Efficient Incremental Decoding for Tree-to-String Translation

Liang Huang
Information Sciences Institute
University of Southern California

Haitao Mi
Institute of Computing Technology
Chinese Academy of Sciences

EMNLP 2010 talk, Cambridge, MA, Oct 2010

MT: Phrase-based vs. Syntax-based

- most of statistical machine translation falls into
  - phrase-based models
    - allow arbitrary reorderings: exponential-time decoding
    - in practice: quadratic-time beam search
      - linear-time with constant distortion limit; pretty fast
  - syntax-based models
    - grammar-based reorderings: polynomial-time decoding
    - in practice: slower than phrase-based when with LM
- Q: borrow phrase-based decoding for syntax-based?

Preview of Results

- a phrase-based-style, incremental decoding algorithm for tree-to-string translation
- polynomial-time in theory, linear-time in practice
- 30 times faster than phrase-based Moses

	in theory	*in practice*
phrase-based	exponential	quadratic
tree-to-string	polynomial	linear

Preview of Results

- a phrase-based-style, incremental decoding algorithm for tree-to-string translation
- polynomial-time in theory, linear-time in practice
- 30 times faster than phrase-based Moses

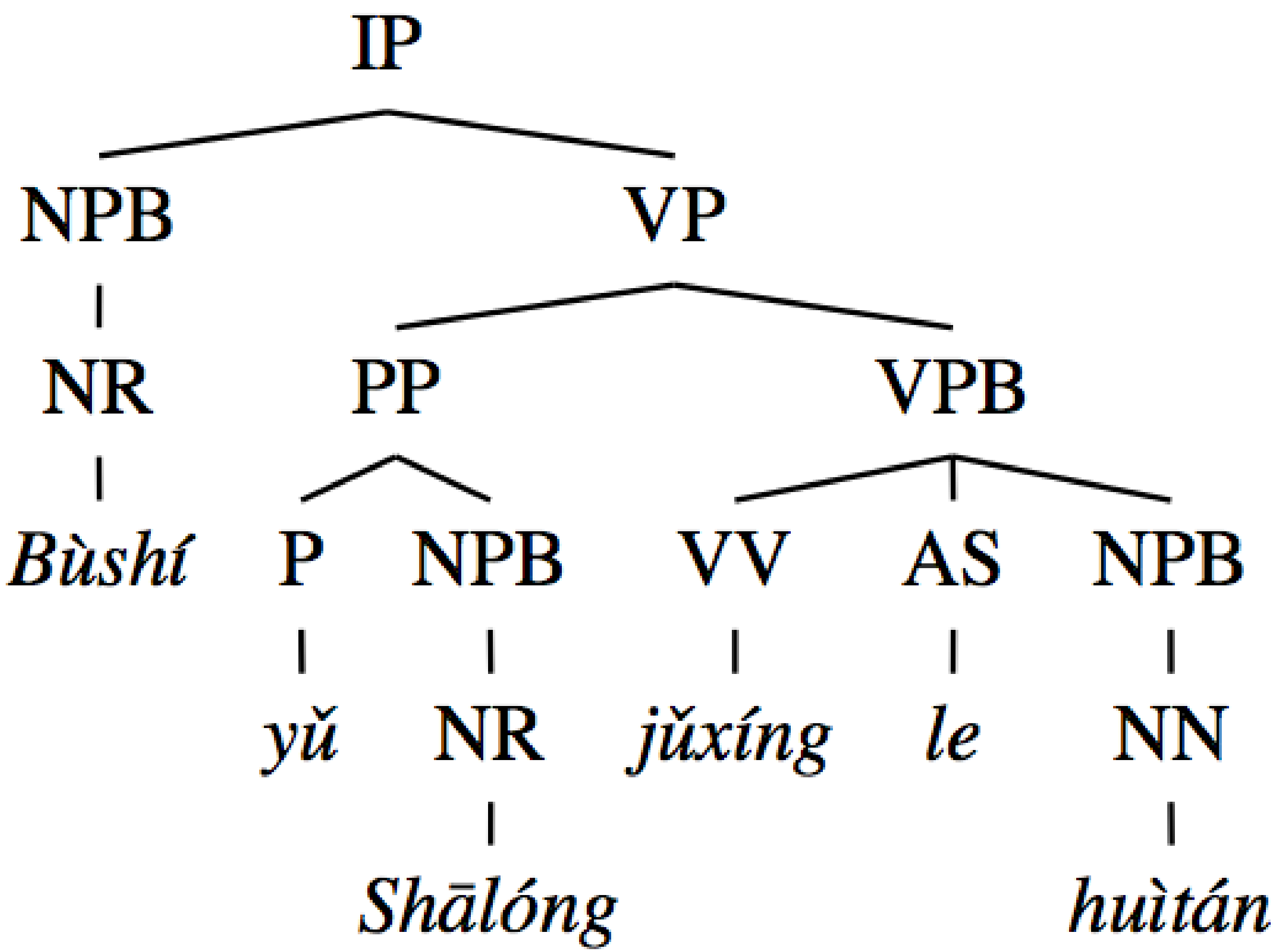
	BLEU	*time*
Moses (in C++)	29.4	10.8s
tree-to-string (in Python)	29.5	0.3s

Outline

- Background: Tree-to-String Translation
- Background: Phrase-based Decoding
- Incremental Decoding for Tree-to-String Translation
- Complexity Analysis
- Experiments

Tree-to-String Translation

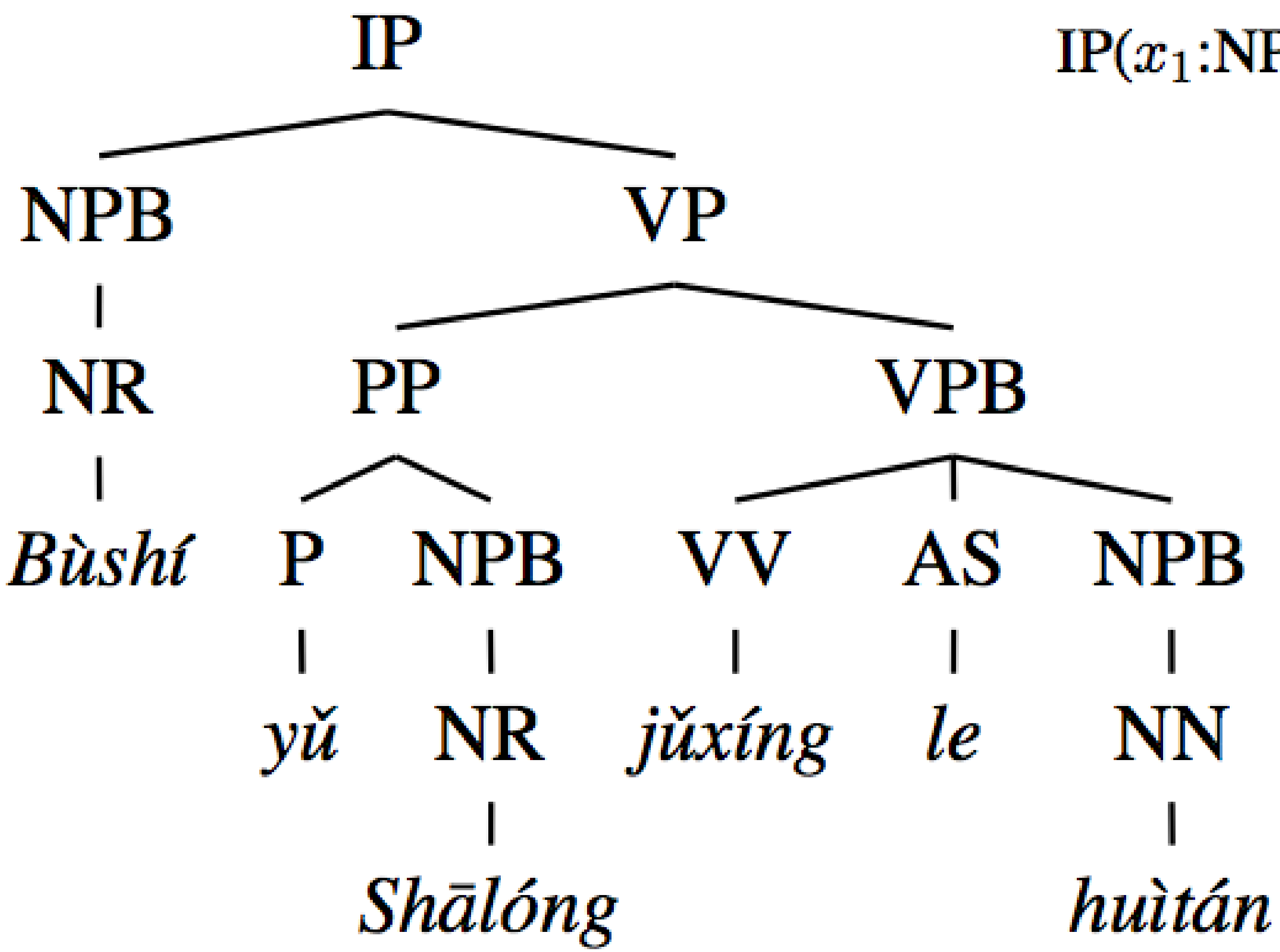
- get 1-best parse tree; then convert to English



(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)

Tree-to-String Translation

- get 1-best parse tree; then convert to English

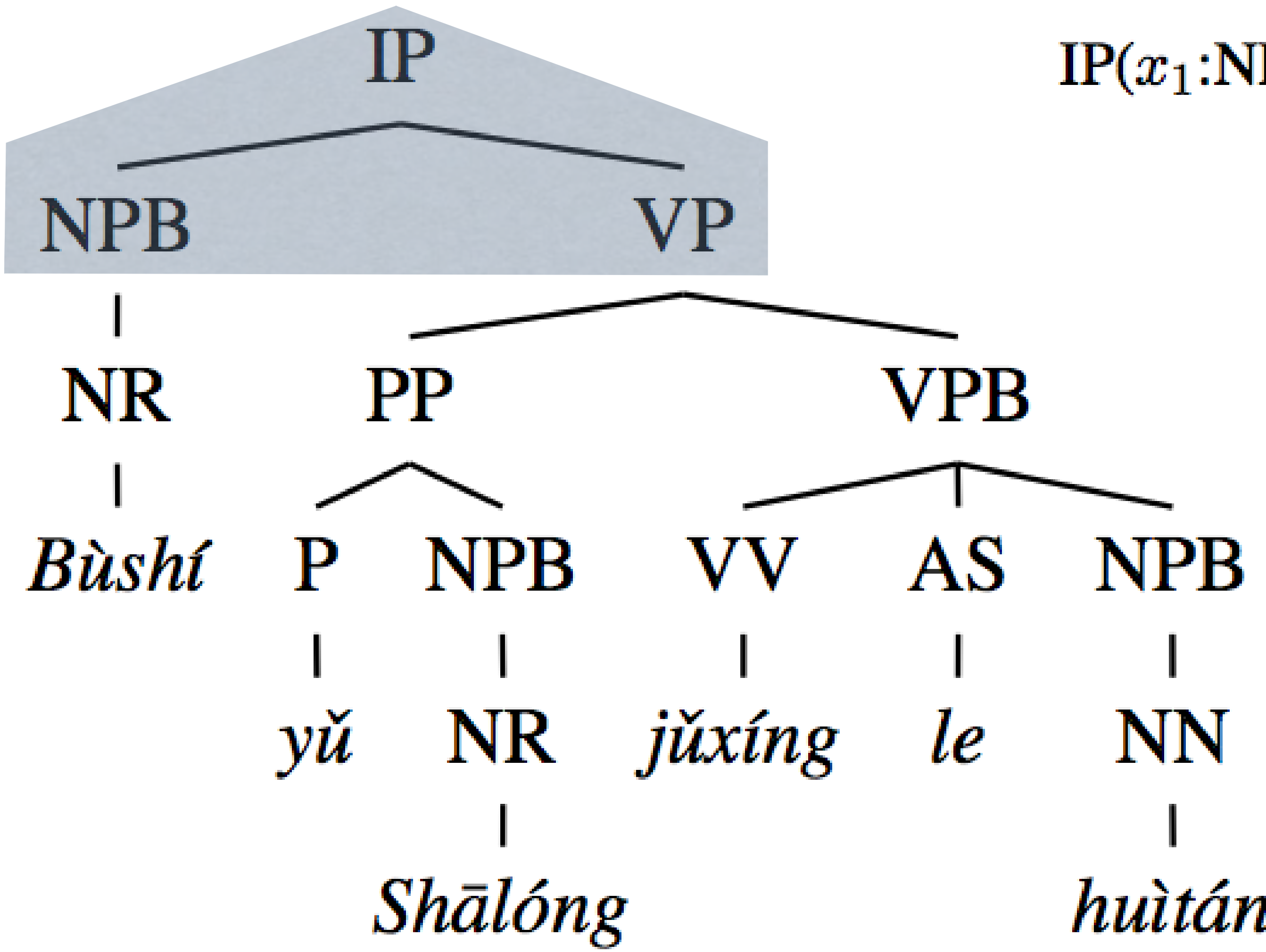


$\text{IP}(x_1\text{:NPB}\ x_2\text{:VP}) \rightarrow x_1\ x_2$

(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)

Tree-to-String Translation

- get 1-best parse tree; then convert to English

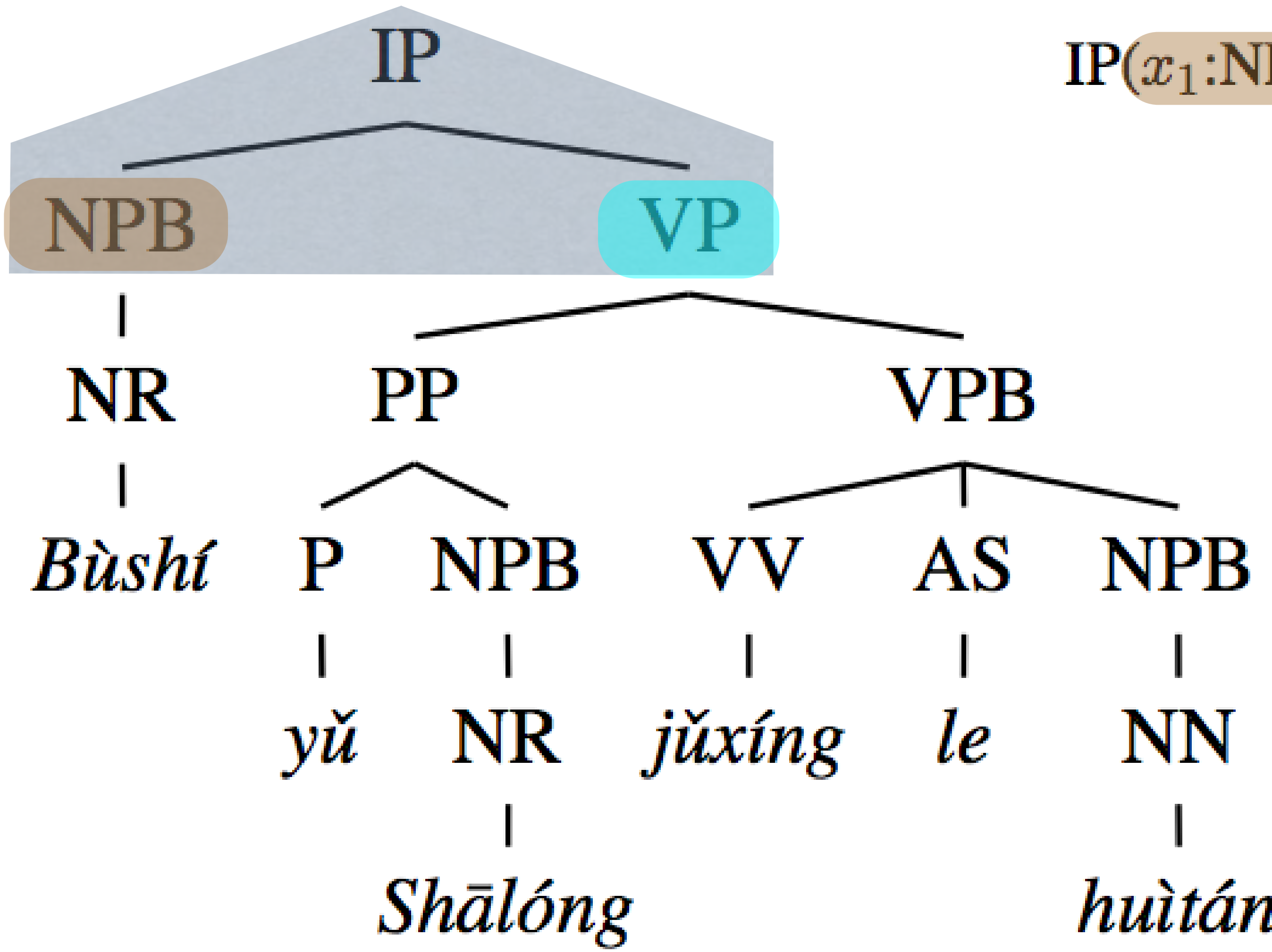


$\text{IP}(x_1\text{:NPB } x_2\text{:VP}) \rightarrow x_1 \ x_2$

(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)

Tree-to-String Translation

- get 1-best parse tree; then convert to English

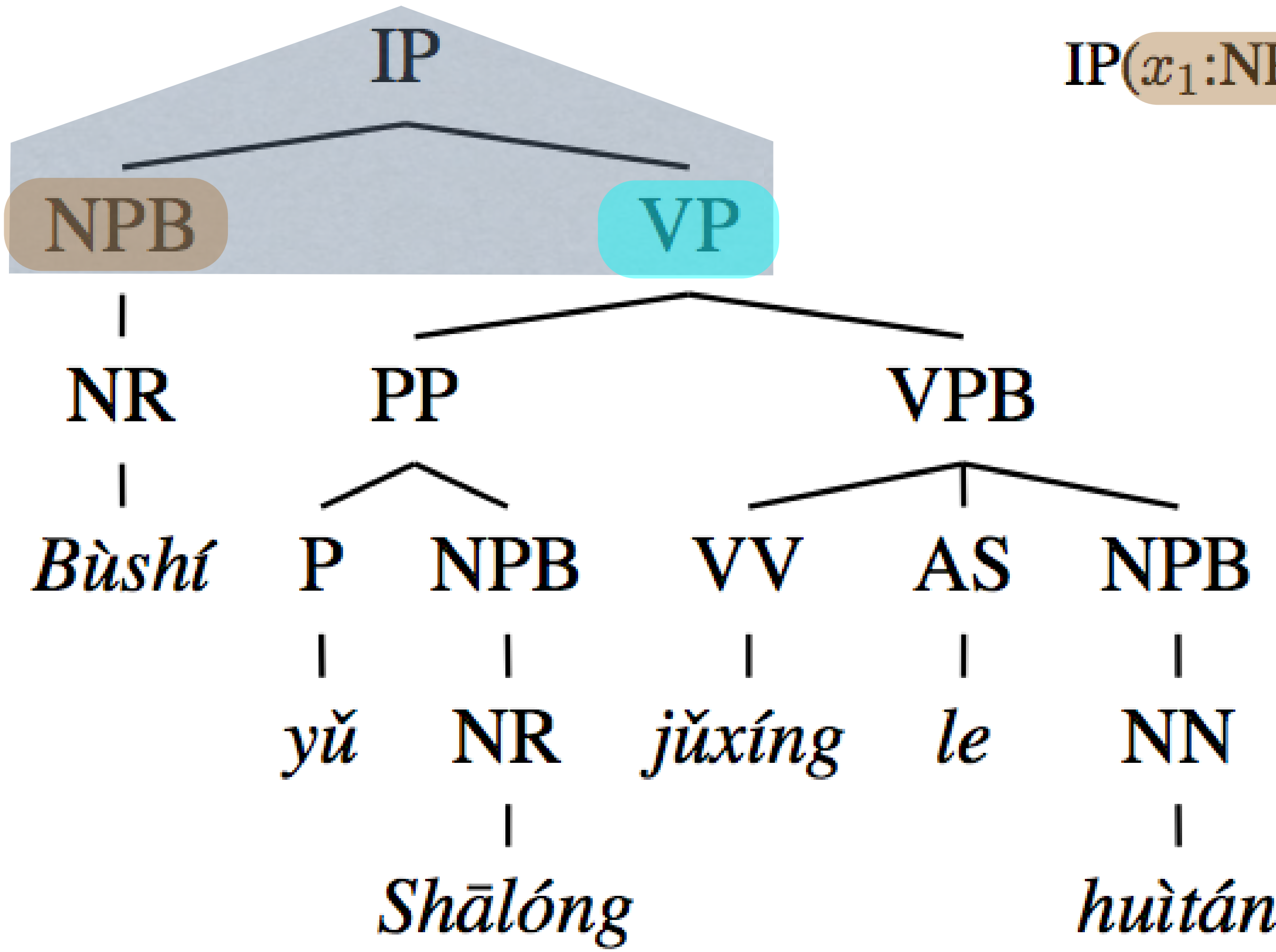


$\text{IP}(x_1\text{:NPB}\ x_2\text{:VP}) \rightarrow x_1\ x_2$

(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)

