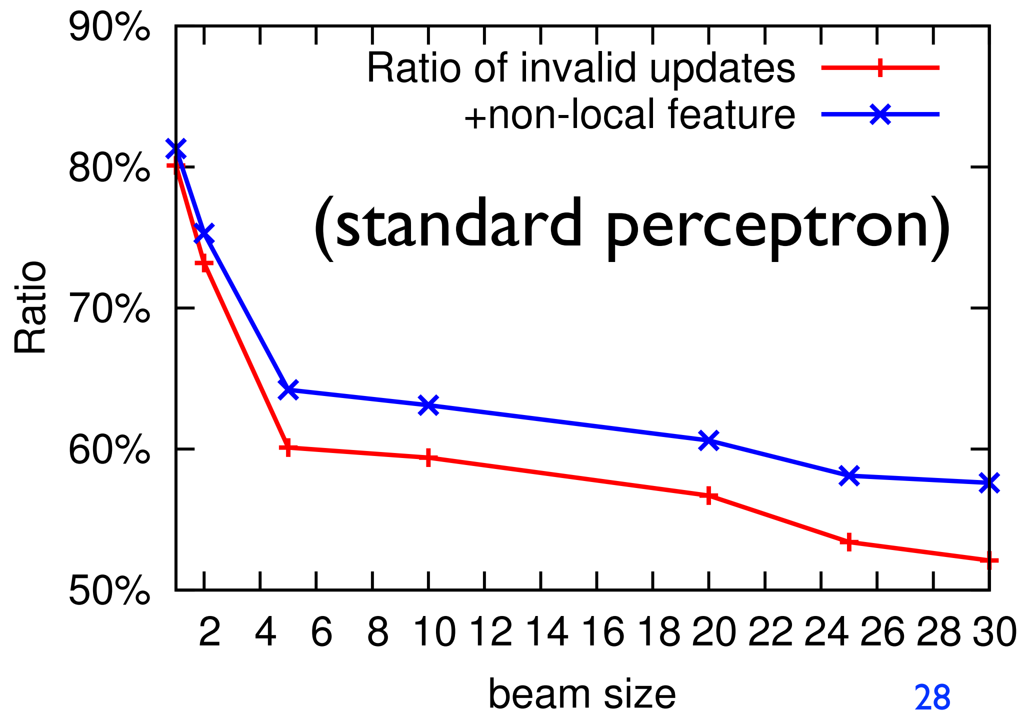


Perceptron: std, early, and max-violation

- standard perceptron (Liang et al's "bold") works poorly
  - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update

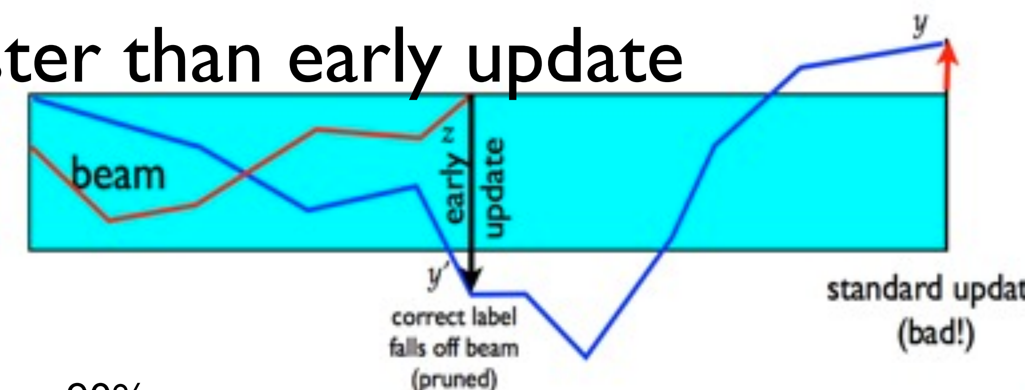
this explains why Liang et al '06 failed
std ~ "bold"; local ~ "local"