Tree-to-String Translation

- get 1-best parse tree; then convert to English

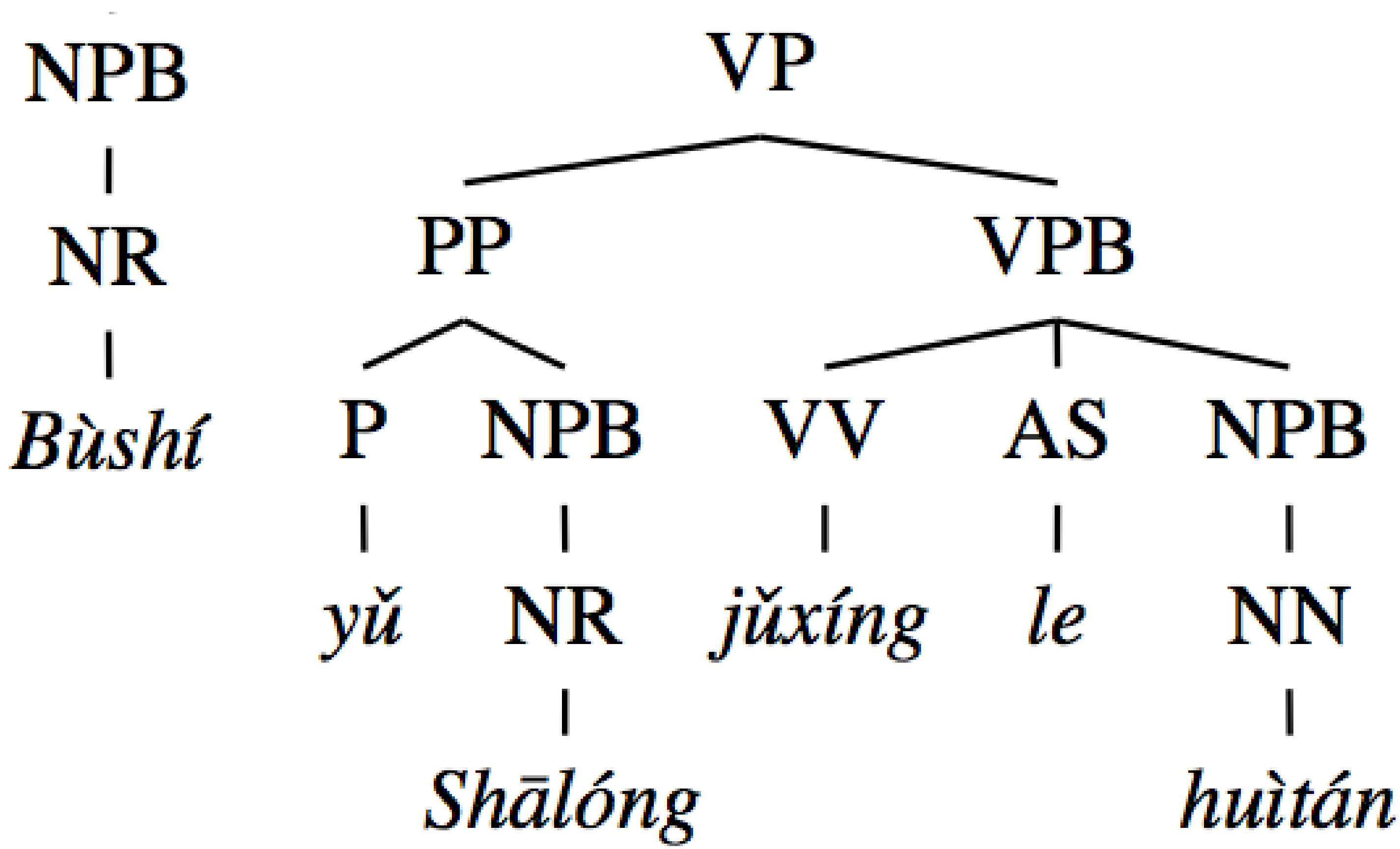


$\mathrm{IP}(x_1{:}\mathrm{NPB}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$

(Galley et al., 2004; Liu et al., 2006; Huang et al., 2006)

Tree-to-String Translation

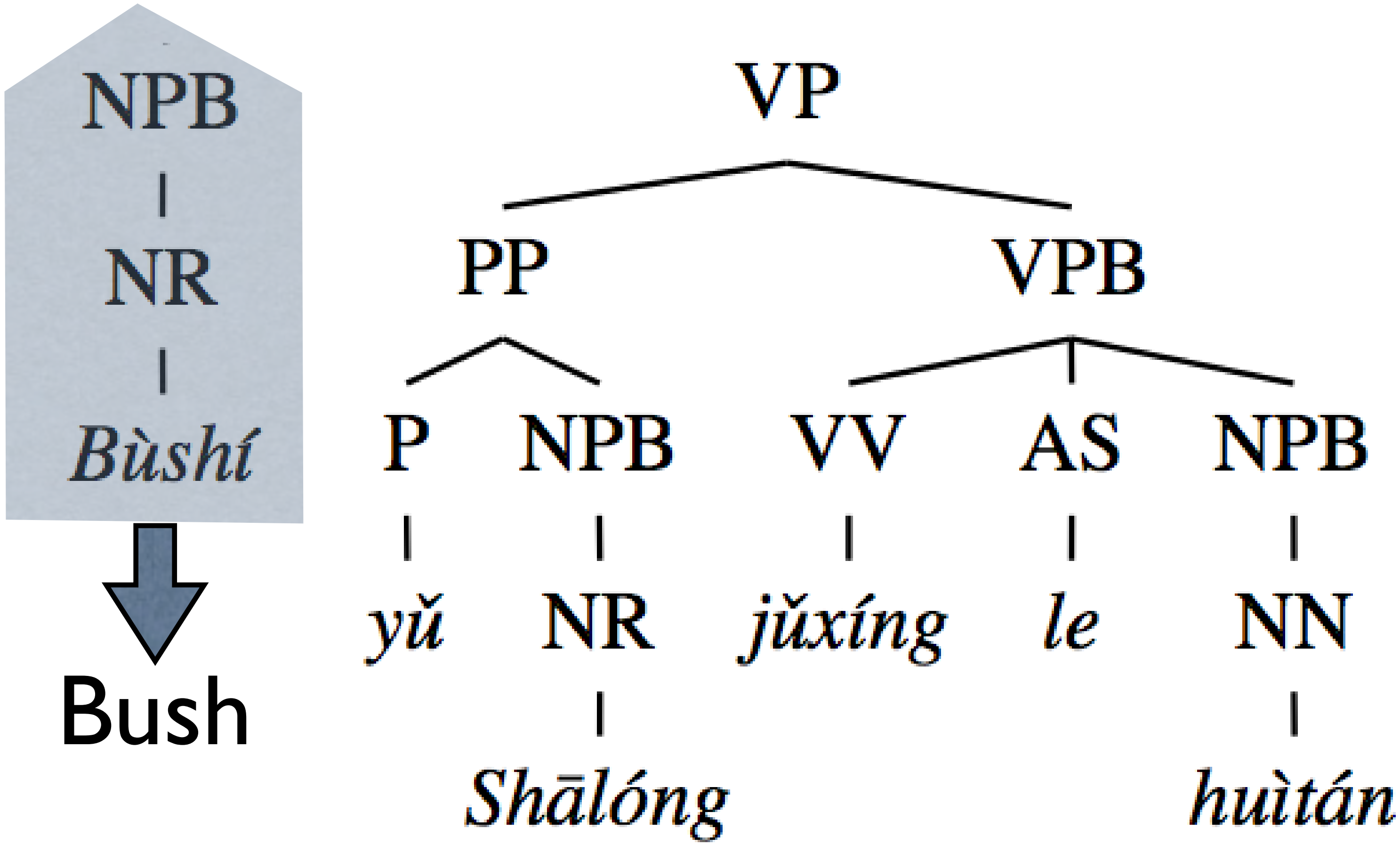
- recursively solve unfinished subproblems



(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- recursively solve unfinished subproblems

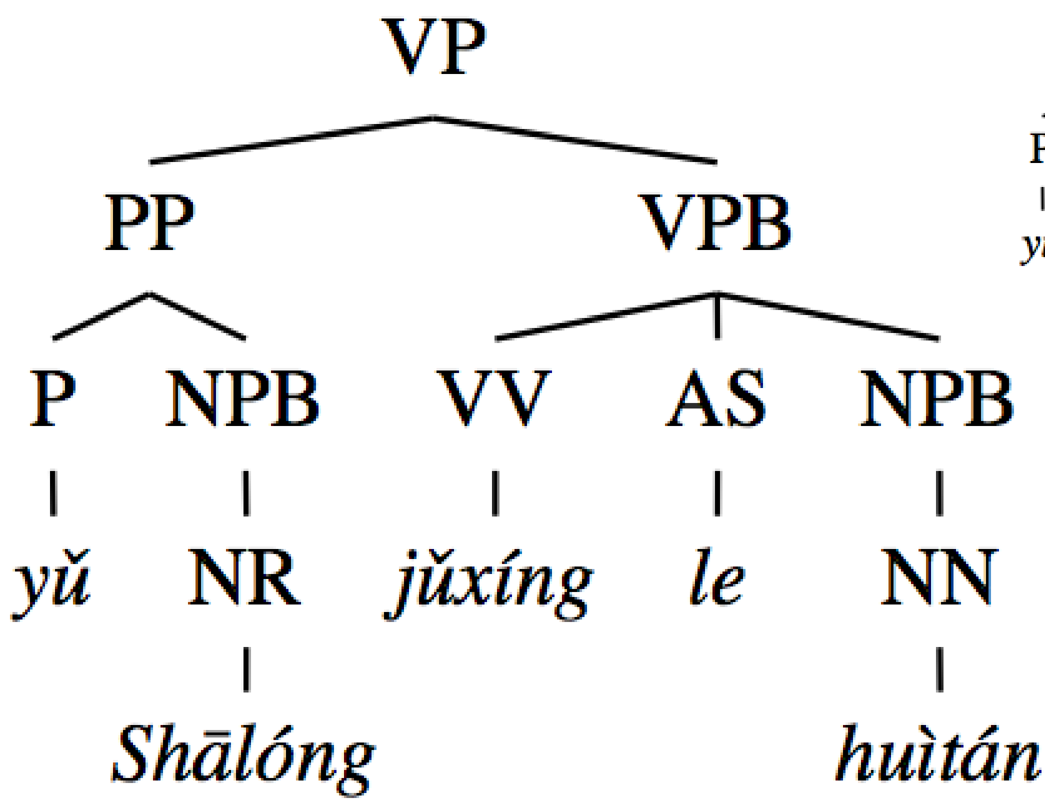


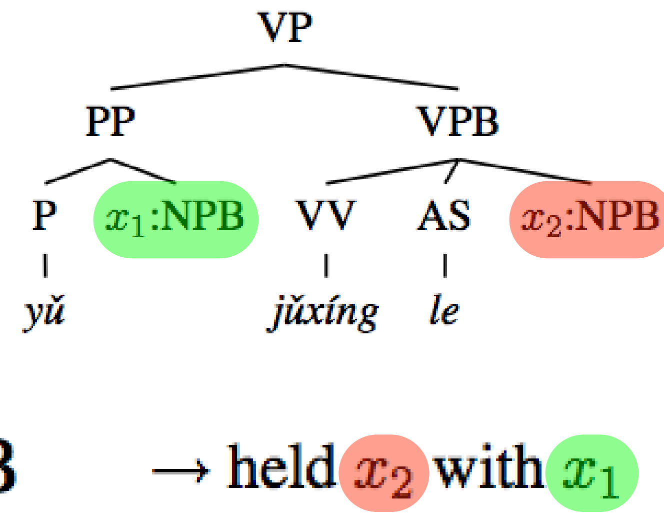
(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- pattern-match tree-to-string translation rules

Bush



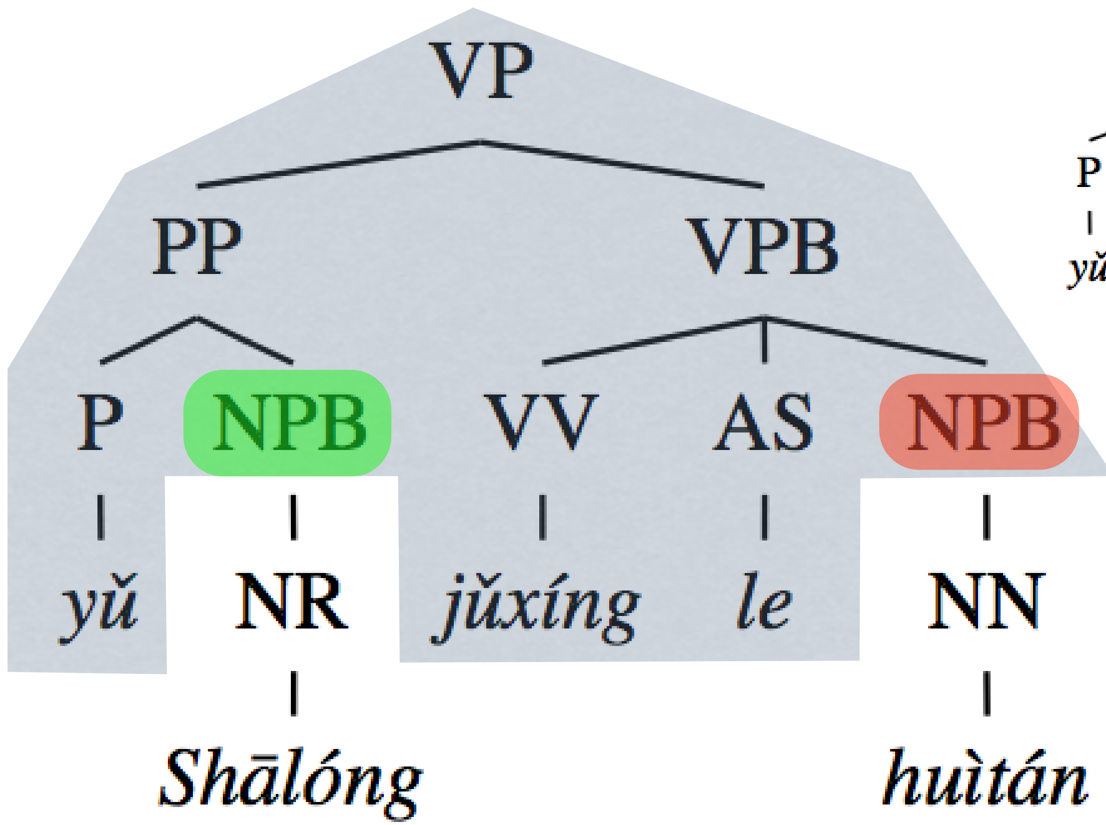


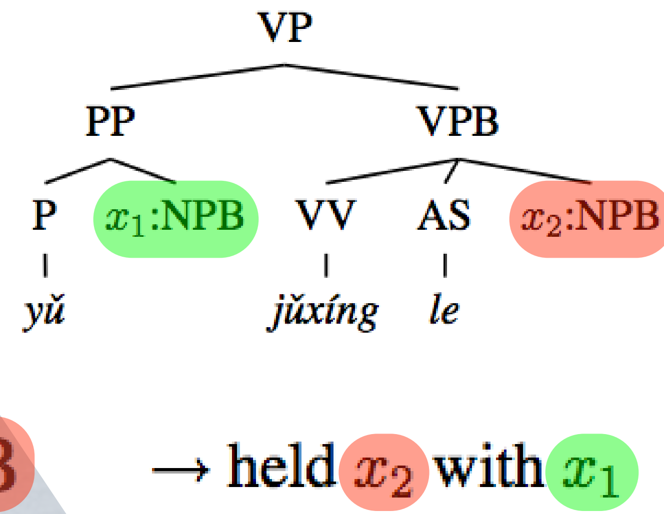
(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- pattern-match tree-to-string translation rules

Bush

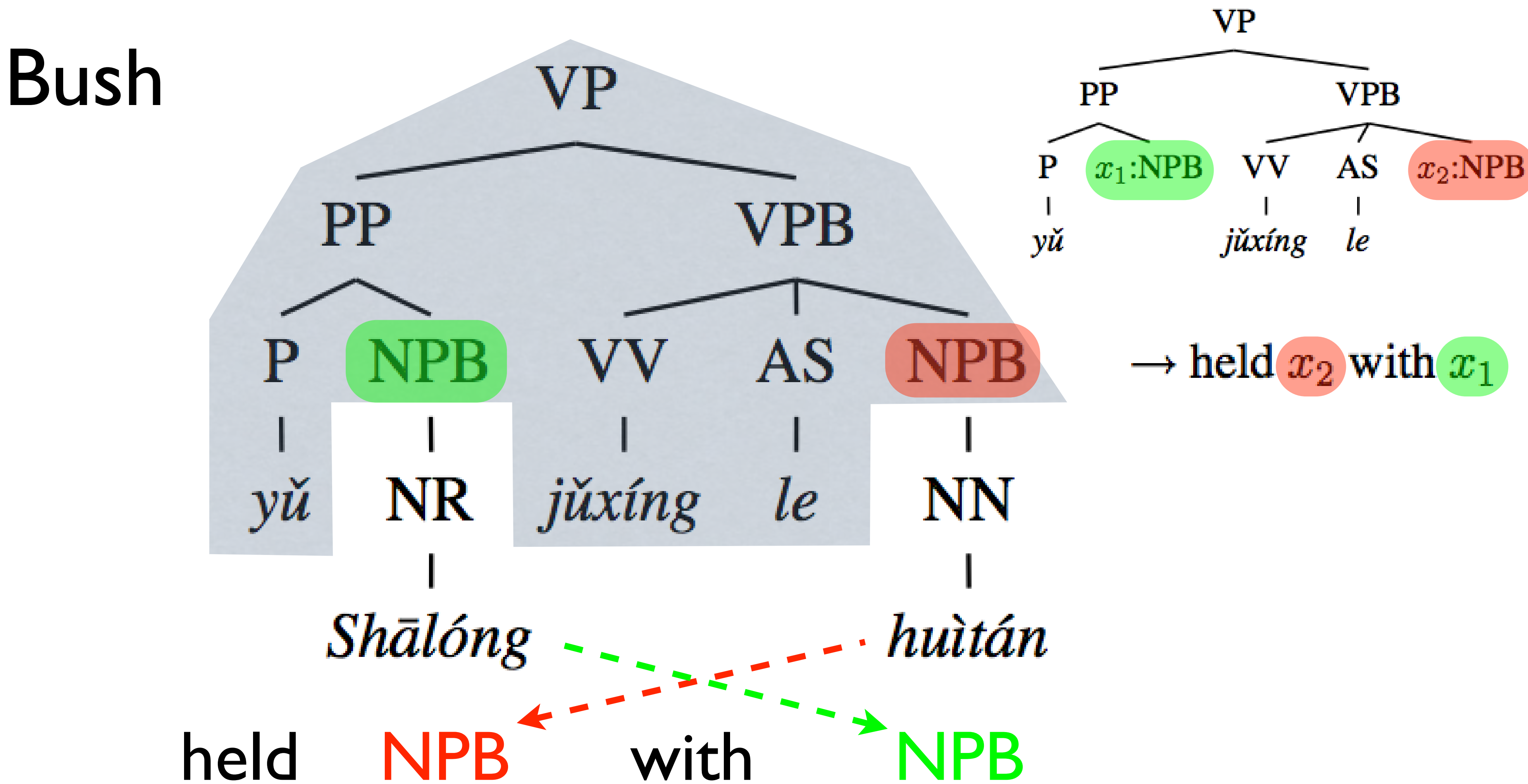




(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- pattern-match tree-to-string translation rules



(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- continue pattern-matching

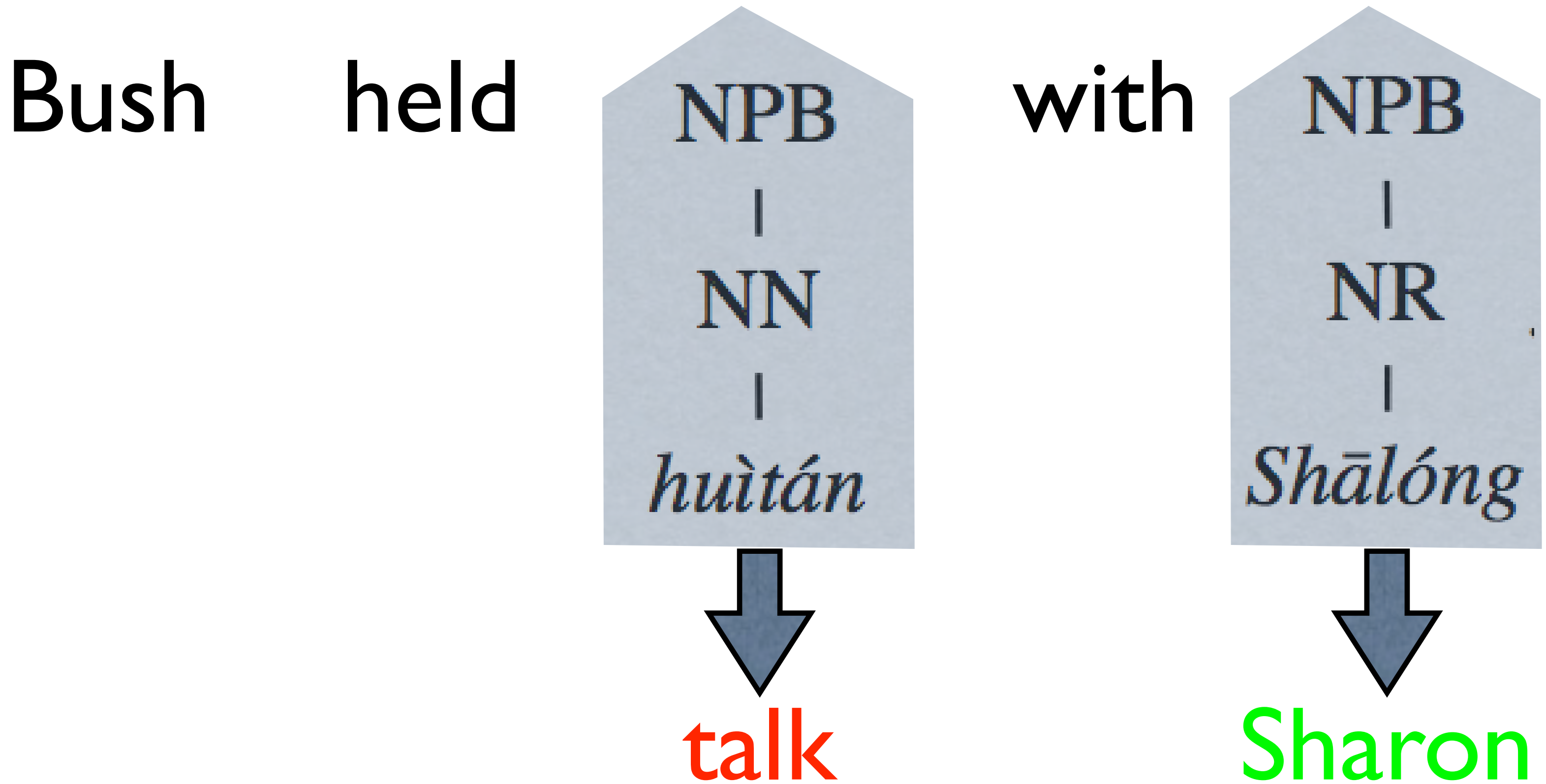
Bush held NPB with NPB

NPB
|
NN
|
huìtán

NPB
|
NR
|
Shālóng

Tree-to-String Translation

- continue pattern-matching



(Liu et al 06; Huang et al 06)

Tree-to-String Translation

- continue pattern-matching

Bush held a talk with Sharon

(Galley et al 04; Liu et al 06; Huang et al 06)

Tree-to-String Translation

- continue pattern-matching

Bush held a talk with Sharon

really simple! and fast: $O(n)$-time decoding!

(Galley et al 04; Liu et al 06; Huang et al 06)

Tree-to-String Translation

- continue pattern-matching

Bush held a talk with Sharon

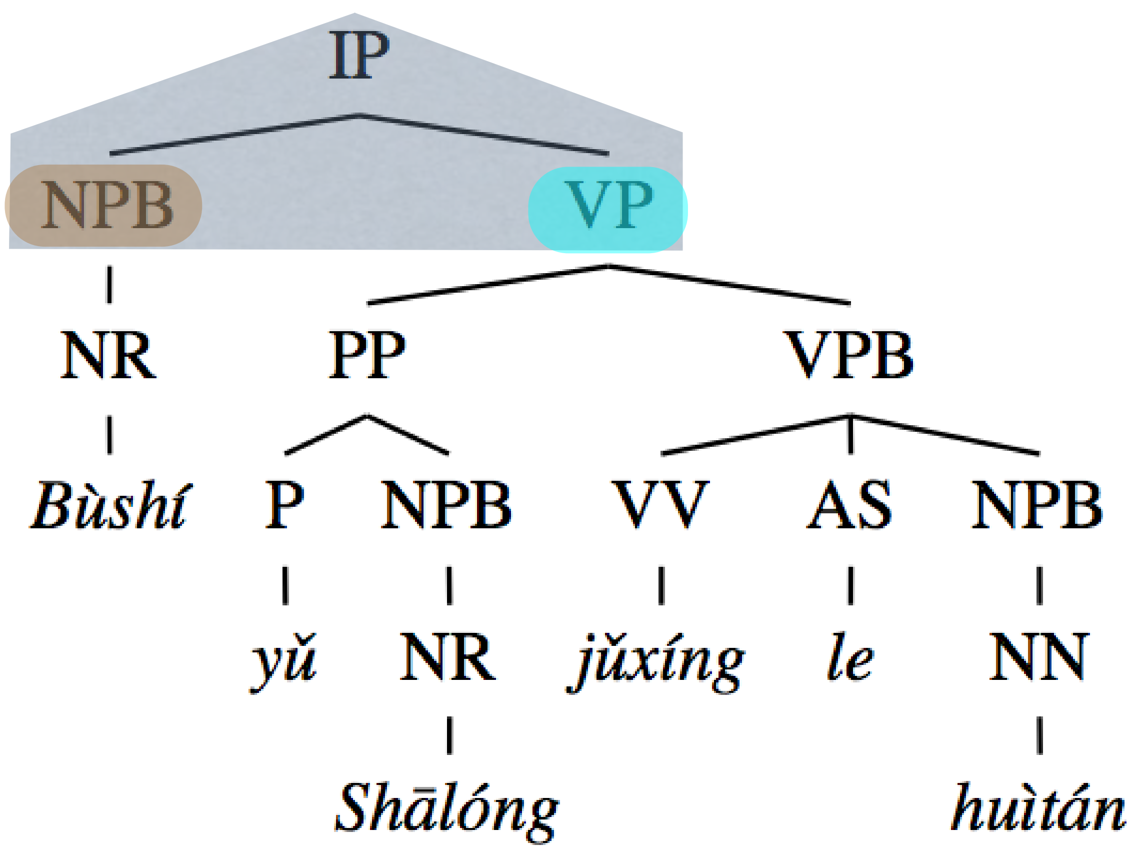
really simple! and fast: $O(n)$-time decoding!

but with language model, it becomes slower...

(Galley et al 04; Liu et al 06; Huang et al 06)

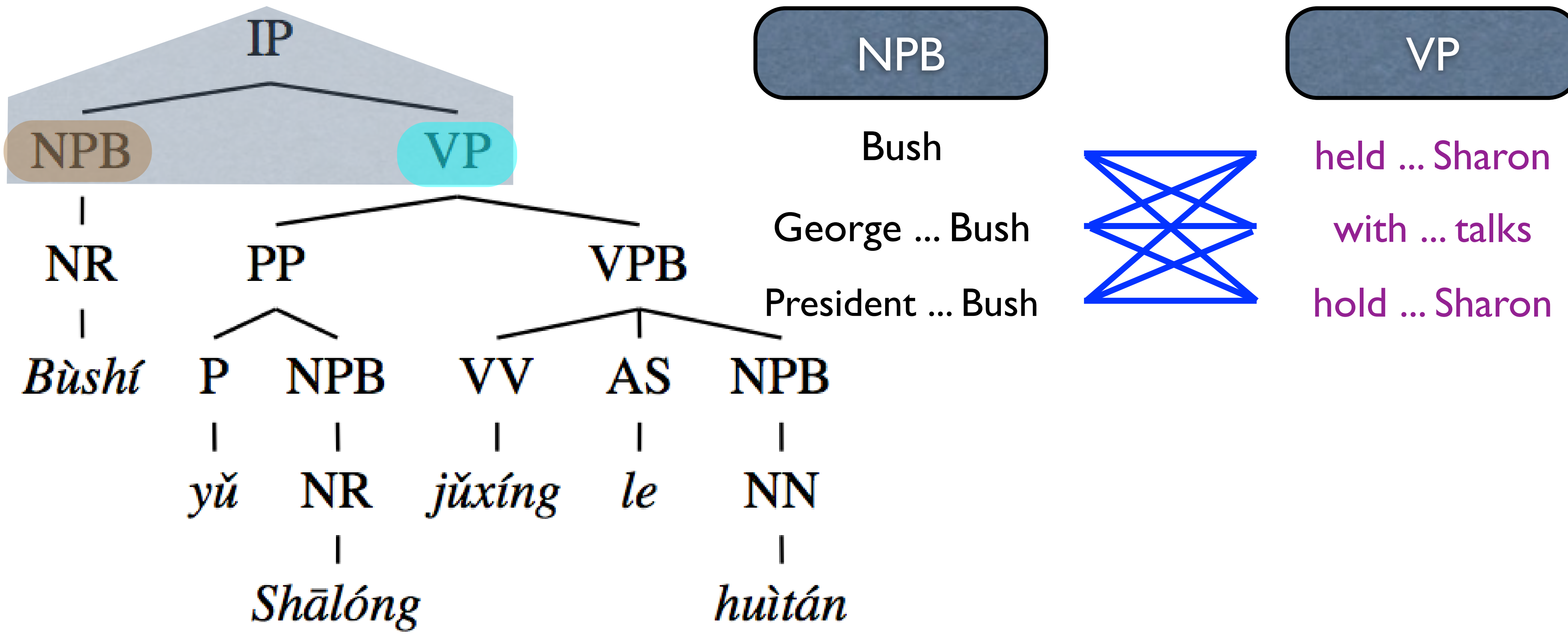
Decoding w/ Language Model

- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product



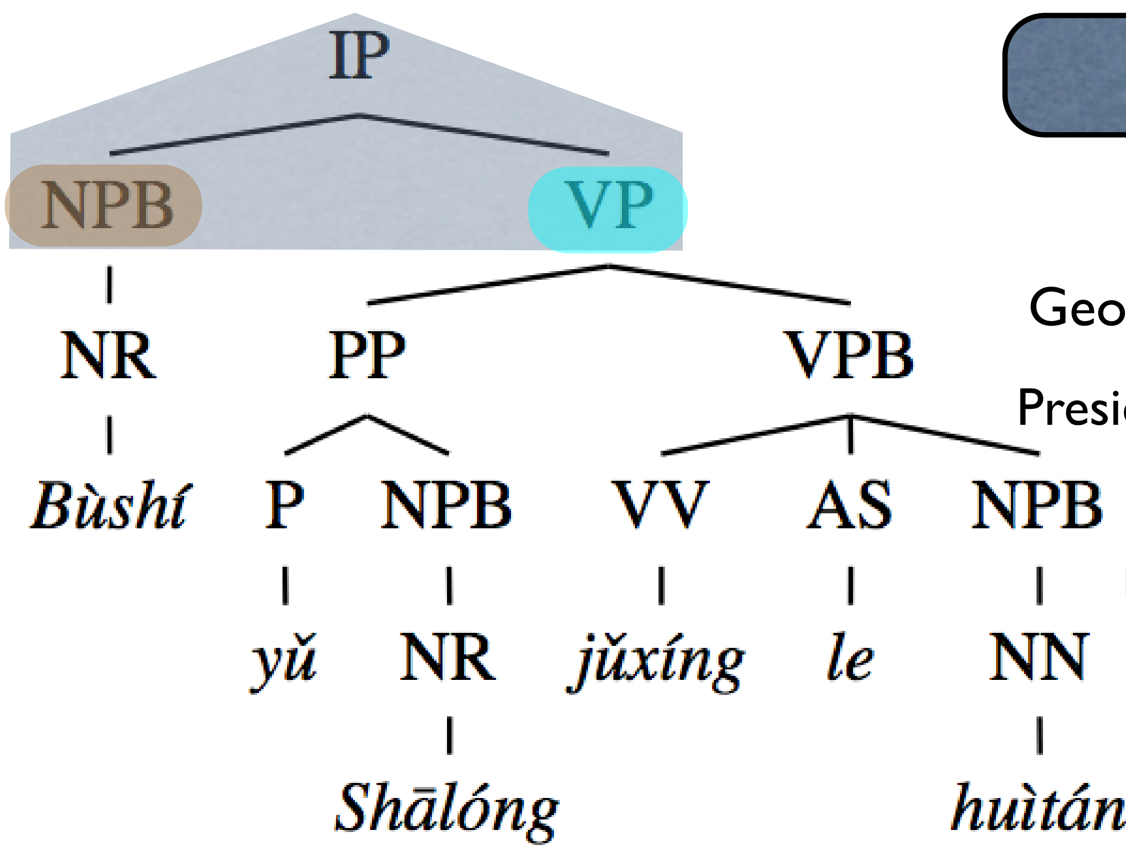
Decoding w/ Language Model

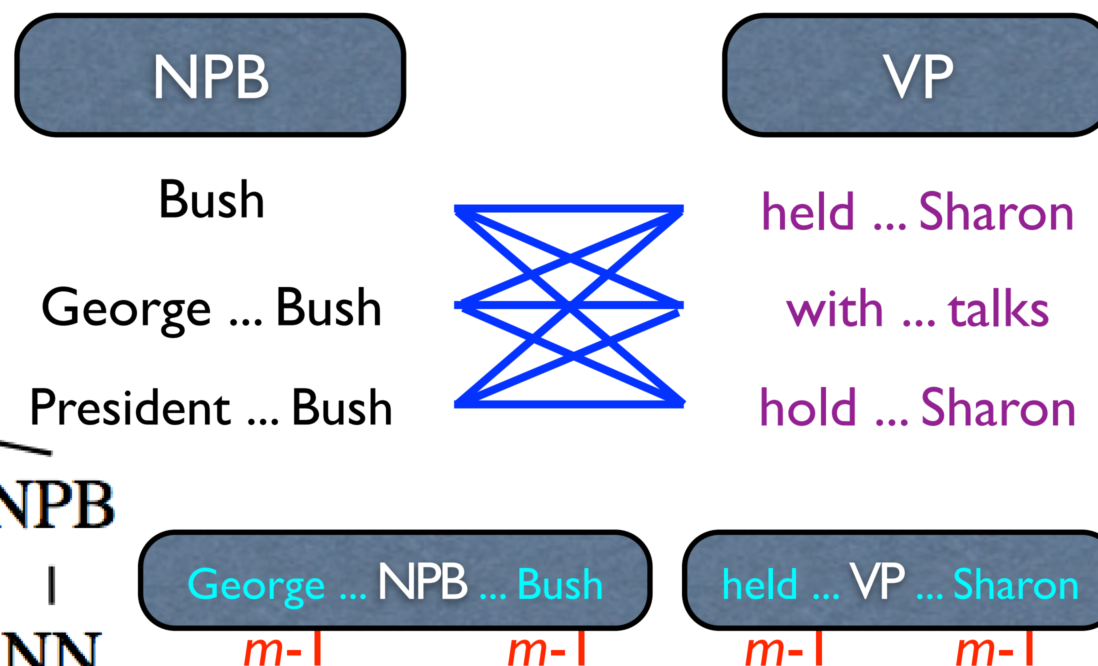
- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product



Decoding w/ Language Model

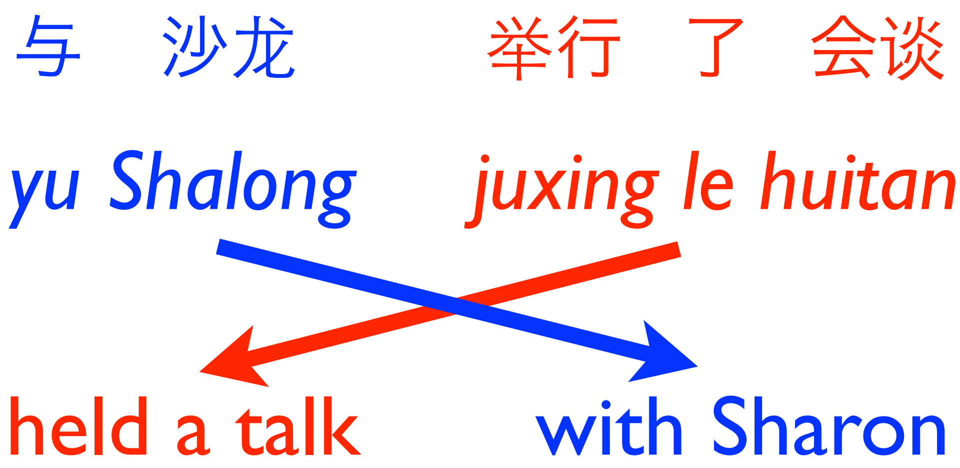
- bottom-up (equivalent to top-down)
- each node is now split into several +LM nodes
- maintain LM signatures at both ends; and cross-product



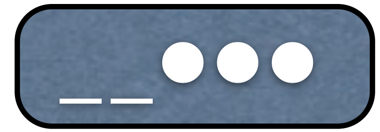


time: $O(n\,V^{4(m-1)})$
for m-gram model

Phrase-based Decoding (-LM)

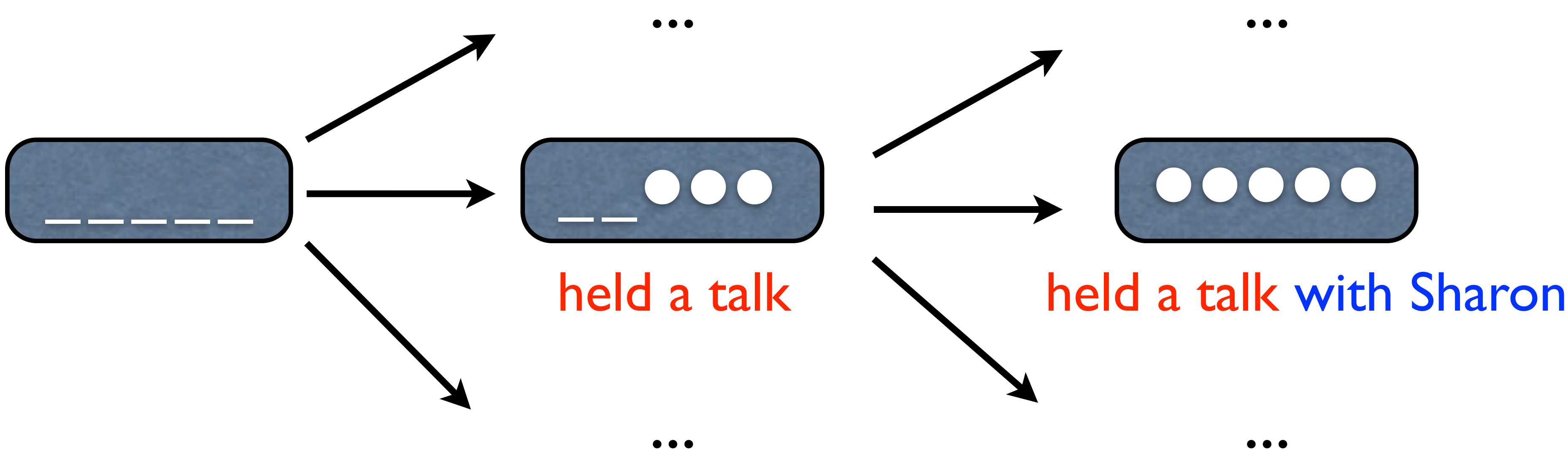


source-side: coverage vector



held a talk

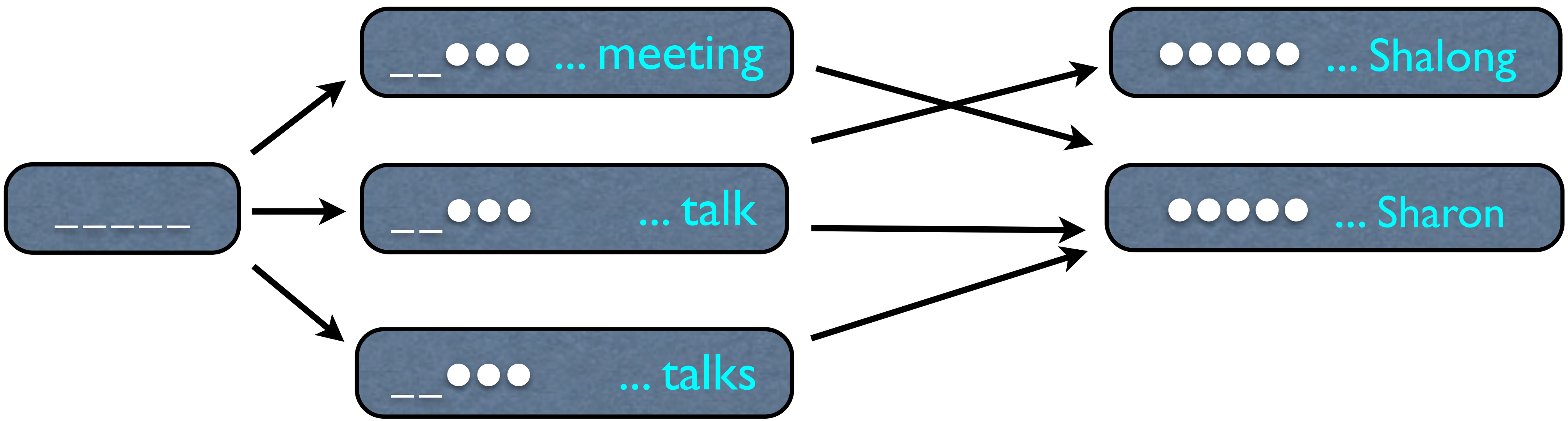
target-side: grow hypotheses
strictly left-to-right



time complexity: $O(2^n n^2)$ -- cf. traveling salesman problem (TSP)