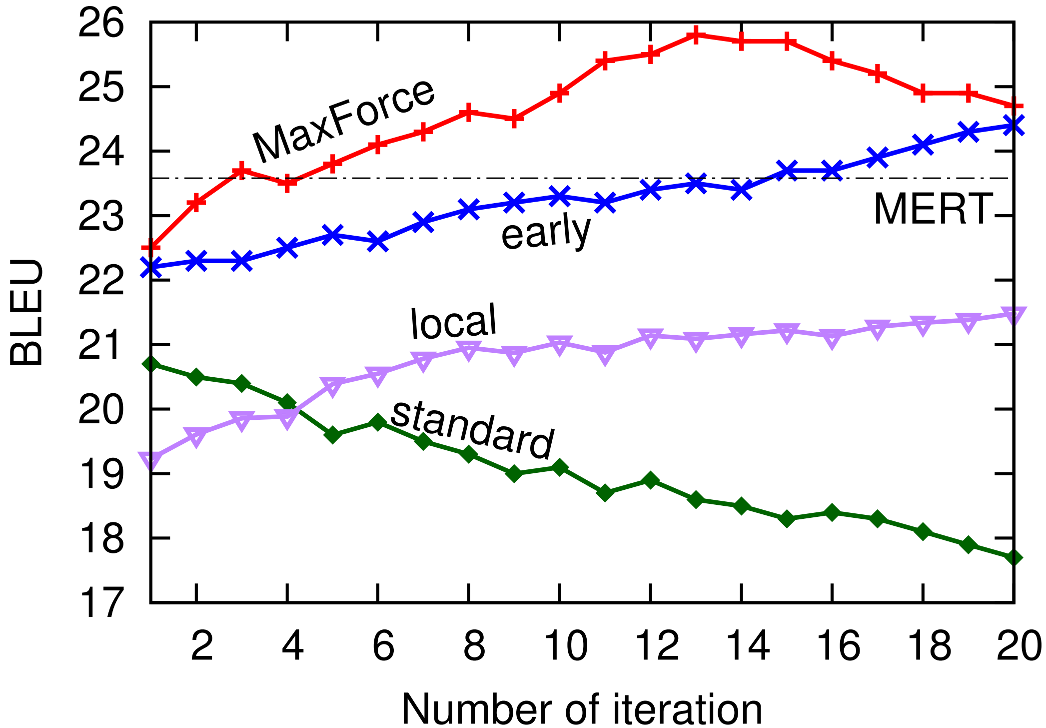


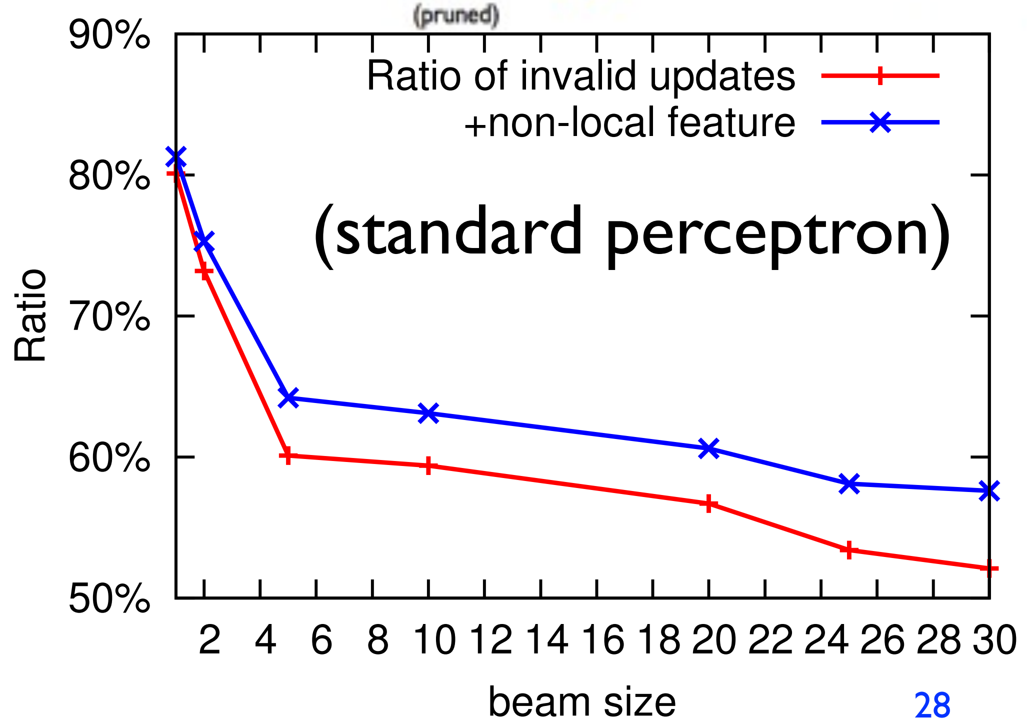
Perceptron: std, early, and max-violation

- standard perceptron (Liang et al's “bold”) works poorly
  - b/c invalid update ratio is very high (search quality is low)
- max-violation converges faster than early update



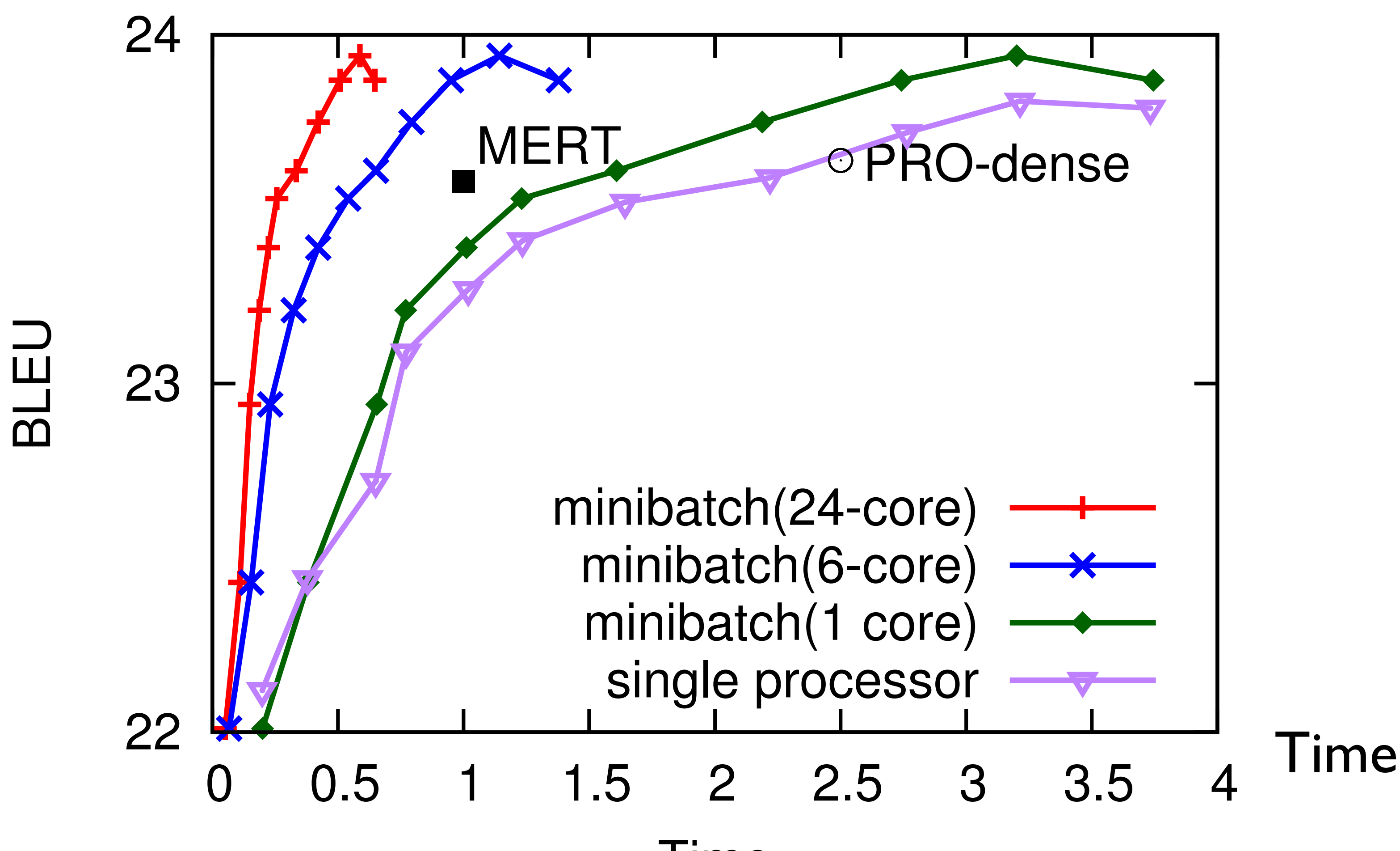
this explains why Liang et al ’06 failed
std ~ “bold”; local ~ “local”





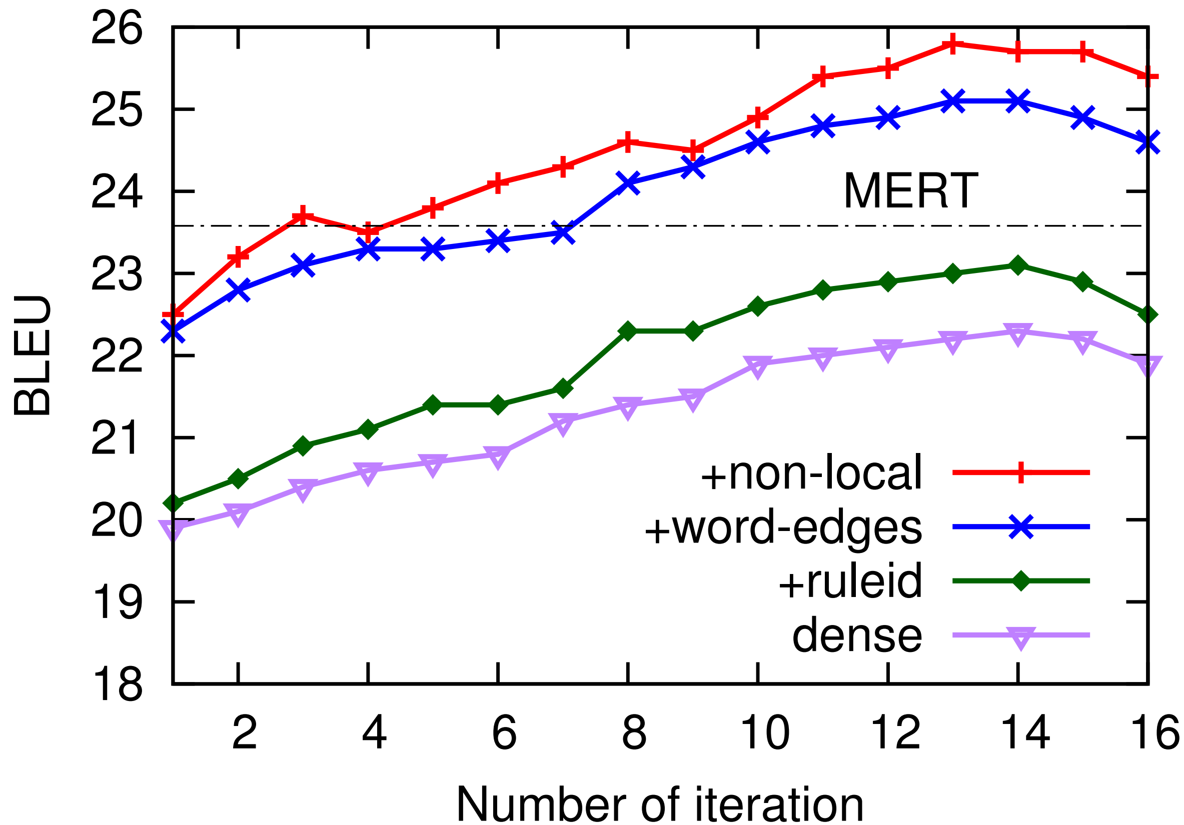
Parallelized Perceptron

- mini-batch perceptron (Zhao and Huang, 2013) much faster than iterative parameter mixing (McDonald et al, 2010)
  - 6 CPUs => ~4x speedup; 24 CPUs => ~7x speedup



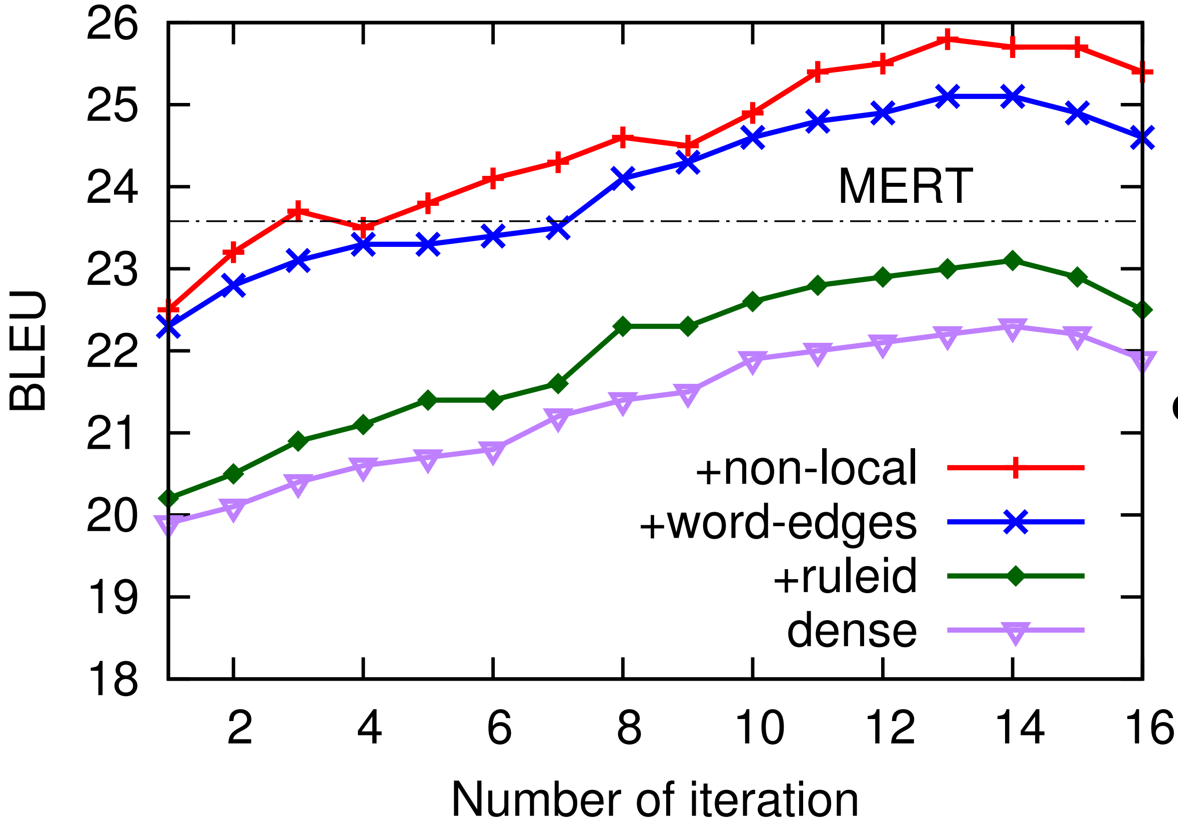
Internal comparison with different features

- dense: 11 standard features for phrase-based MT
- ruleid: rule identification feature
- word-edges: word-edges features with back-offs
- non-local: non-local features with back-offs