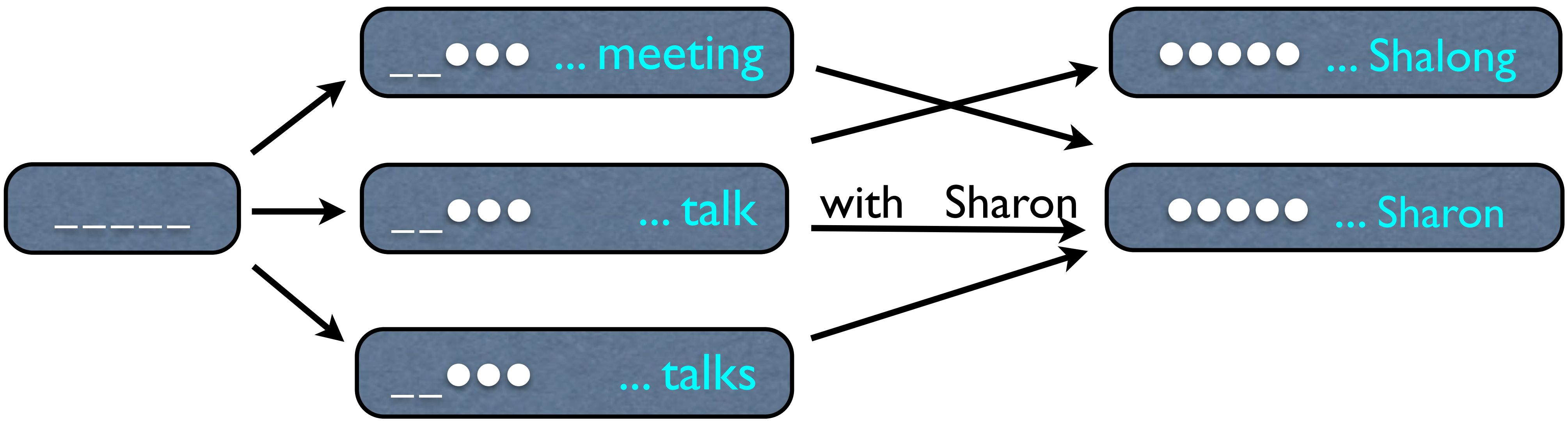
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space



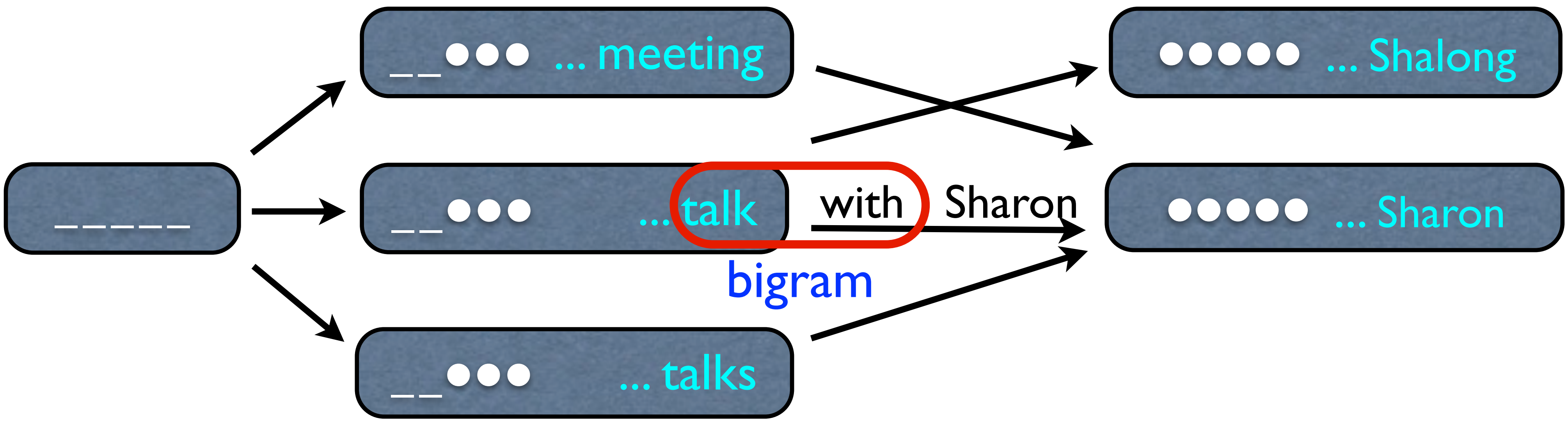
Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space



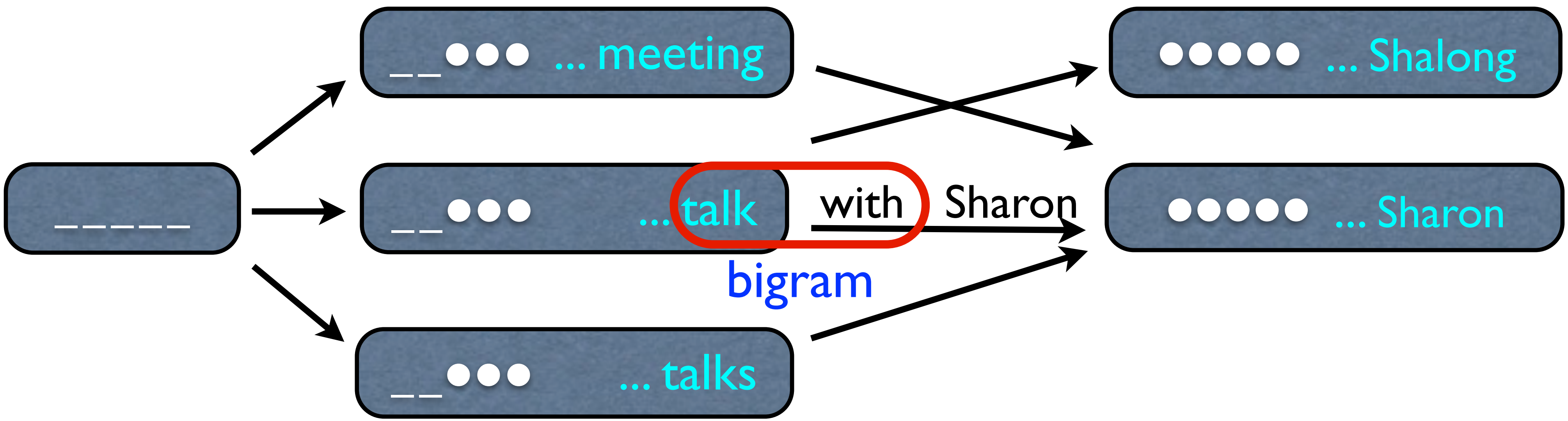
Phrase-based Decoding (+LM)

- "refined" graph: annotated with language model words
- still dynamic programming, just larger search space



Phrase-based Decoding (+LM)

- “refined” graph: annotated with language model words
- still dynamic programming, just larger search space



time: $O(2^n n^2)$ => $O(2^n n^2 V^m)$

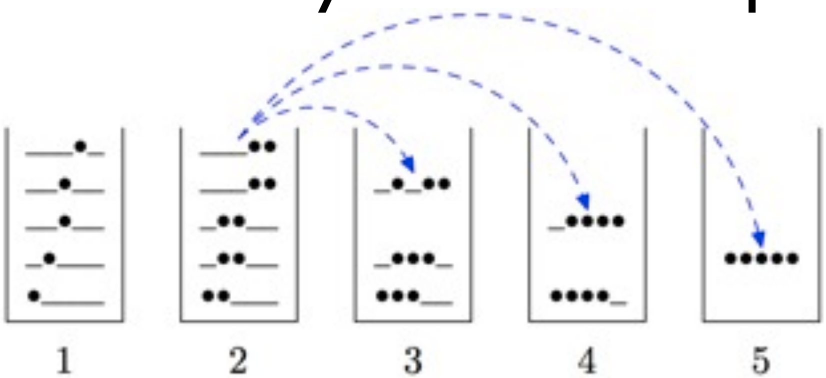
for *m*-gram language models

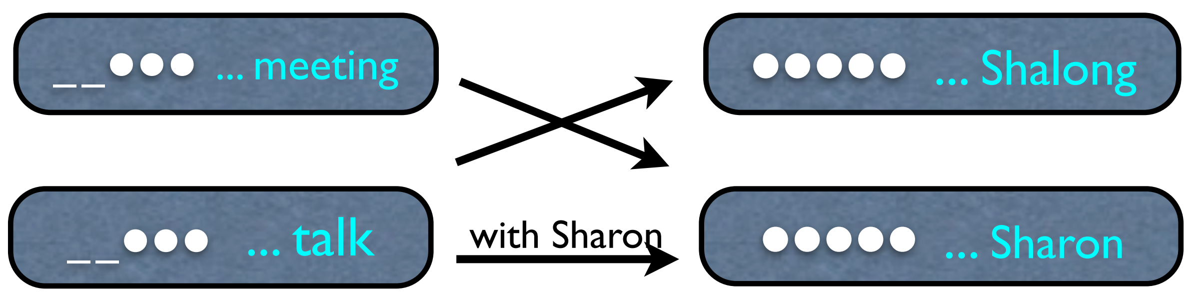
Why Phrase-based is Fast?

- phrase-based is exponential-time in theory
- in practice, linear-time w/ beam search + distortion limit
- key difference due to incremental expansion:
  - only need to keep rightmost LM words

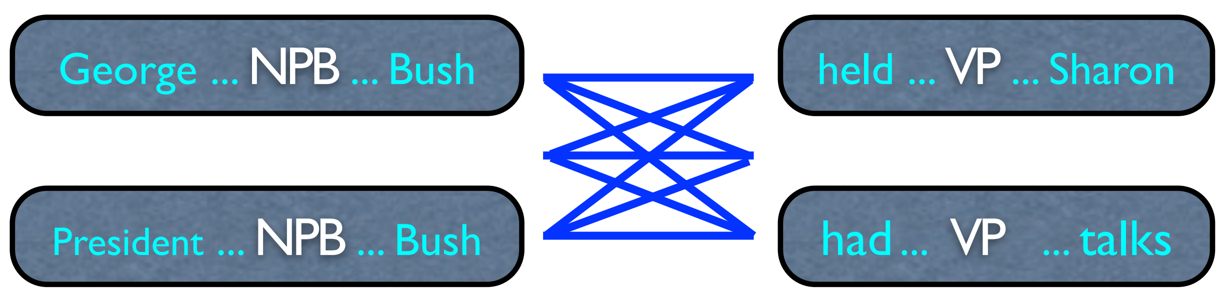
Why Phrase-based is Fast?

- phrase-based is exponential-time in theory
- in practice, linear-time w/ beam search + distortion limit
- key difference due to incremental expansion:
    - only need to keep rightmost LM words





Q: can tree-to-string also become incremental?

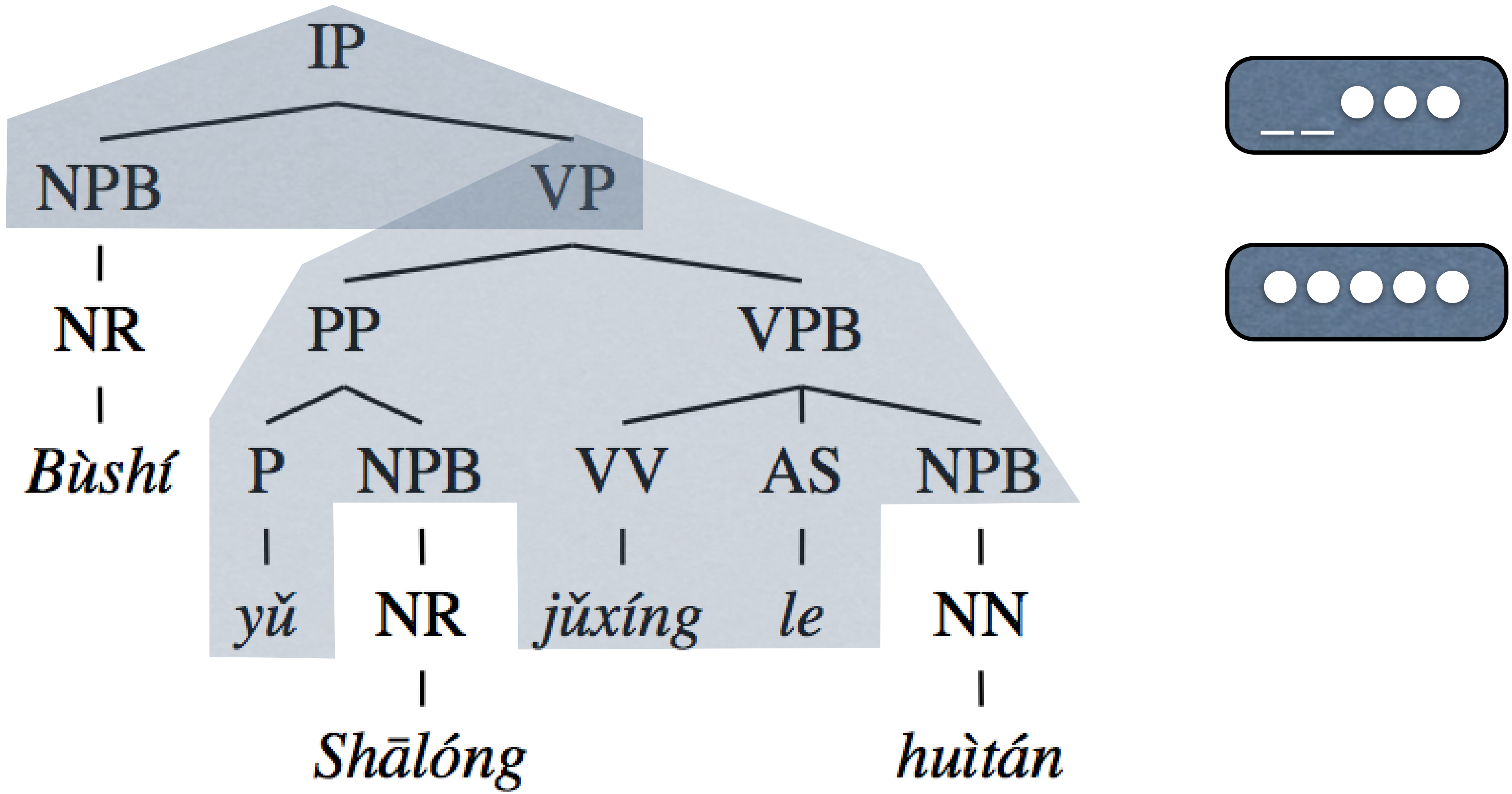


Outline

- Background: Tree-to-String Translation
- Background: Phrase-based Decoding
- Incremental Decoding for Tree-to-String Translation
- Complexity Analysis
- Experiments

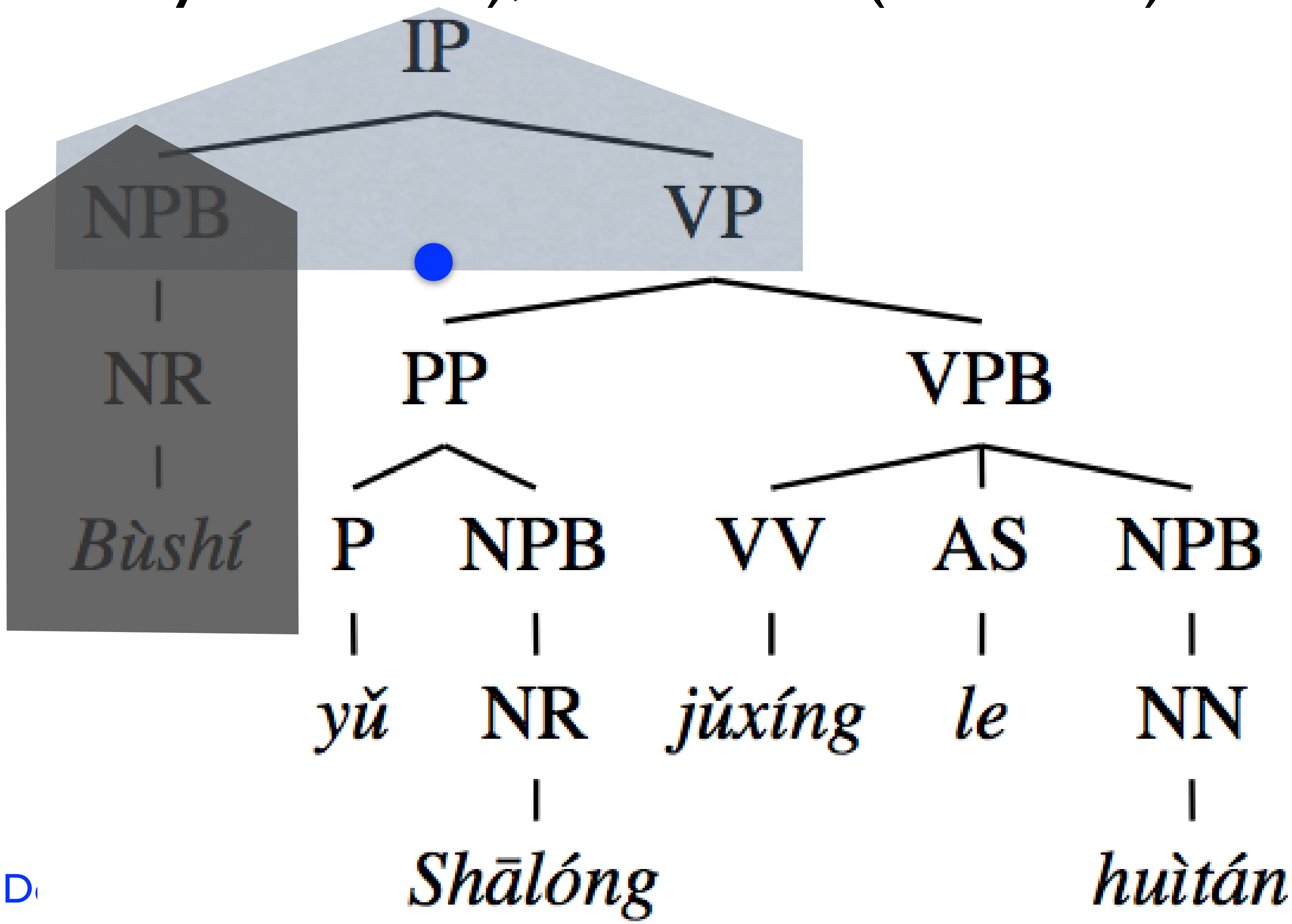
Incremental for Tree-to-String

- key intuition: tree coverage-vector?