Internal comparison with different features

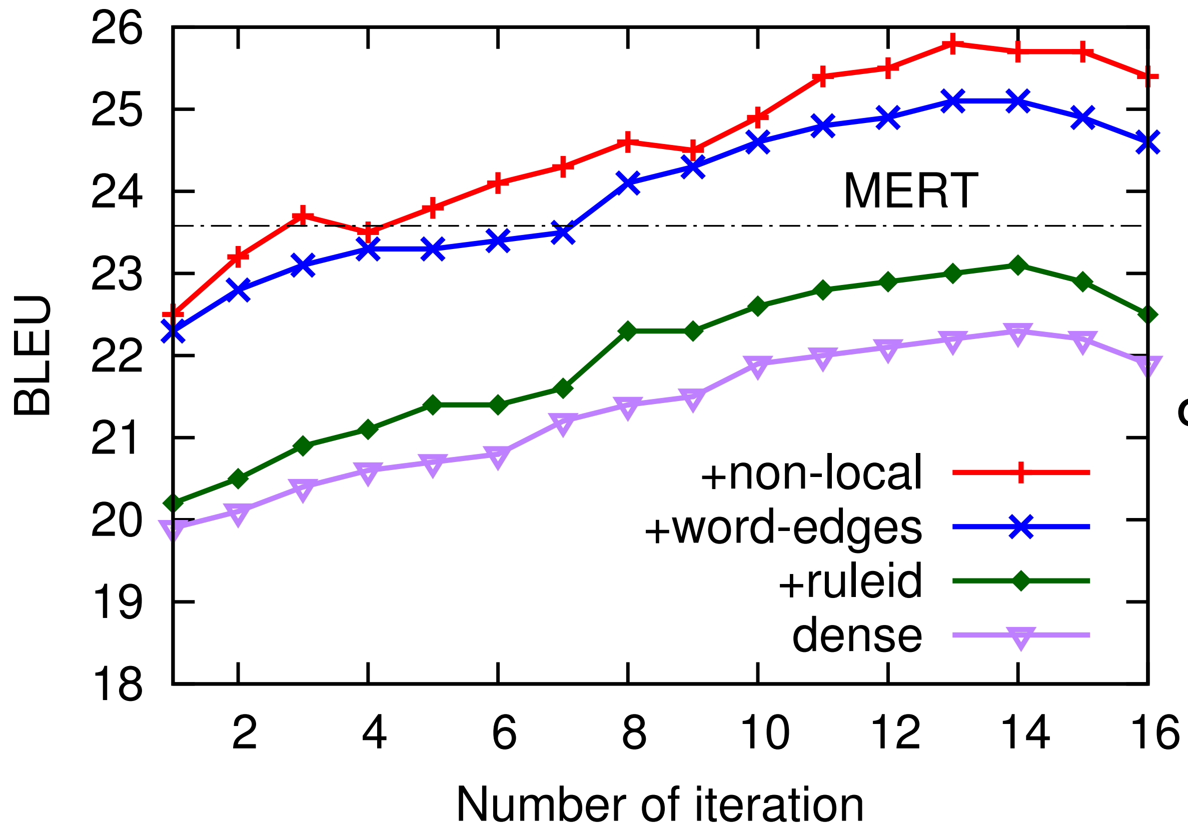
- dense: 11 standard features for phrase-based MT
- ruleid: rule identification feature
- word-edges: word-edges features with back-offs
- non-local: non-local features with back-offs



dense: 11 features

Internal comparison with different features

- dense: 11 standard features for phrase-based MT
- ruleid: rule identification feature
- word-edges: word-edges features with back-offs
- non-local: non-local features with back-offs



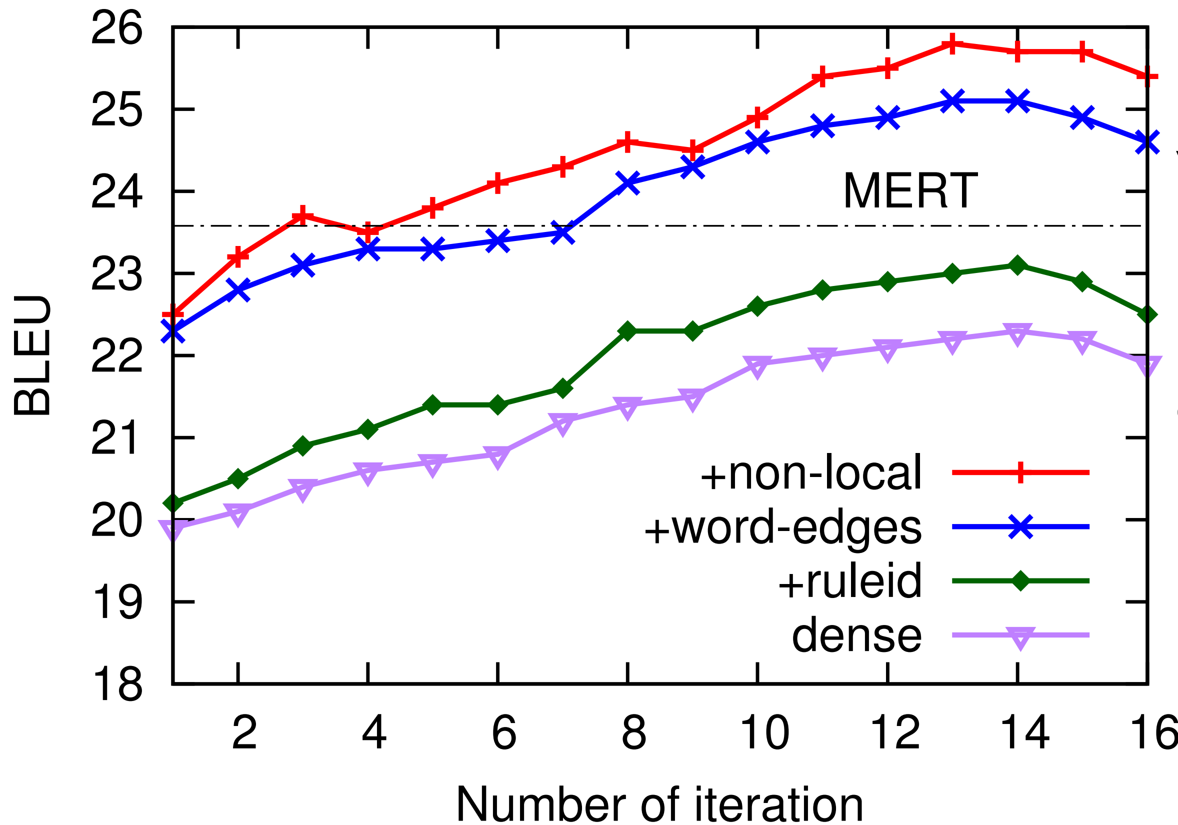
ruleid: 0.1%

+0.9 bleu

dense: 11 features

Internal comparison with different features

- dense: 11 standard features for phrase-based MT
- ruleid: rule identification feature
- word-edges: word-edges features with back-offs
- non-local: non-local features with back-offs



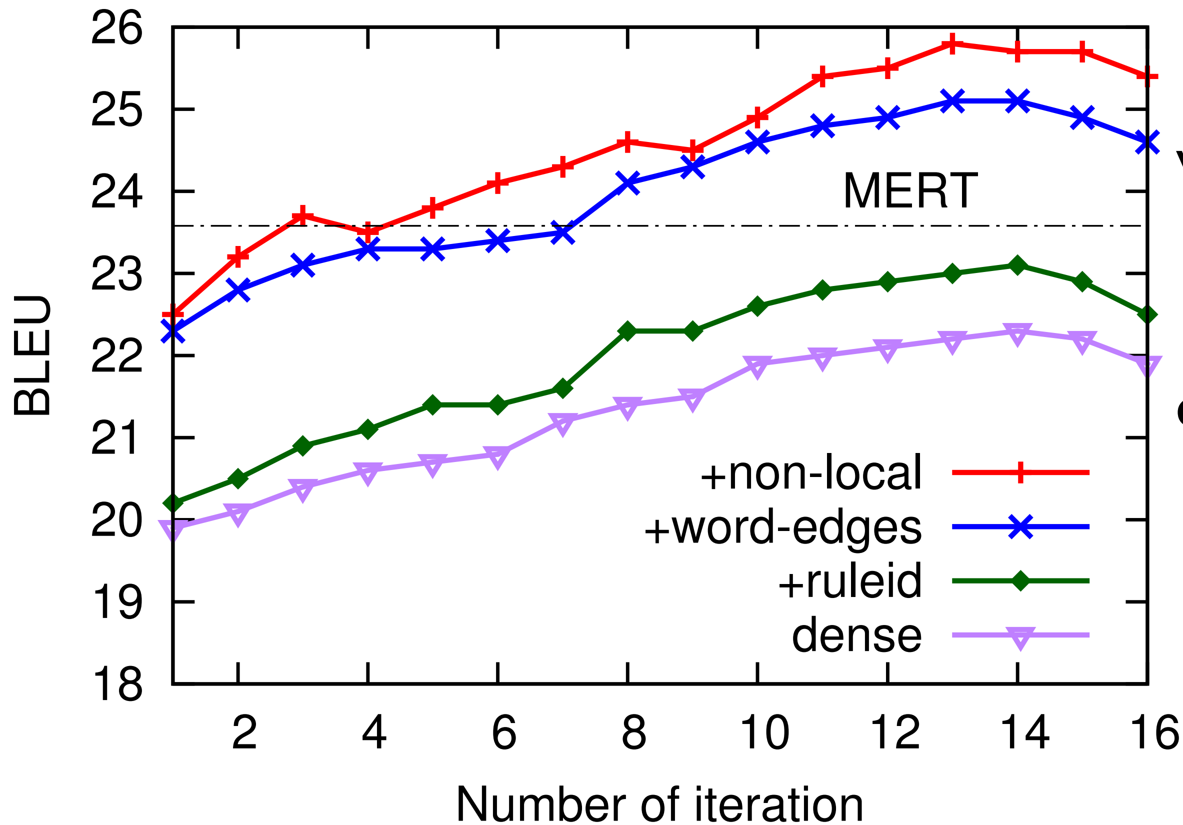
wordedges: 99.6% +2.3

ruleid: 0.1% +0.9 bleu

dense: 11 features

Internal comparison with different features

- dense: 11 standard features for phrase-based MT
- ruleid: rule identification feature
- word-edges: word-edges features with back-offs
- non-local: non-local features with back-offs



non-local: 0.3% +0.7

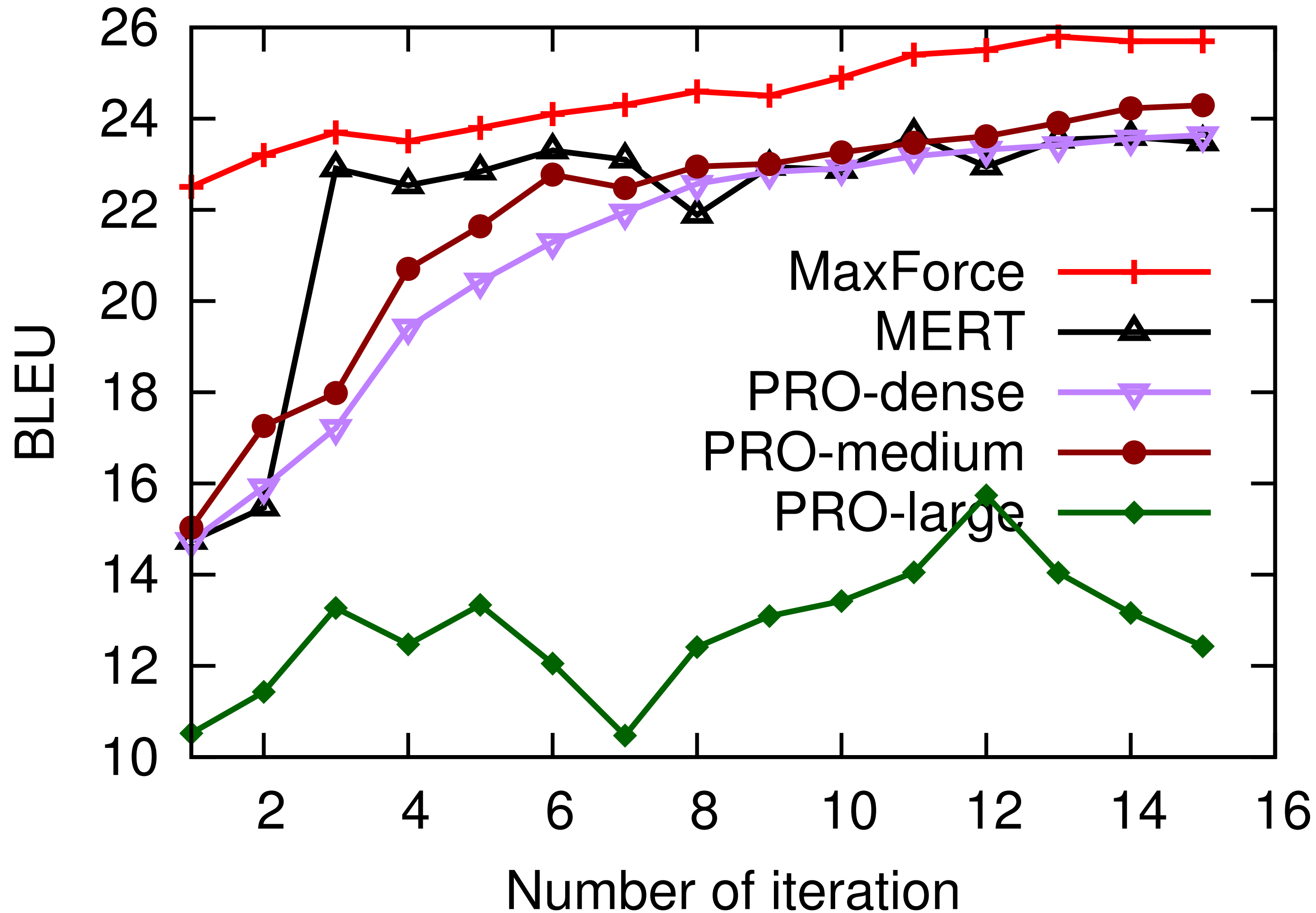
wordedges: 99.6% +2.3

ruleid: 0.1% +0.9 bleu

dense: 11 features

External comparison with MERT & PRO

- MERT, PRO-dense/medium/sparse all tune on dev-set
- PRO-sparse use the same feature as ours



Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6
			3k	26.3	23.0

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6
			3k	26.3	23.0
			36k	17.7	14.3

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6
			3k	26.3	23.0
			36k	17.7	14.3
	MaxForce	**Train set**	**23M**	**27.8**	**24.5**

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6
			3k	26.3	23.0
			36k	17.7	14.3
	MaxForce	**Train set**	**23M**	**27.8**	**24.5**
				+2.3	