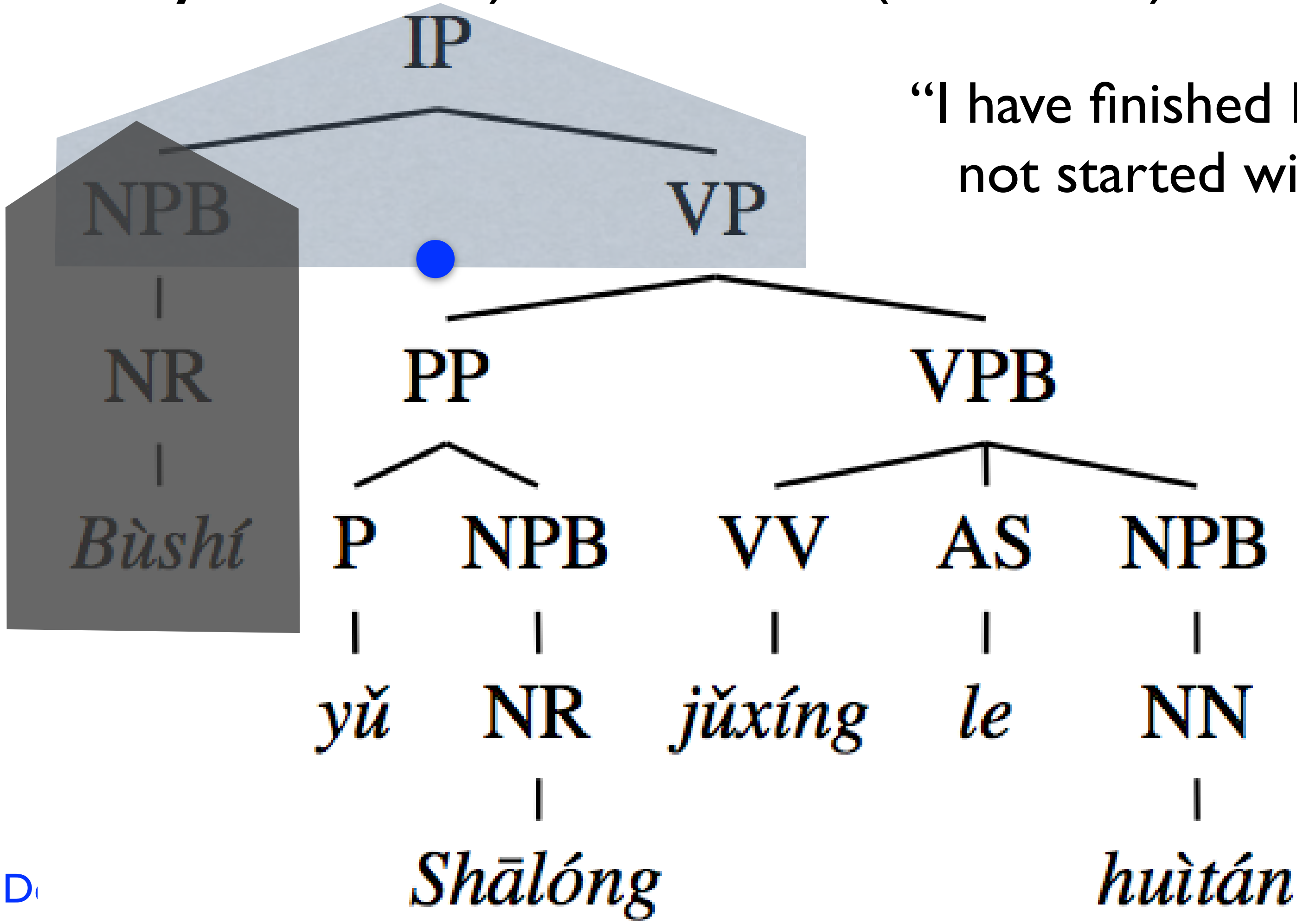
Tree Coverage Vector as Stack

- stack (*active* derivation history): [ε→.IP] [IP→ NPB.VP]
- three colors for nodes: white (uncovered), grey (partially covered), and black (covered)



Tree Coverage Vector as Stack

- stack (*active* derivation history): [ε→.IP] [IP→ NPB.VP]
- three colors for nodes: white (uncovered), grey (partially covered), and black (covered)



"I have finished NPB subtree but not started with VP subtree"

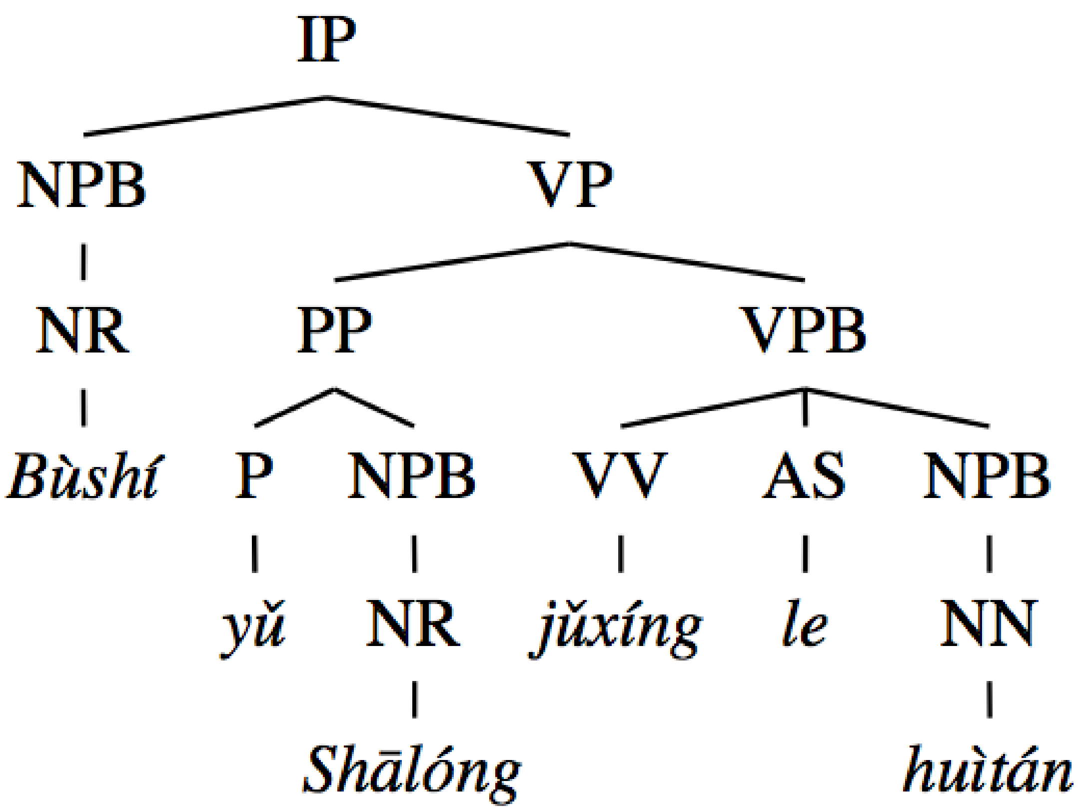
Example Incremental Decoding

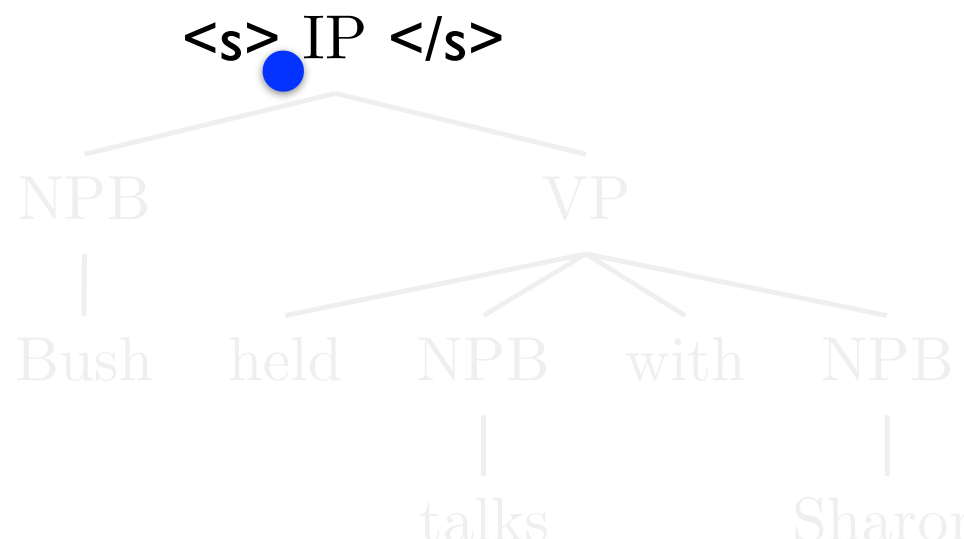
[ε→<s> •IP </s>]

stack

<s>

hypothesis

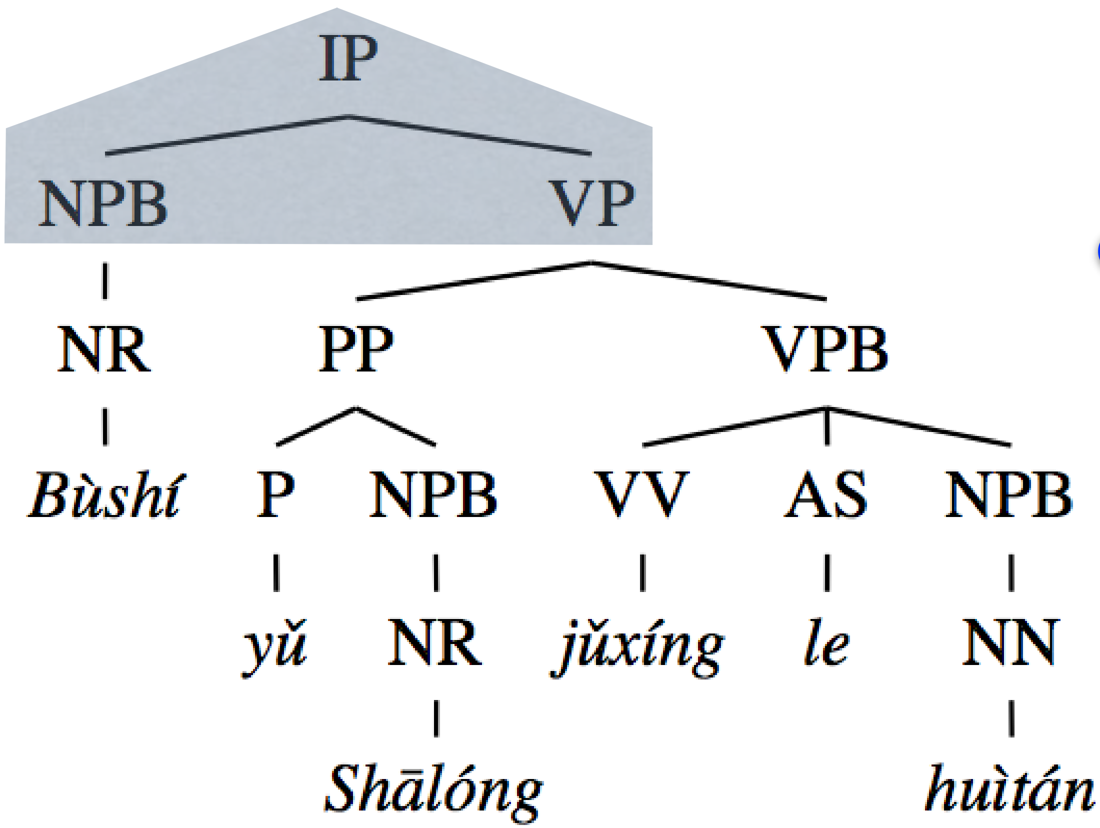


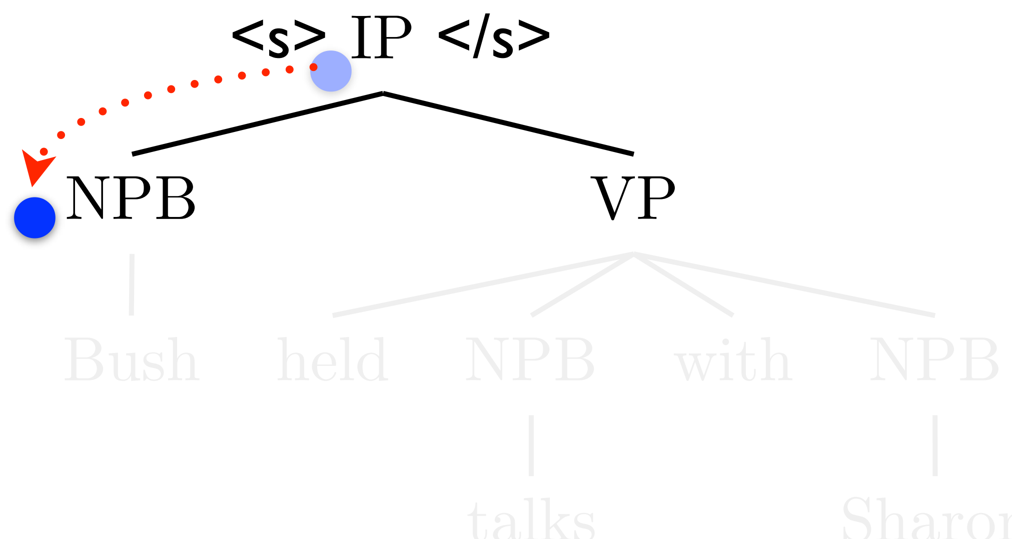


Example Incremental Decoding

[ε→<s> •IP </s>] [IP→•NPB VP] *stack*

<s> *hypothesis*



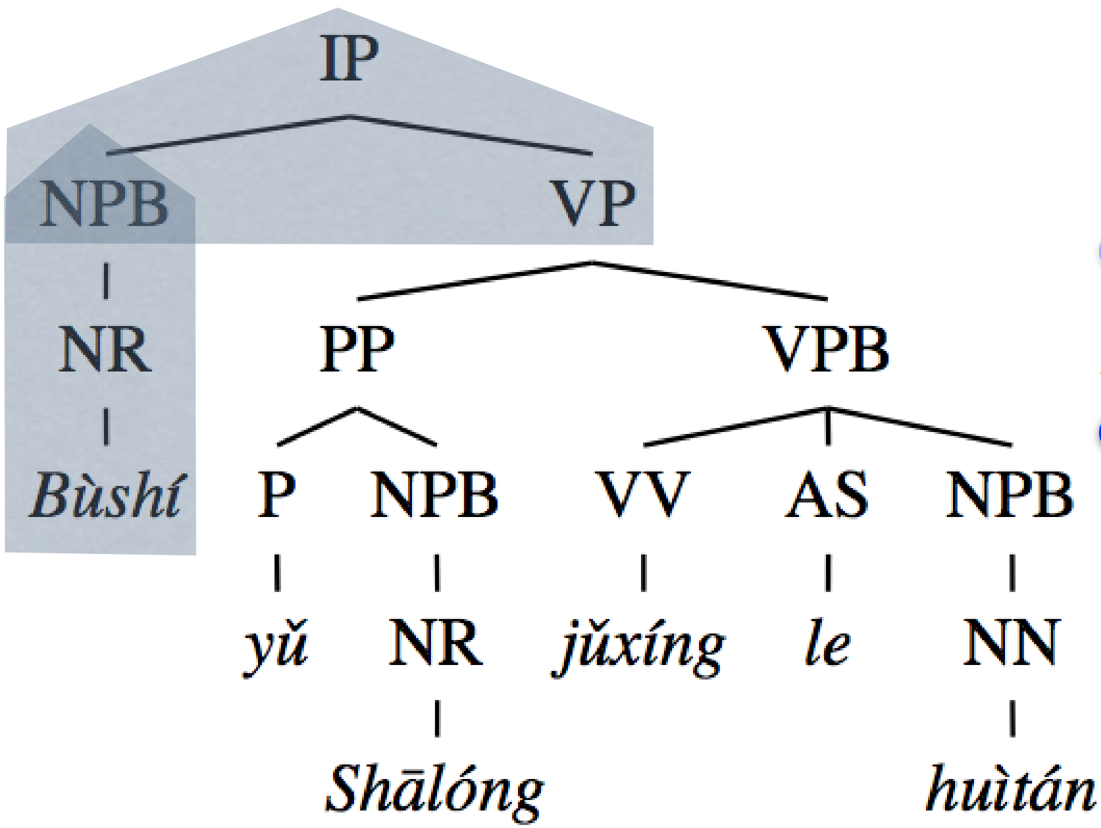


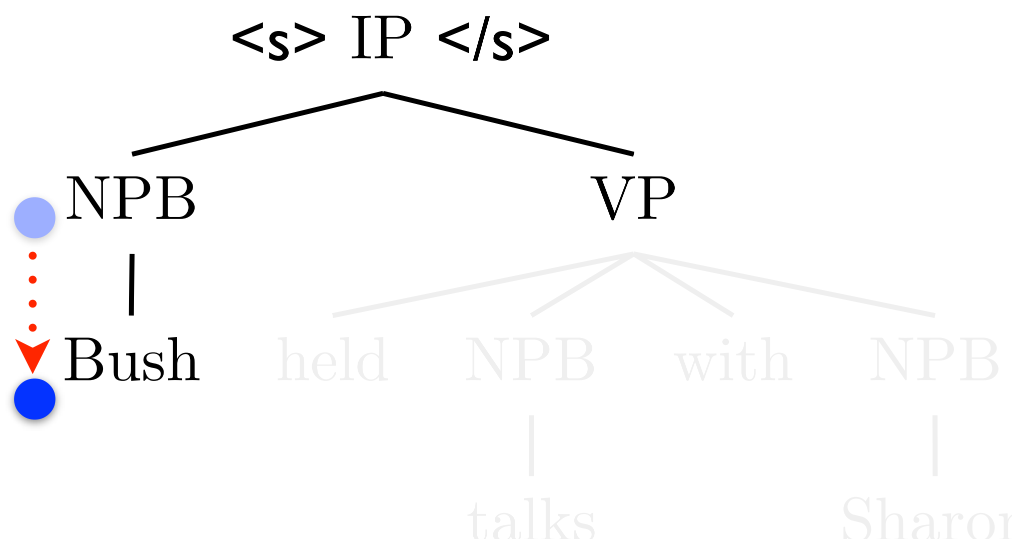
action: predict (push)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→•NPB VP] [NPB→•Bush] *stack*

<s> *hypothesis*



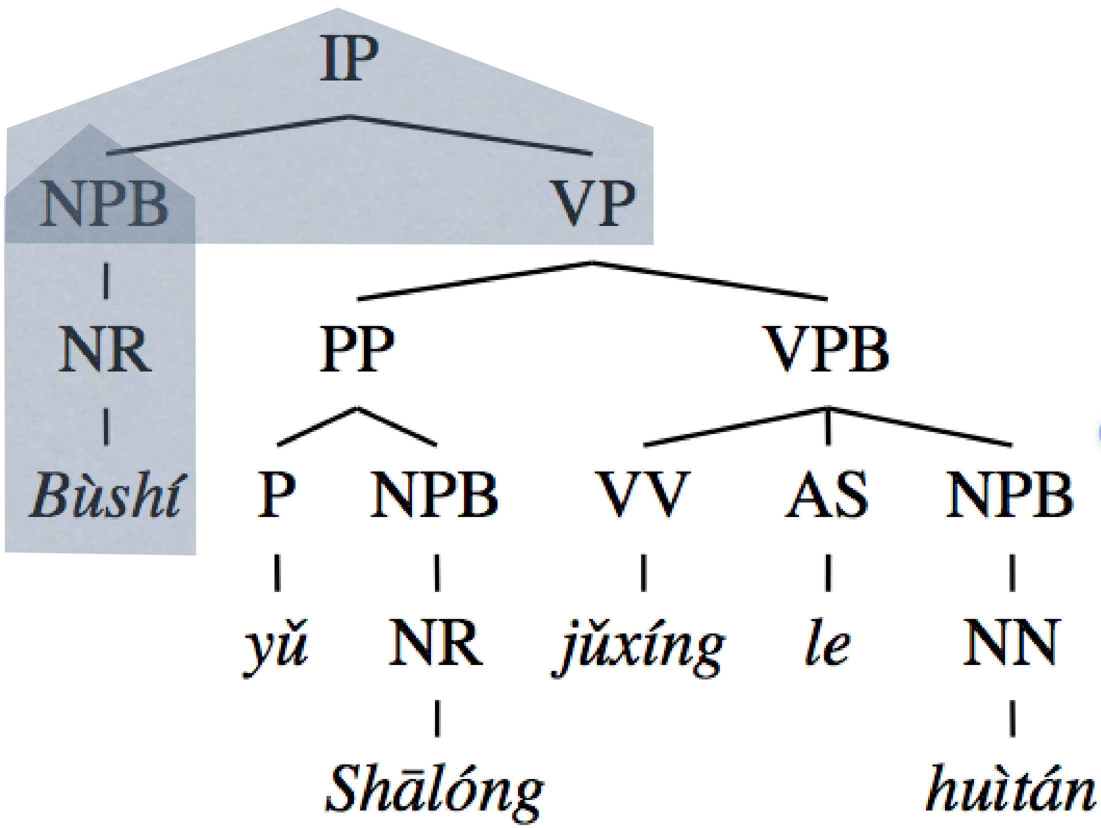


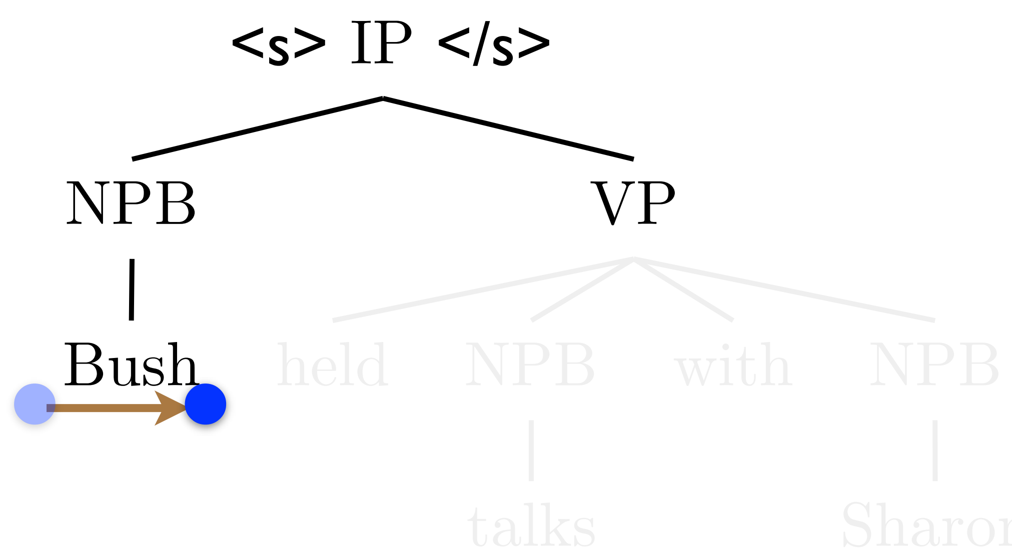
action: predict (push)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→•NPB VP] [NPB→Bush•] *stack*