Final Results on FBIS Data

- Moses: state-of-the-art phrase-based system in C++
- Cubit: phrase-based system (Huang and Chiang, 2007) in python
  - almost identical baseline scores with MERT
  - max-violation takes ~47 hours on 24 CPUs (23M features)

System	Alg.	Tune on	Features	Dev	Test
Moses	MERT	dev set	11	25.5	22.5
Cubit	MERT	dev set	11	25.4	22.5
	PRO	dev set	11	25.6	22.6
			3k	26.3	23.0
			36k	17.7	14.3
	MaxForce	**Train set**	**23M**	**27.8**	**24.5**
				+2.3	**+2.0**

Results on Spanish-English set

- Data-set: Europarl corpus, 170k sentences
- dev/test set: newtest2012 / 2013 (one-reference only)
  - +1 in 1-ref bleu ~ +2 in 4-ref bleu
  - bleu improvement is comparable to Chinese w/ 4-refs

system	algorithm	#feat.	dev	test
Moses	Mert	11	27.4	24.4
Cubit	MaxForce	21M	**28.7**	**25.5**

Sp-En	sent.	word.
Reachable ratio	55%	43.9%

Results on Spanish-English set

- Data-set: Europarl corpus, 170k sentences
- dev/test set: newtest2012 / 2013 (one-reference only)
  - +1 in 1-ref bleu ~ +2 in 4-ref bleu
  - bleu improvement is comparable to Chinese w/ 4-refs

system	algorithm	#feat.	dev	test
Moses	Mert	11	27.4	24.4
Cubit	MaxForce	21M	**28.7**	**25.5**
			+1.3	**+1.1**

Sp-En	sent.	word.
Reachable ratio	55%	43.9%

Conclusion

- a simple yet effective online learning approach for MT
  - scaled to (a large portion of) the training set for the first time
  - able to incorporate 20M sparse lexicalized features
  - no need to define BLEU+1, or hope/fear derivations
  - no learning rate or hyperparameters
  - +2.3/+2.0 BLEU points better than MERT/PRO
- the three ingredients that made it work
  - violation-fixing perceptron: early-update and max-violation
  - forced decoding lattice helps
  - minibatch parallelization scales it up to big data

Roadmap of the techniques

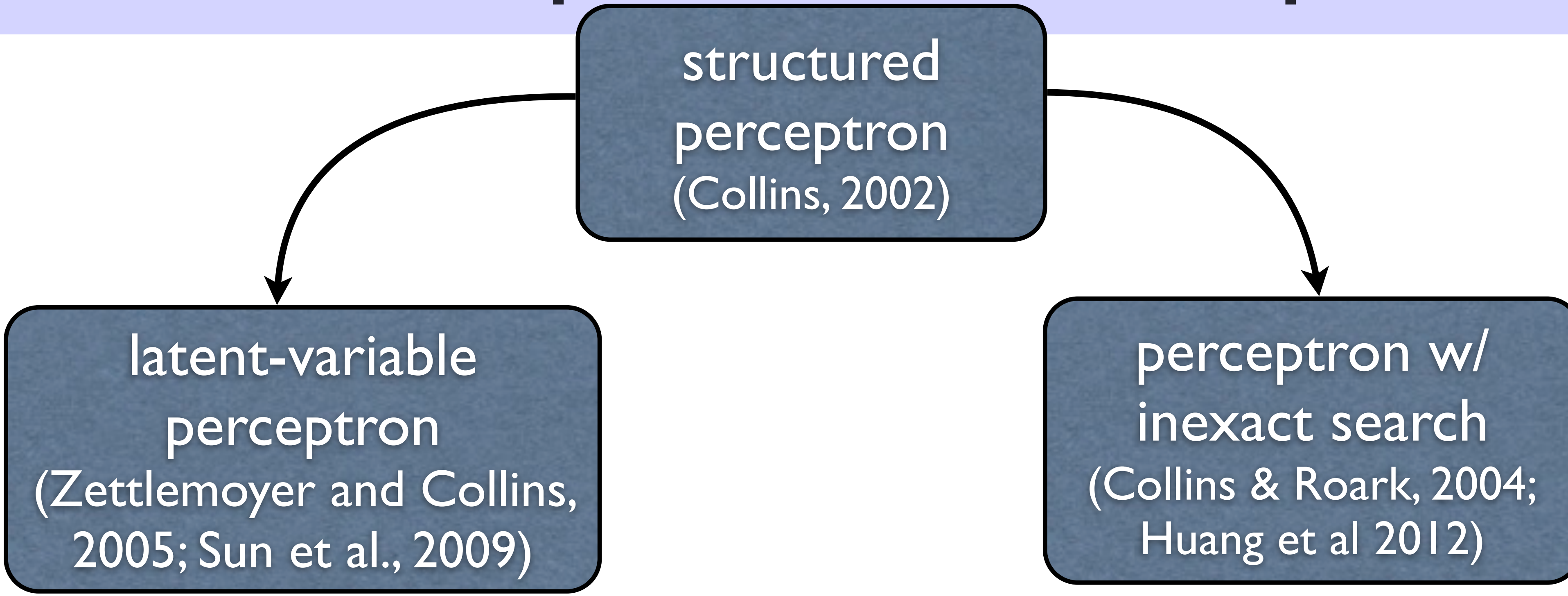
structured
perceptron
(Collins, 2002)

Roadmap of the techniques

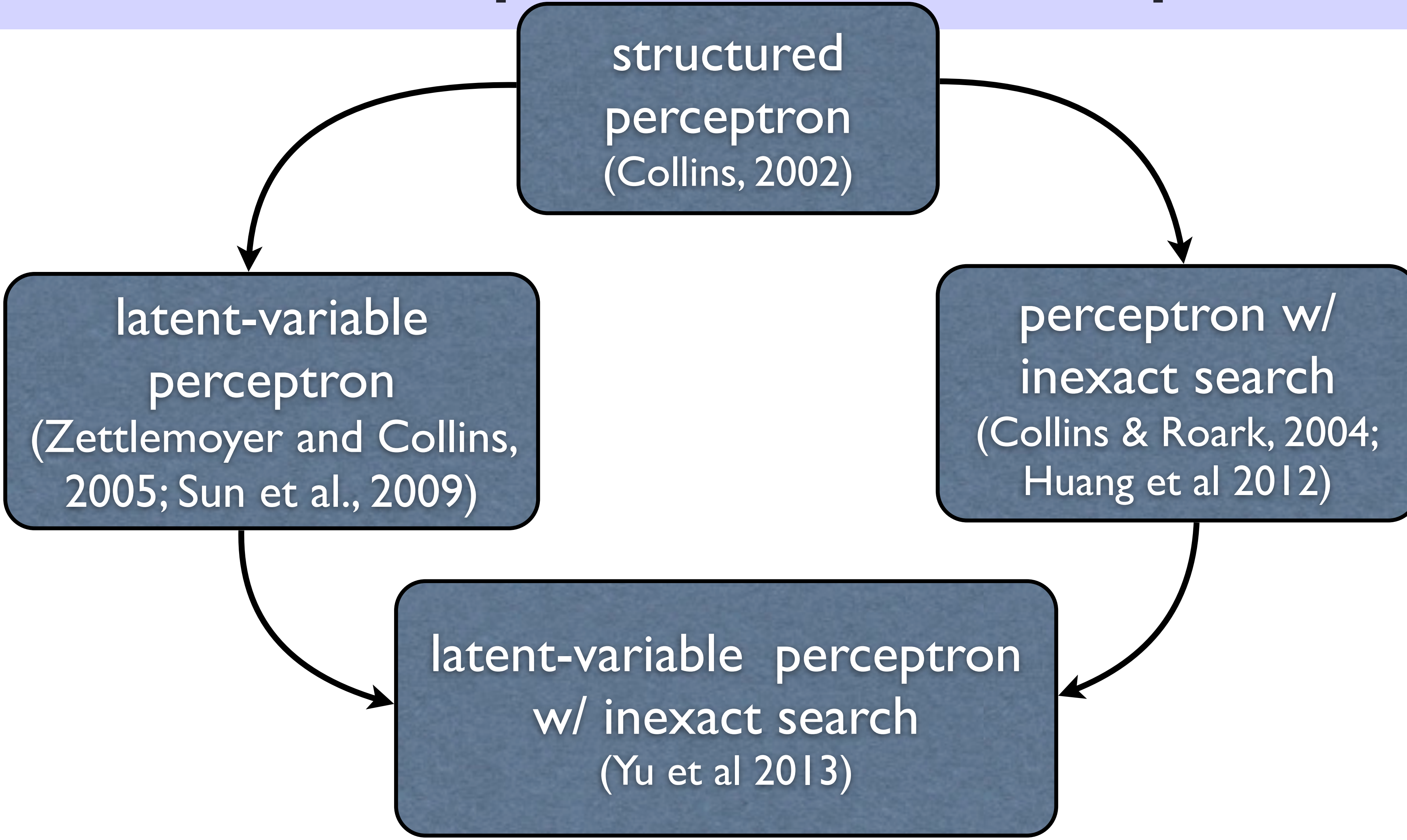
structured perceptron
(Collins, 2002)

latent-variable perceptron
(Zettlemoyer and Collins, 2005; Sun et al., 2009)