<s> Bush *hypothesis*



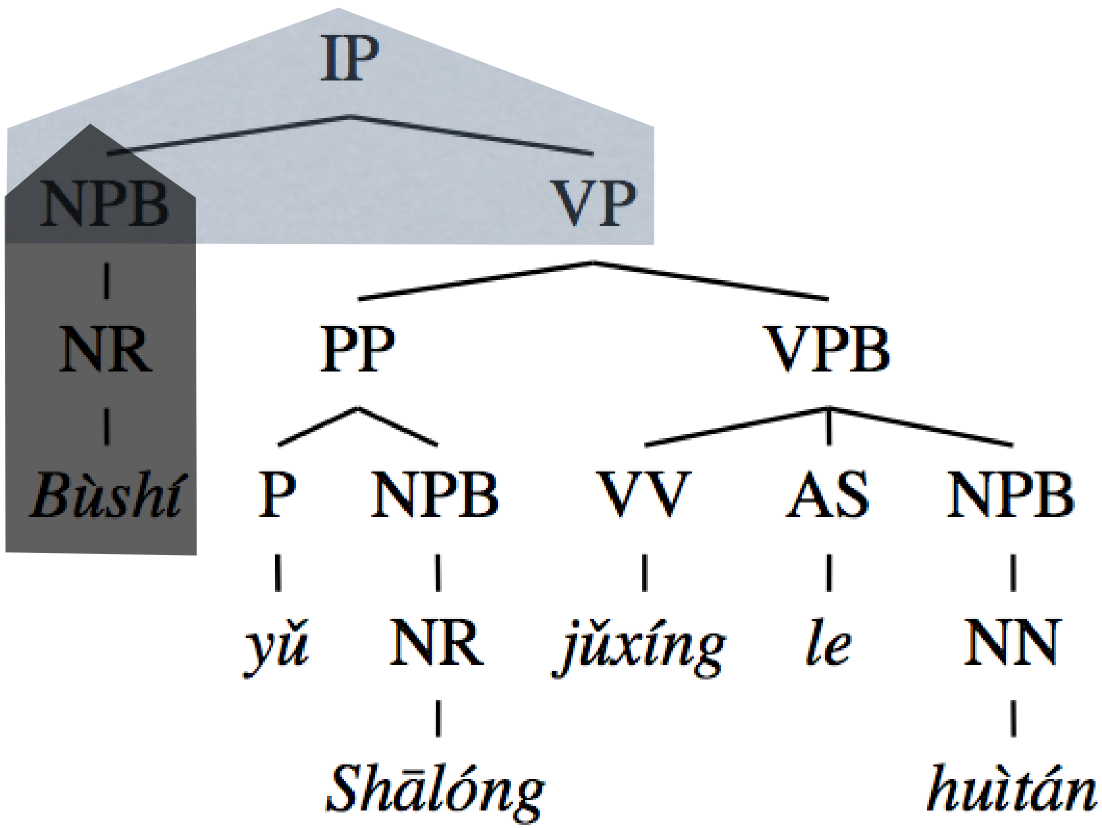


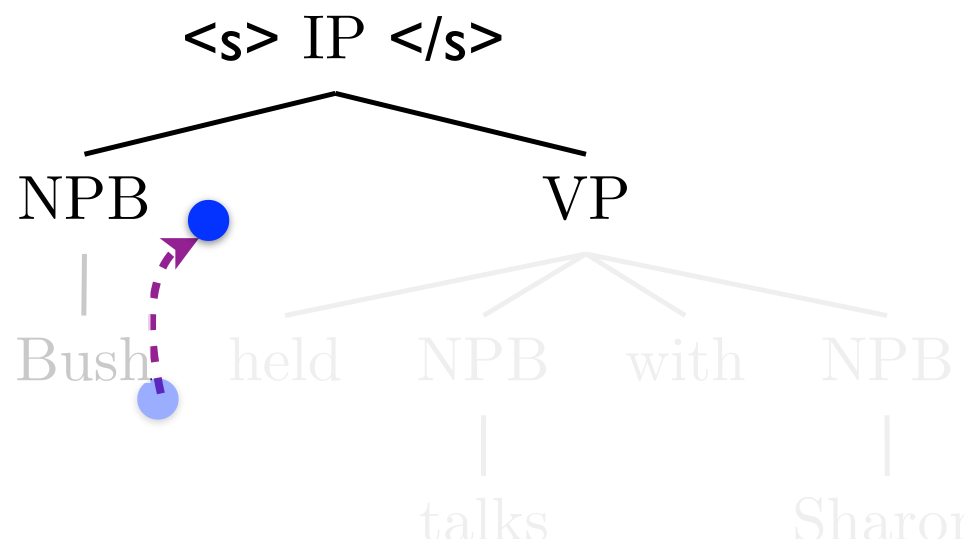
action: scan

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] *stack*

<s> Bush *hypothesis*



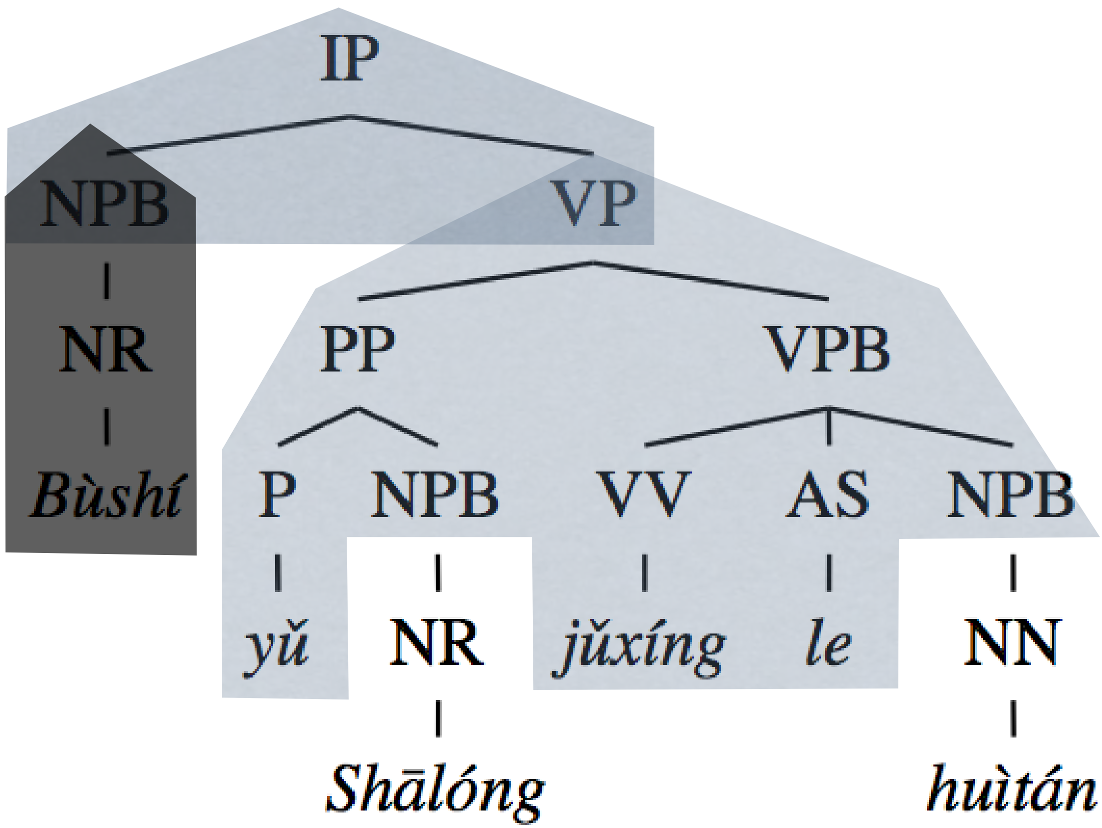


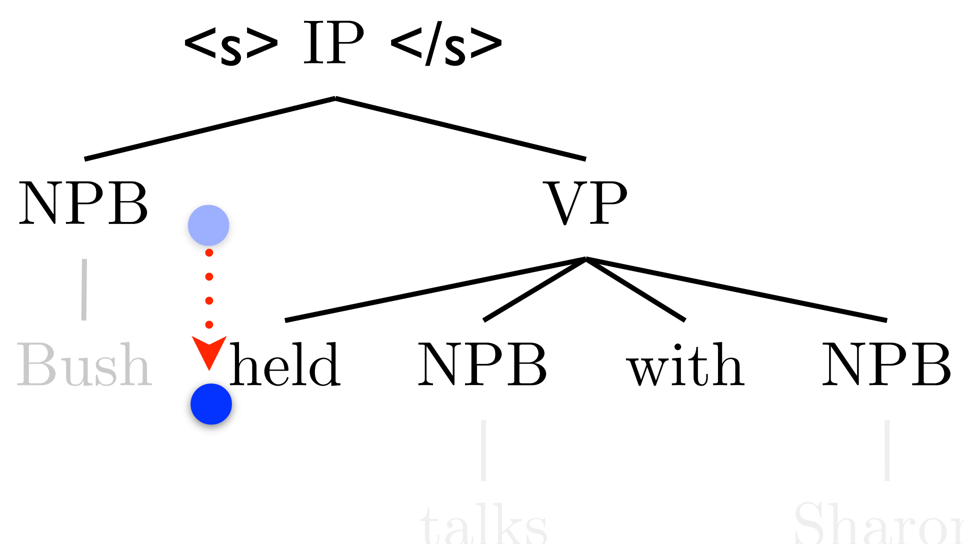
action: complete (pop)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →•held NPB with NPB] *stack*

<s> Bush *hypothesis*



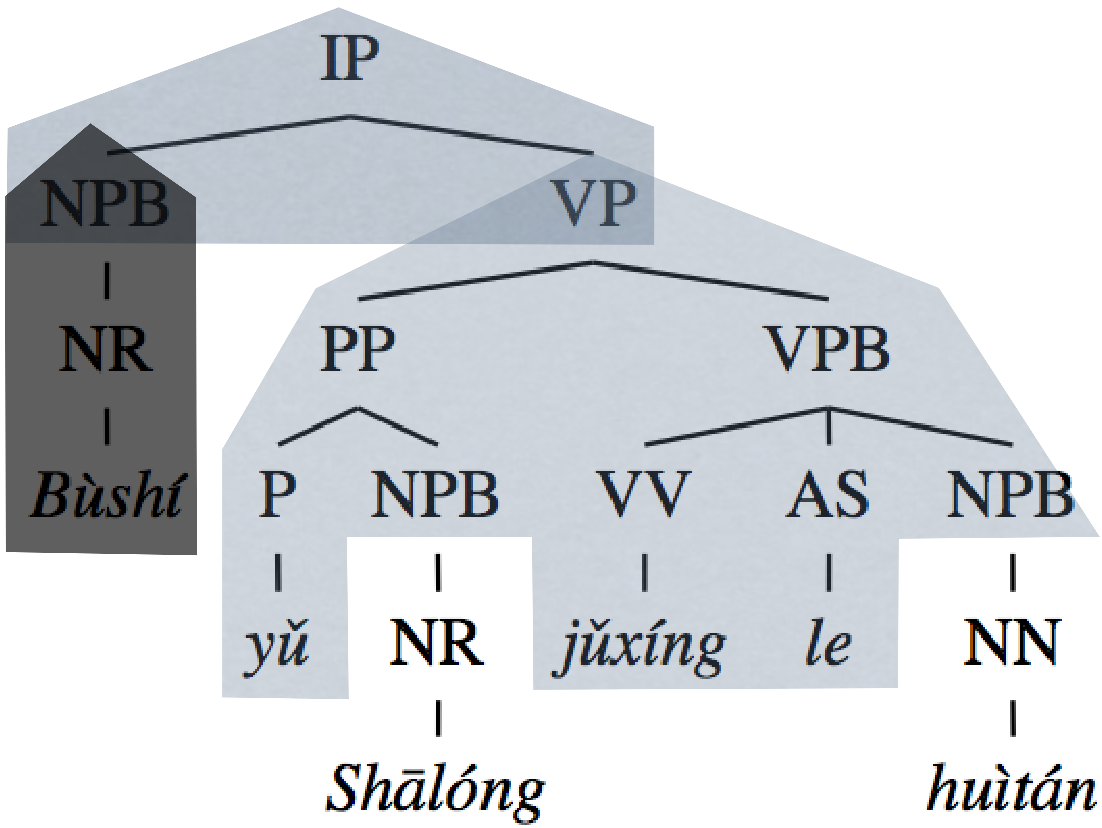


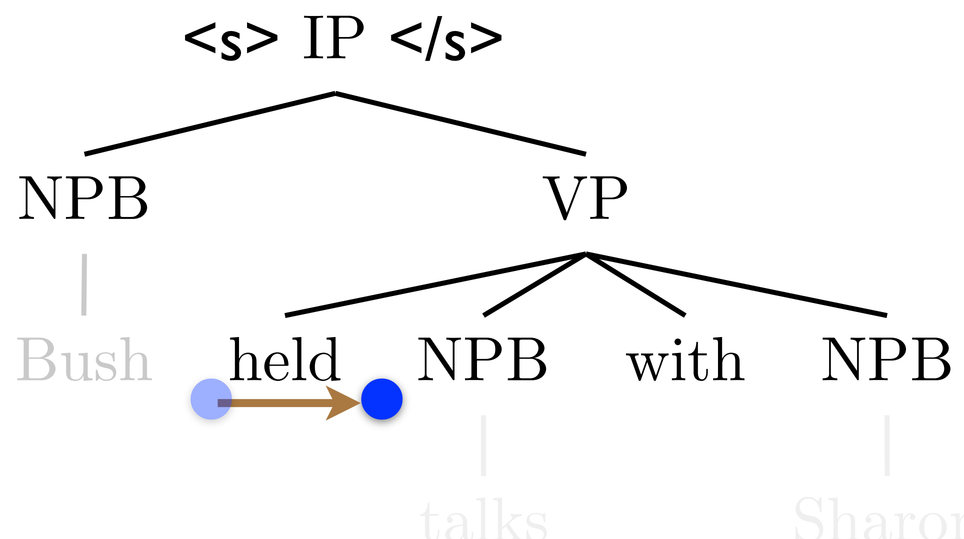
action: predict (push)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held•NPB with NPB]

<s> Bush held



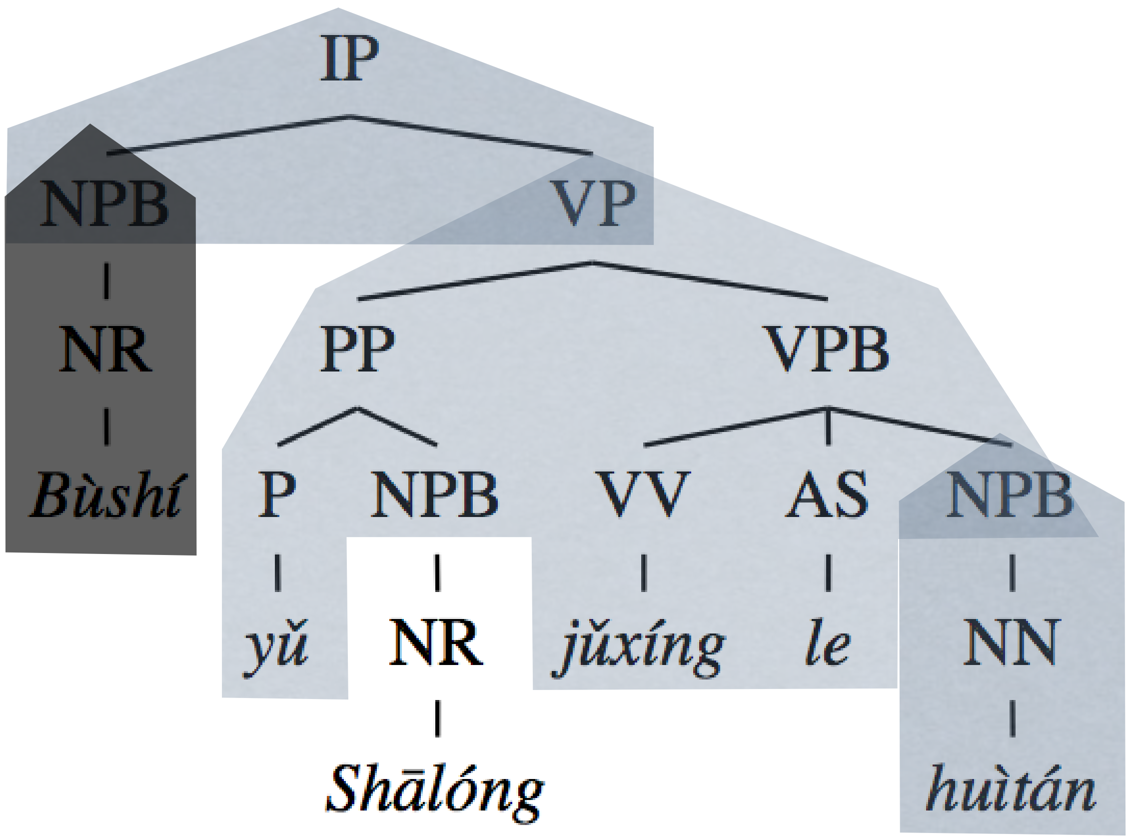


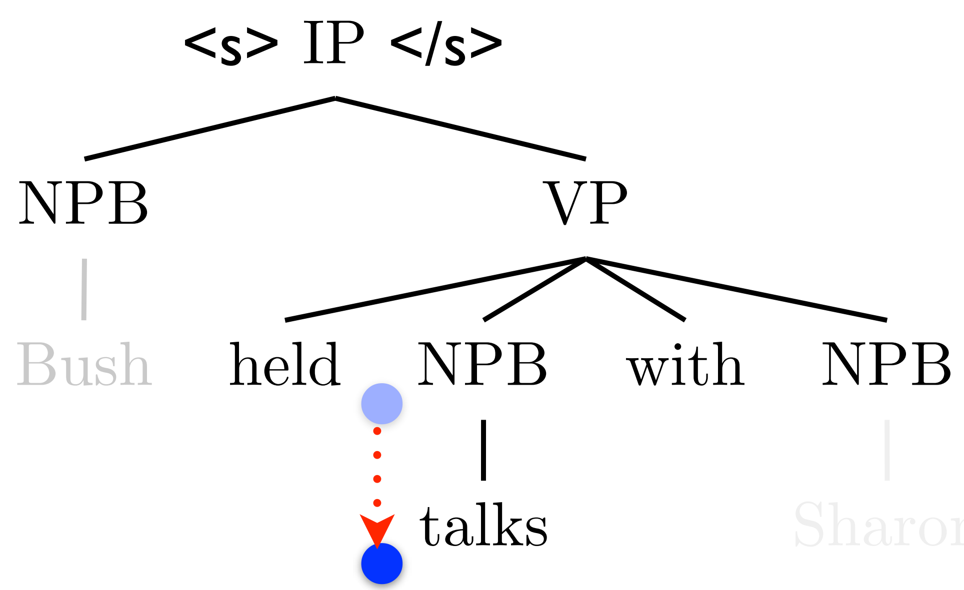
action: scan

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held•NPB with NPB] [NPB→•talks]