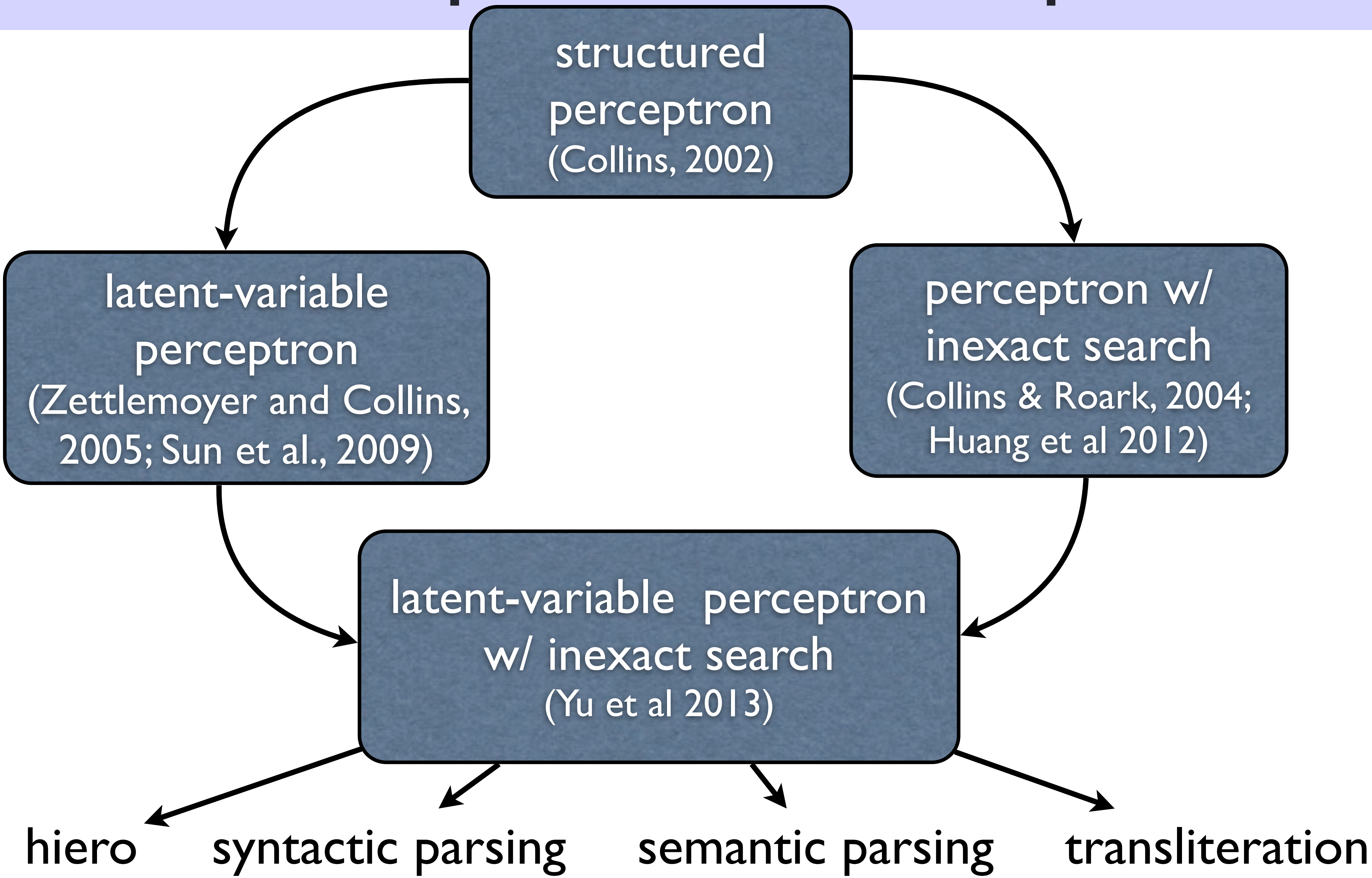
Roadmap of the techniques



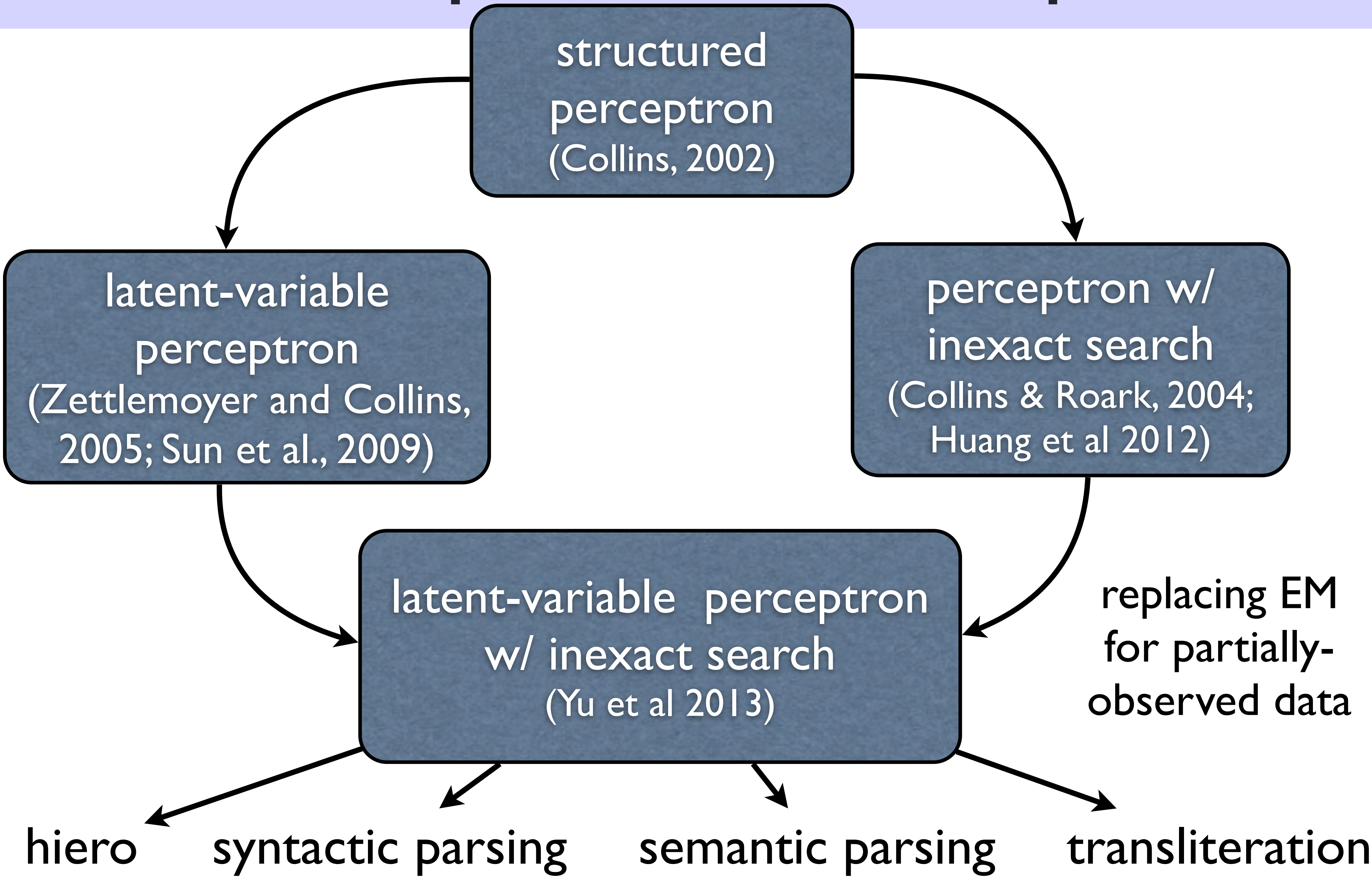
Roadmap of the techniques



Roadmap of the techniques



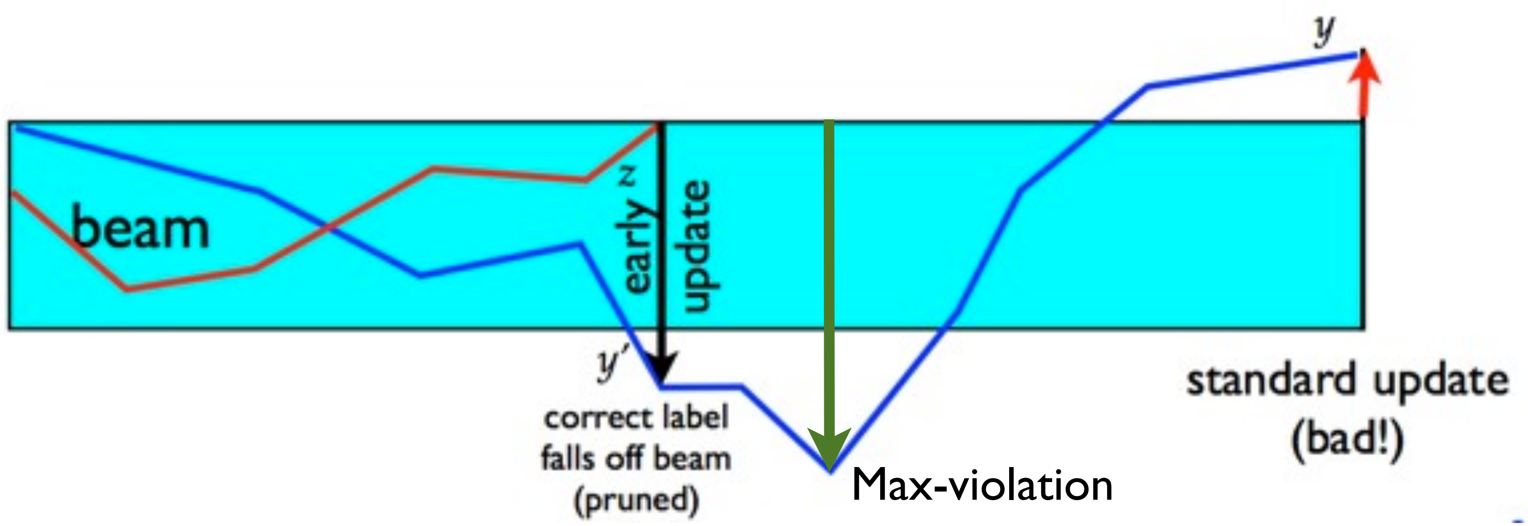
Roadmap of the techniques



20 years of Statistical MT

- word alignment: IBM models (Brown et al 90, 93)
- translation model (choose one from below)
  - SCFG (ITG: Wu 95, 97; Hiero: Chiang 05, 07) or STSG (GHKM 04, 06; Liu+ 06; Huang+ 06)
  - PBMT (Och+Ney 02; Koehn et al 03)
- evaluation metric: BLEU (Papineni et al 02)
- decoding algorithm: cube pruning (Chiang 07; Huang+Chiang 07)
- training algorithm (choose one from below)
  - MERT (Och 03): ~10 dense features on dev set
  - MIRA (Chiang et al 08-12) or PRO (Hopkins+May 11): ~10k feats on dev set
  - MaxForce: 20M+ feats on training set; +2/+1.5 BLEU over MERT/PRO
    - Max-Violation Perceptron with Forced Decoding: fixes search errors
    - first successful effort of online large-scale discriminative training for MT

When learning with vastly inexact search, you should use a principled method such as max-violation.



Thank you!