<s> Bush held



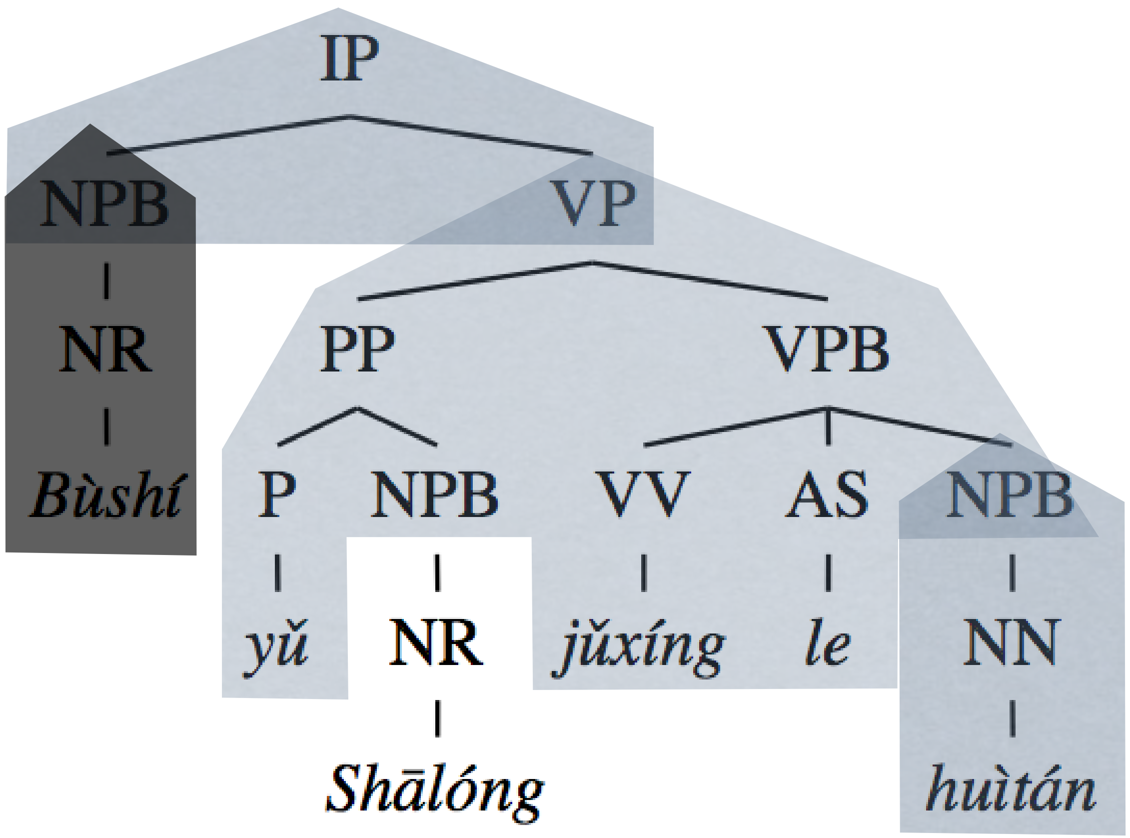


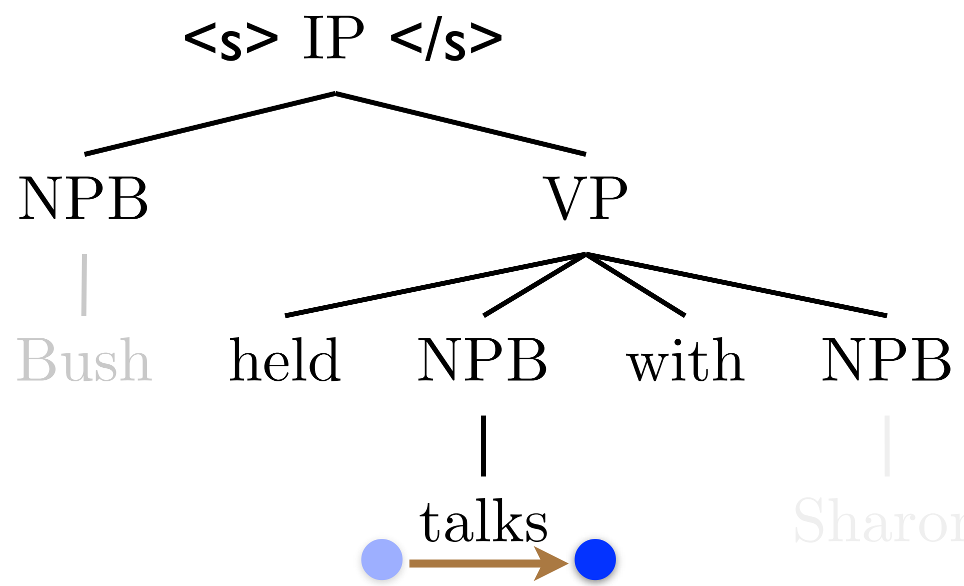
action: predict (push)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held•NPB with NPB] [NPB→talks•]

<s> Bush held talks



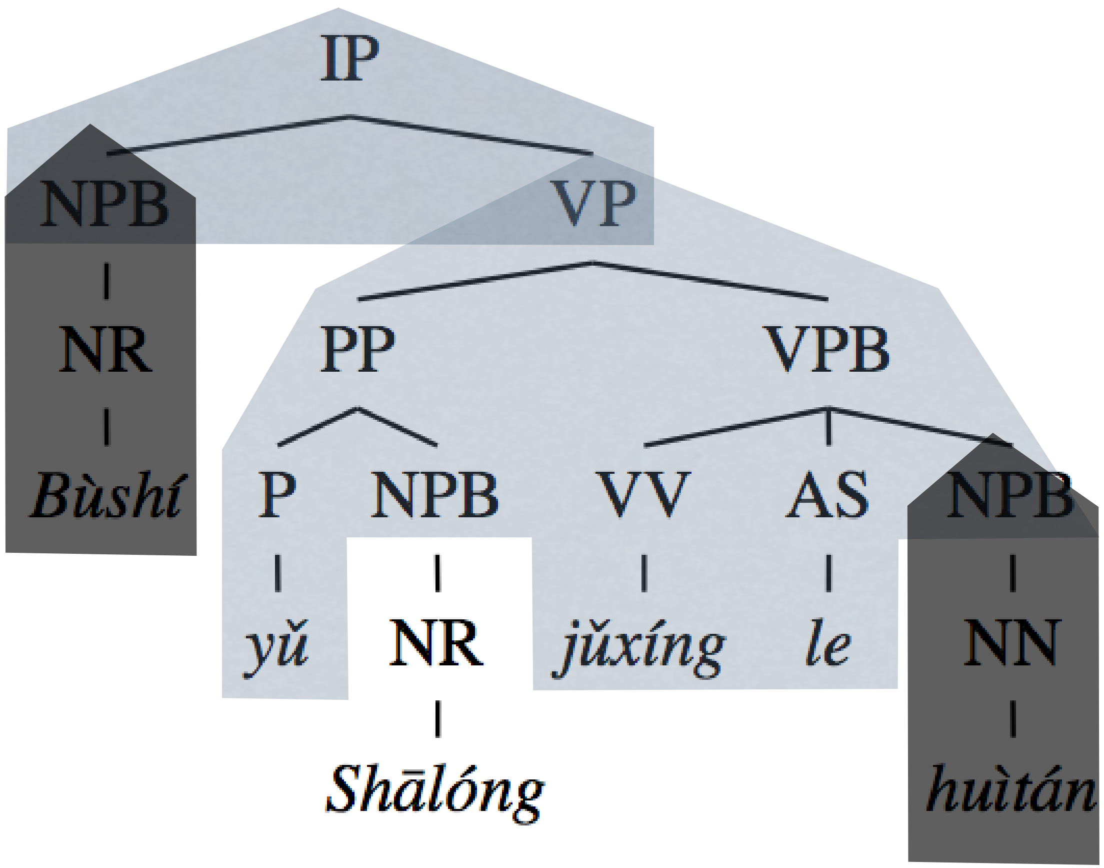


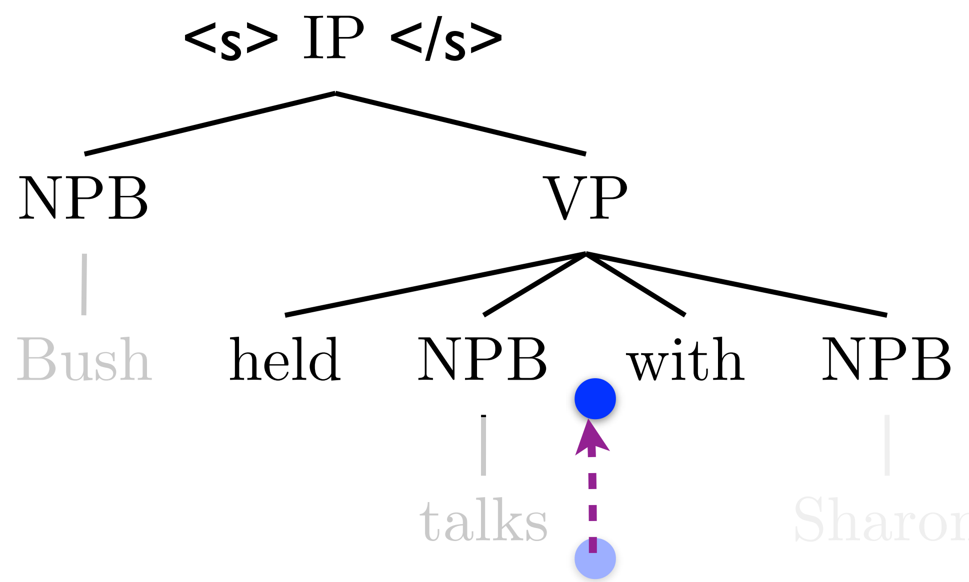
action: scan

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held NPB•with NPB]

<s> Bush held talks



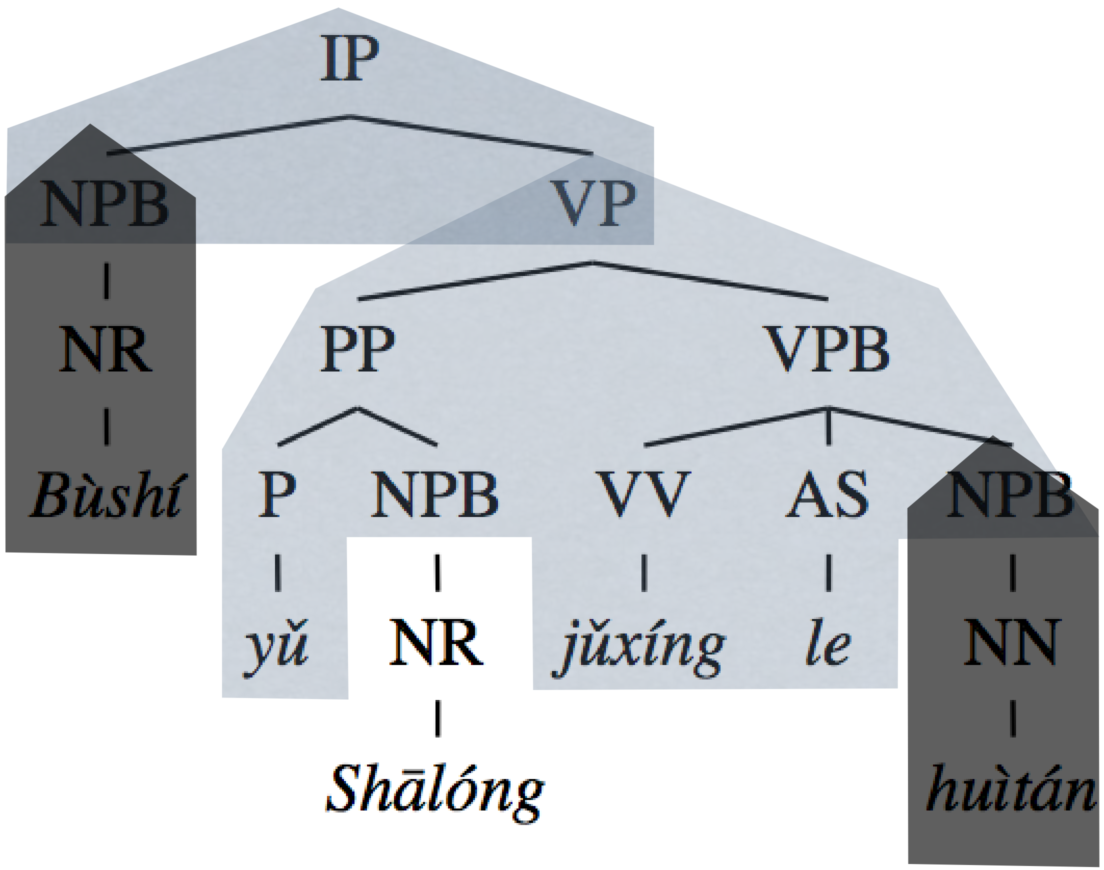


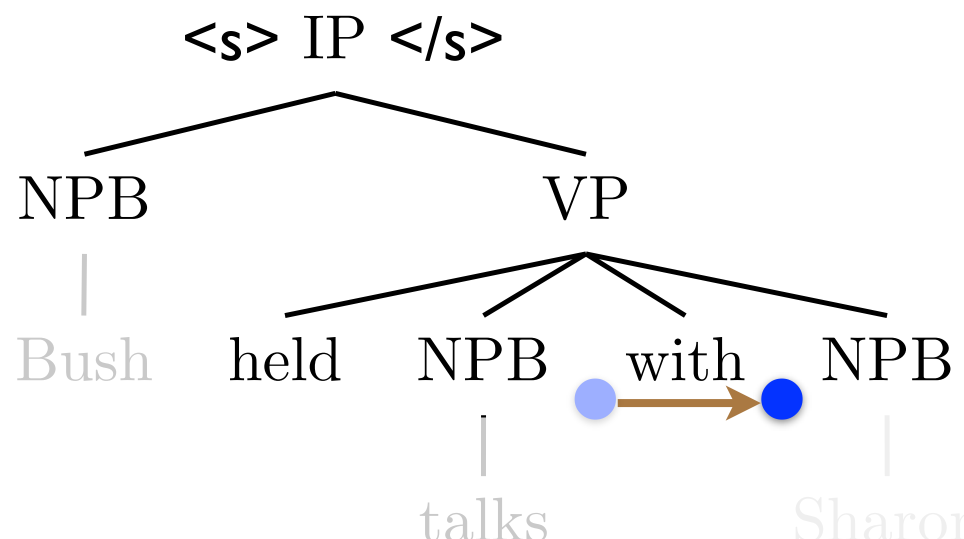
action: complete (pop)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held NPB with•NPB]

<s> Bush held talks with



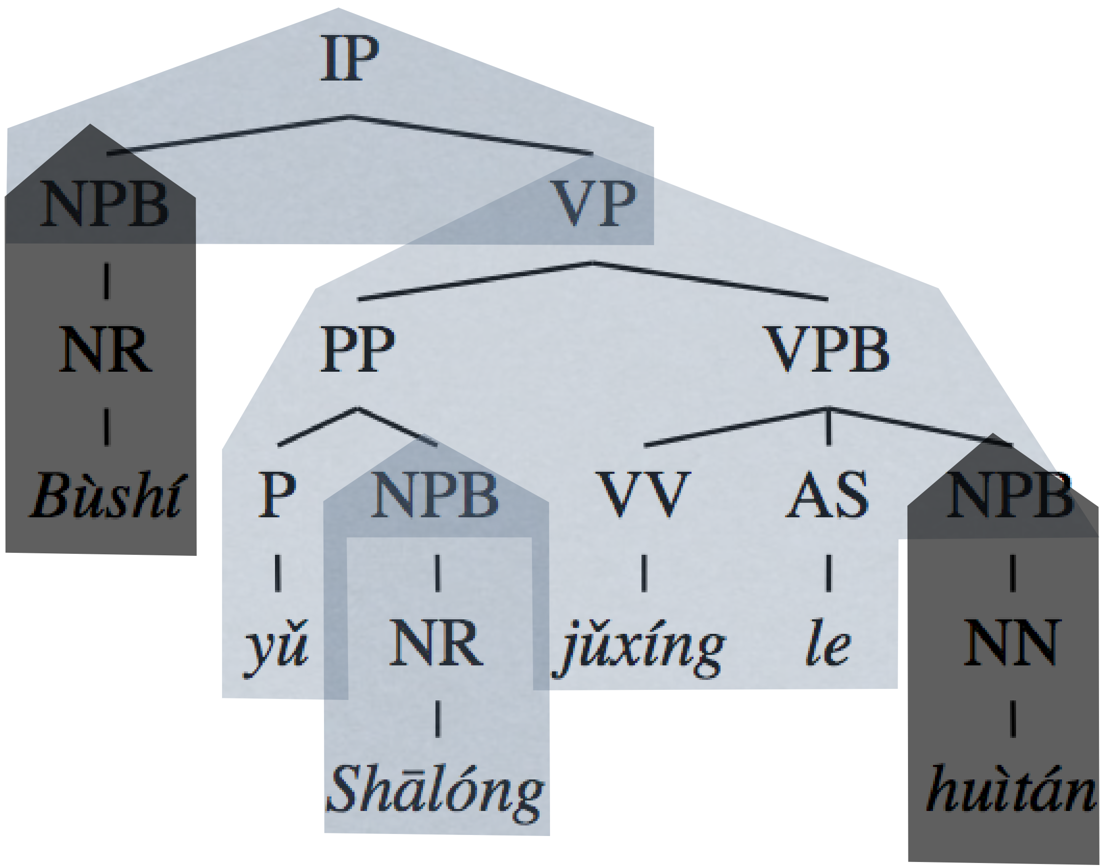


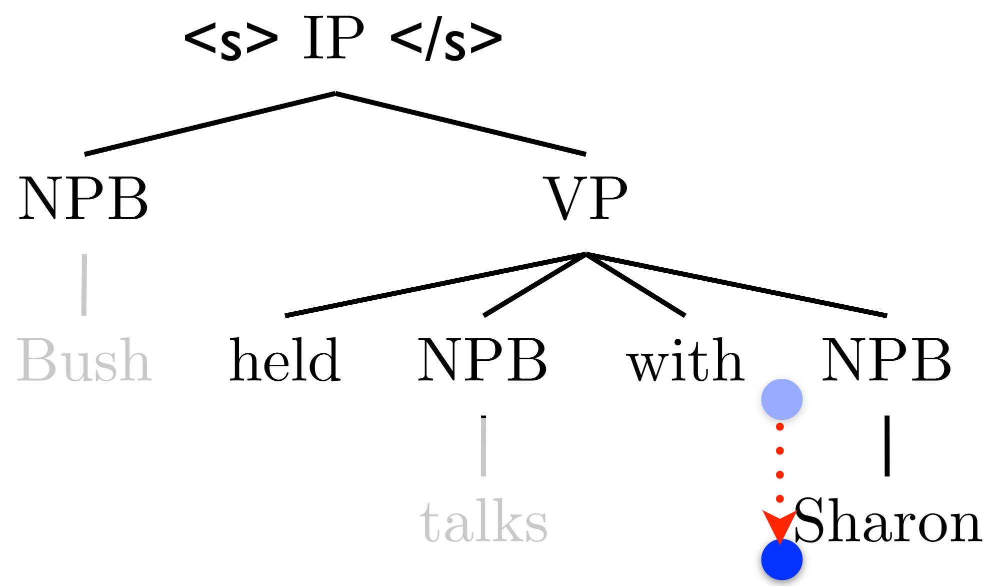
action: scan

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held NPB with•NPB][NPB→•Sharon]

<s> Bush held talks with



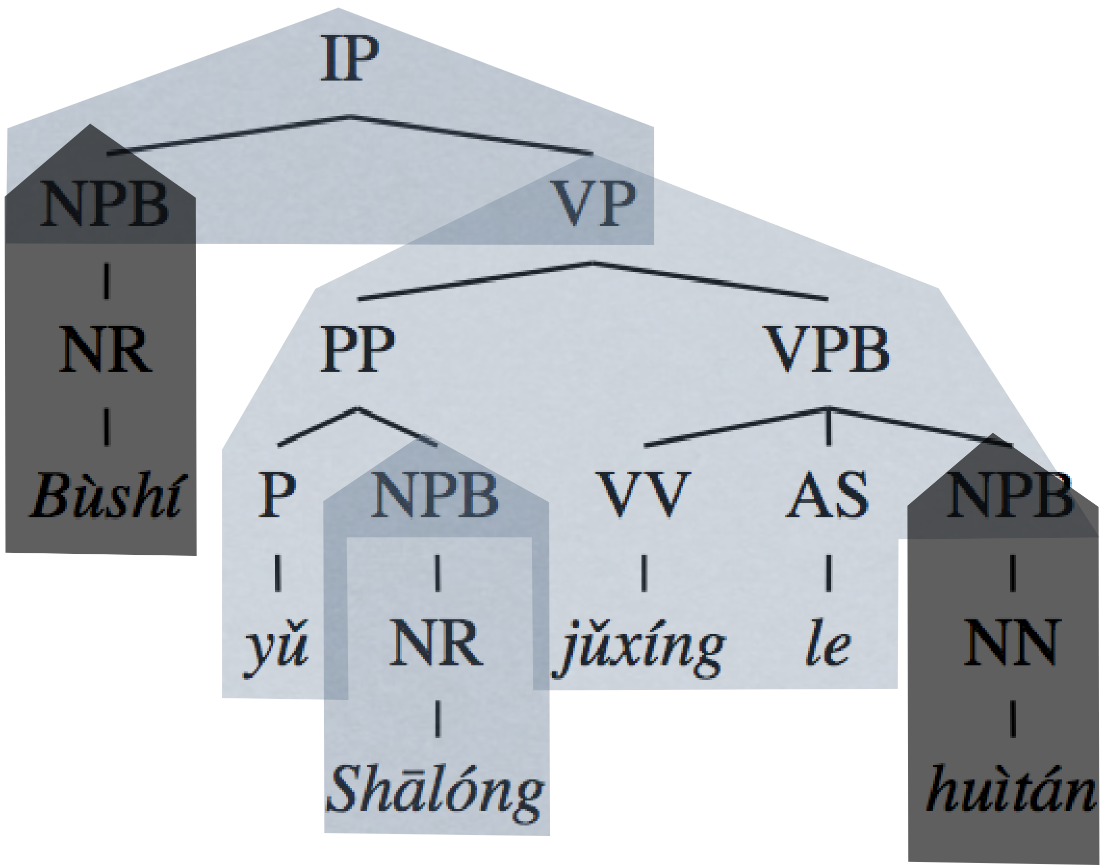


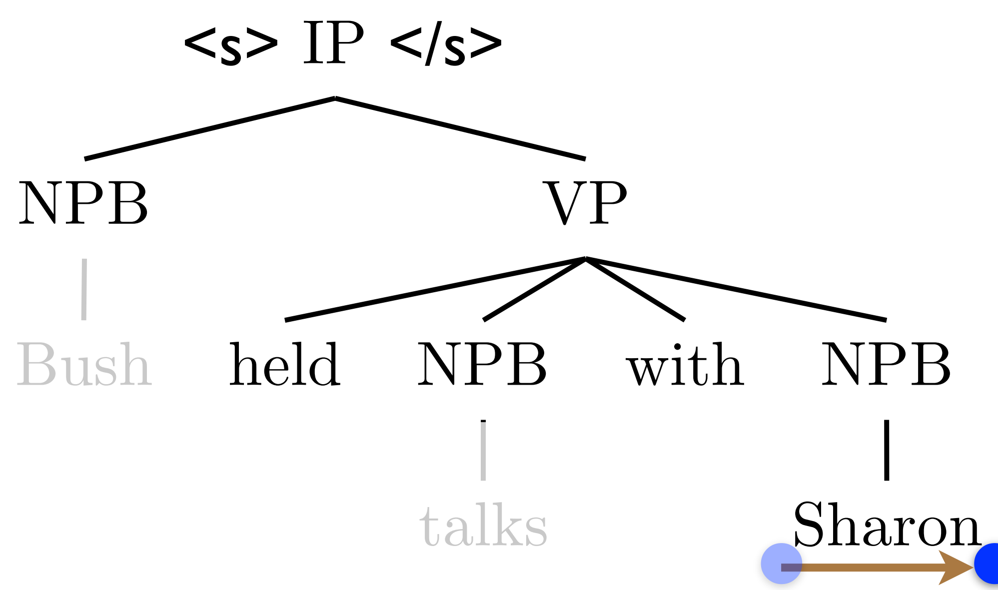
action: predict (push)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held NPB with•NPB][NPB→Sharon•]

<s> Bush held talks with Sharon



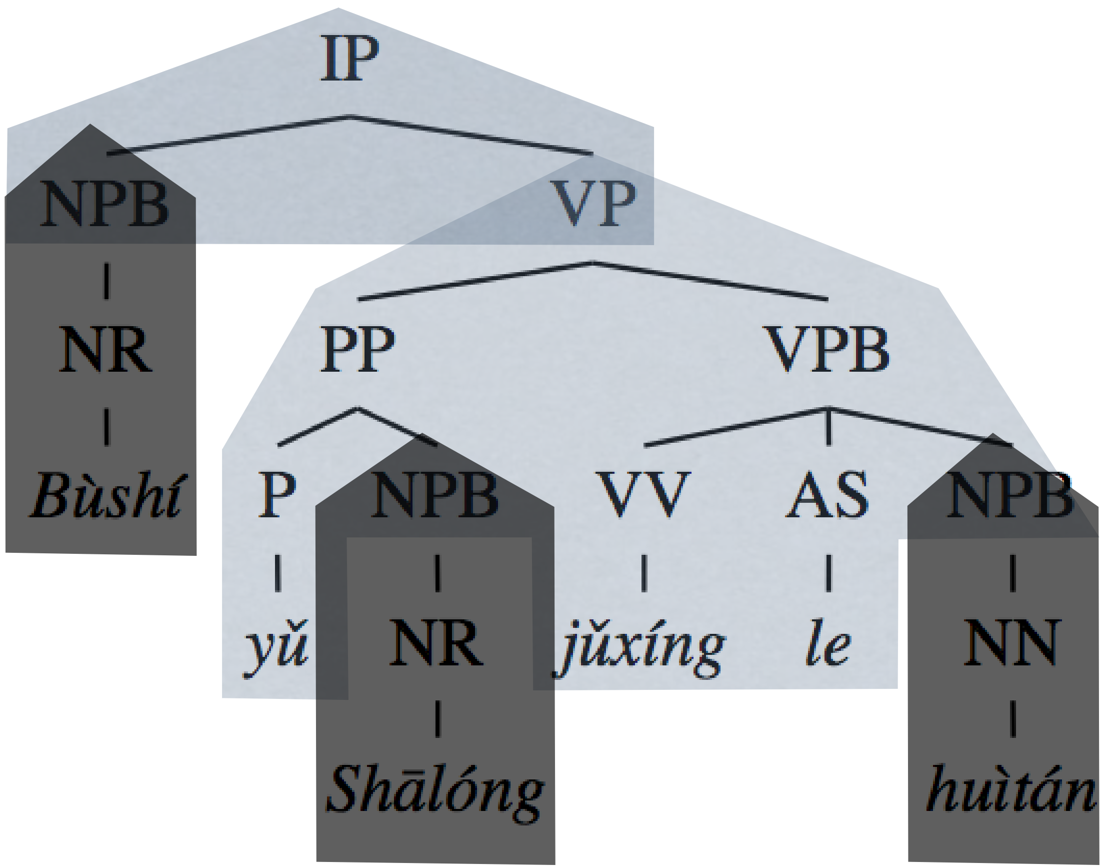


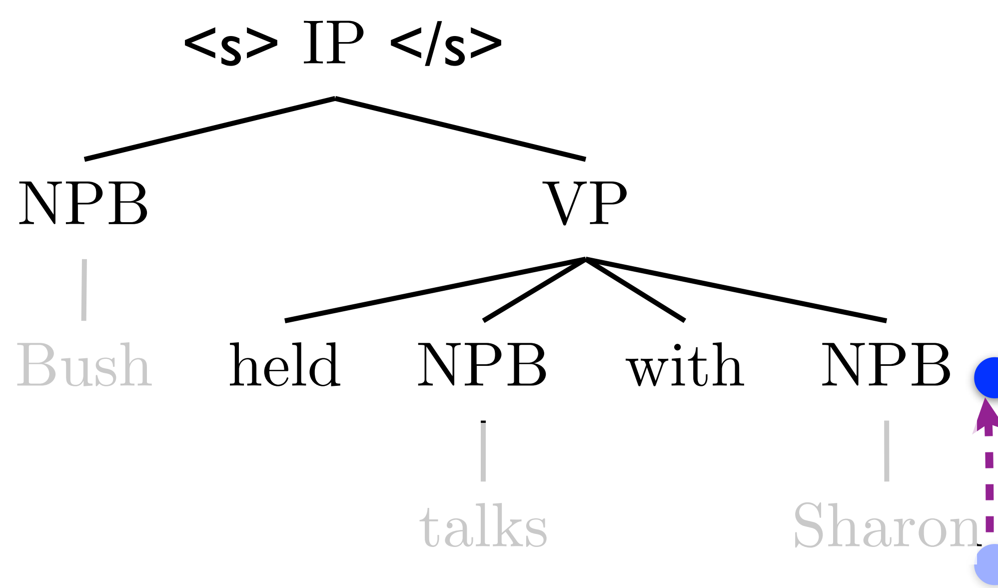
action: scan

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB•VP] [VP →held NPB with NPB•]

<s> Bush held talks with Sharon



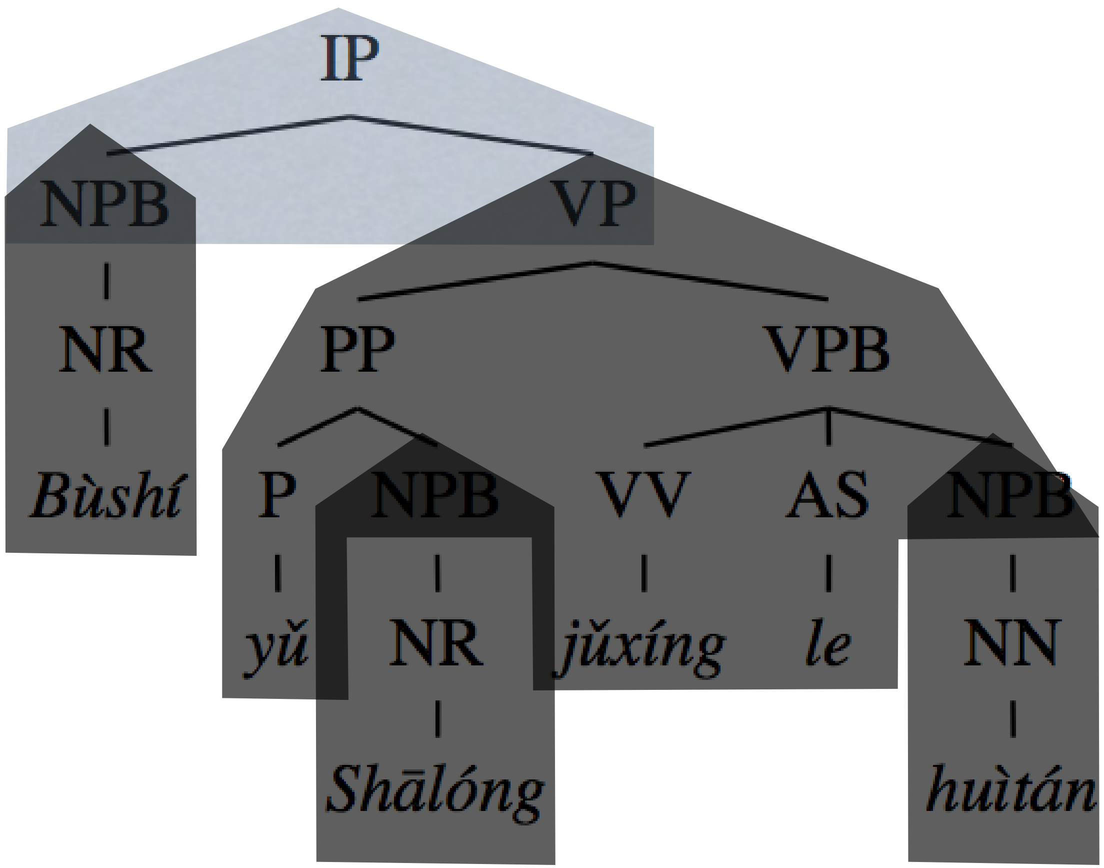


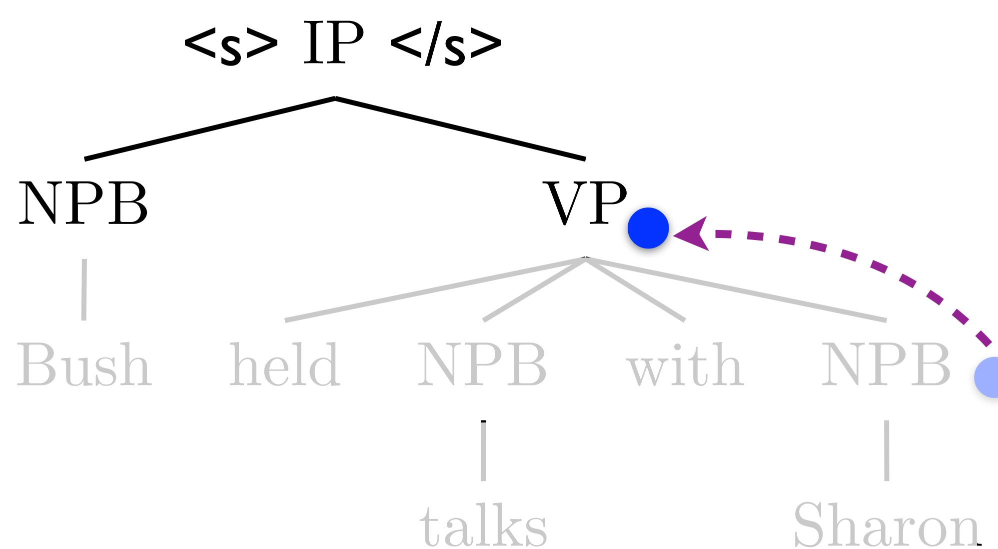
action: complete (pop)

Example Incremental Decoding

[ε→<s> •IP </s>] [IP→ NPB VP•]

<s> Bush held talks with Sharon



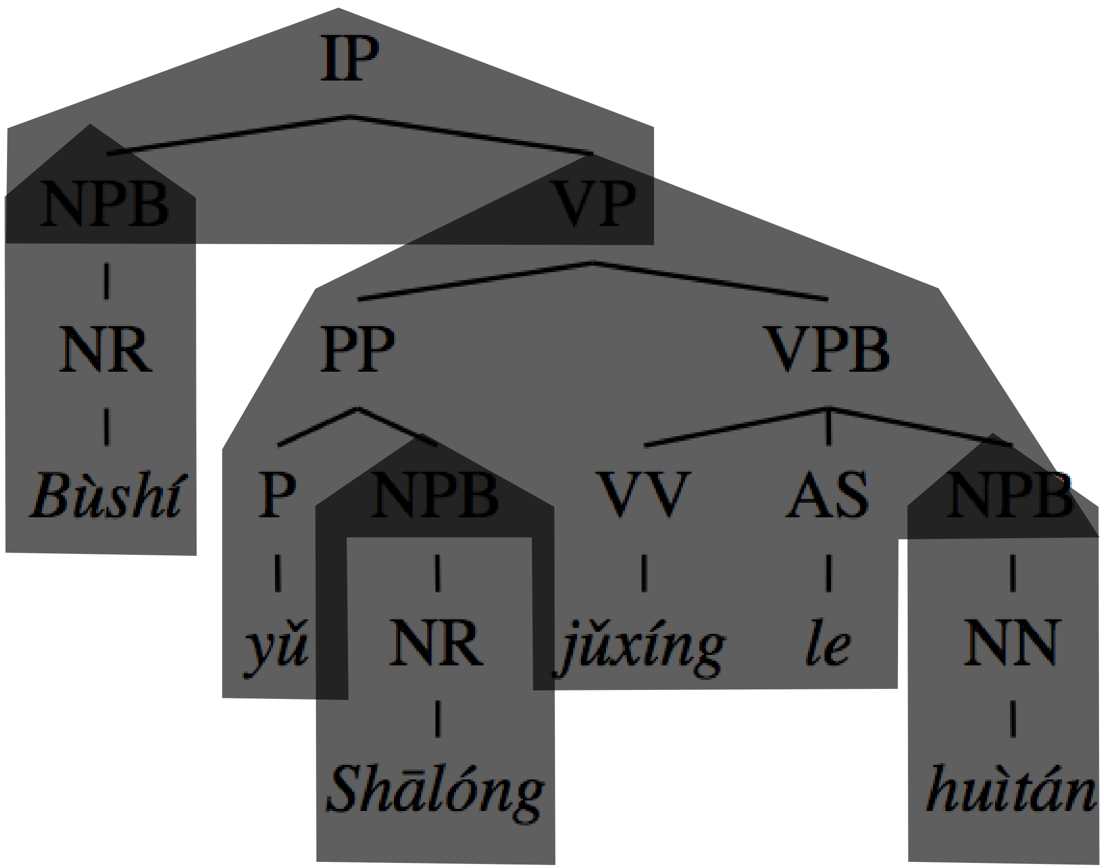


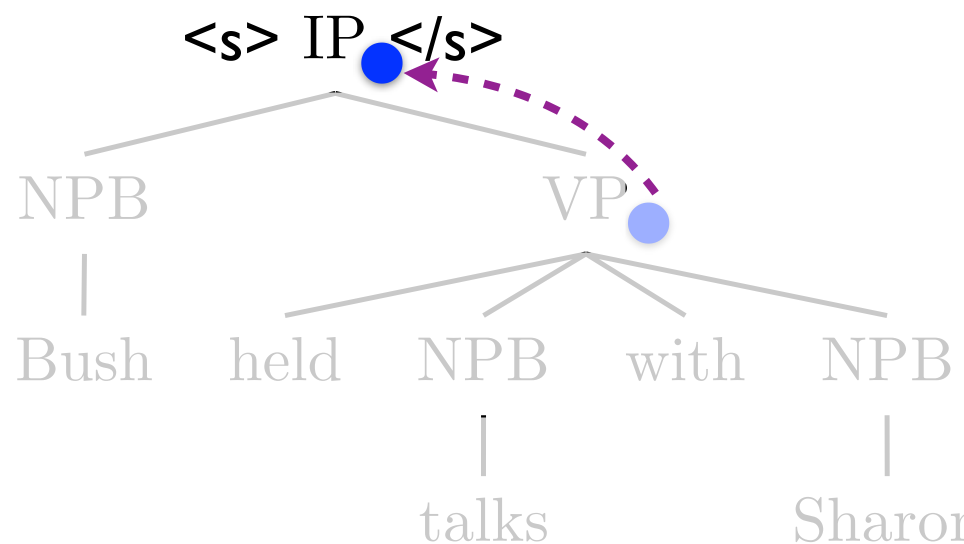
action: complete (pop)

Example Incremental Decoding

[ε→<s> IP• </s>]

<s> Bush held talks with Sharon



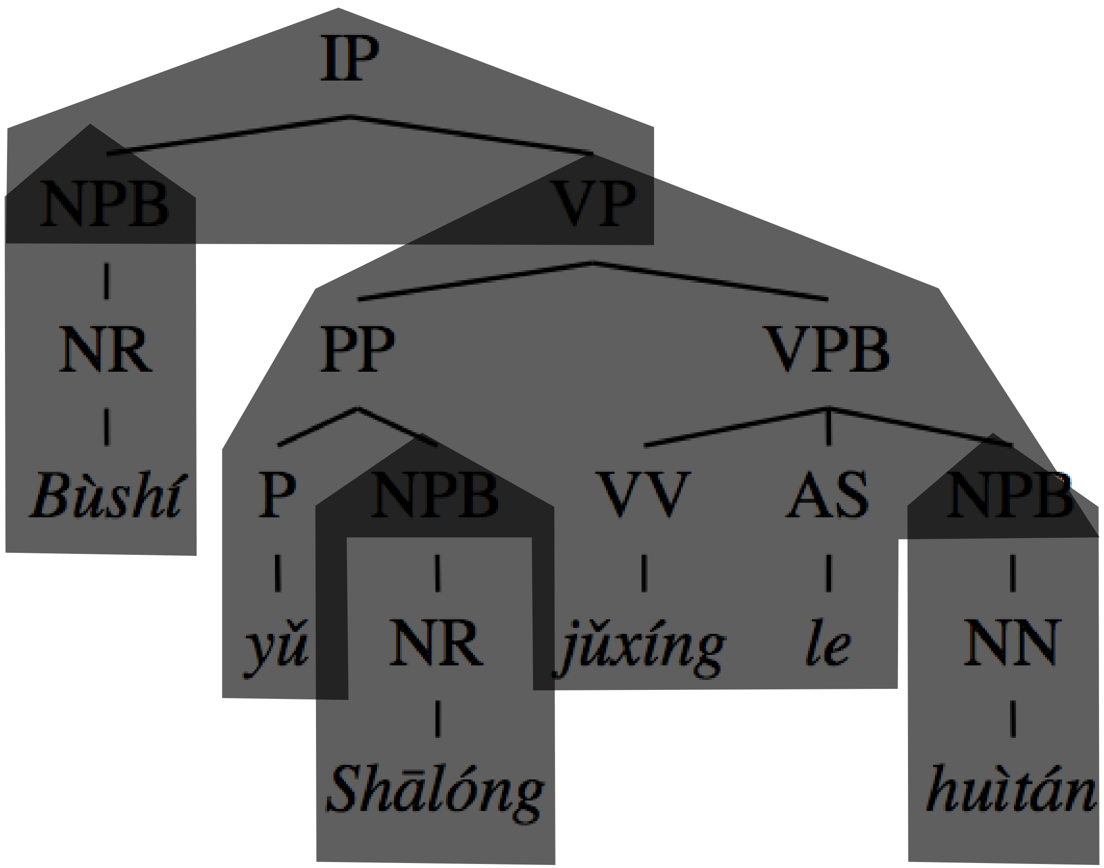


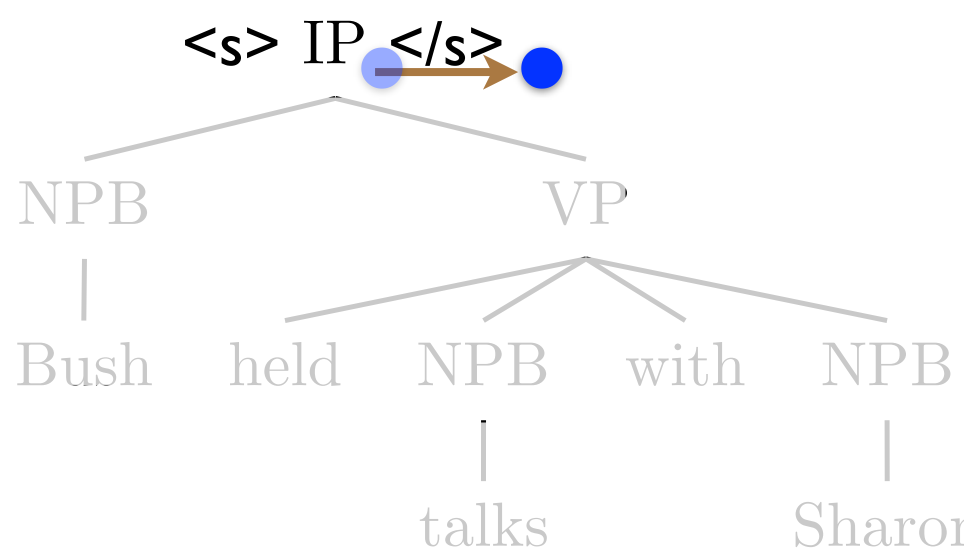
action: complete (pop)

Example Incremental Decoding

[ε→<s> IP </s>•]

<s> Bush held talks with Sharon </s>



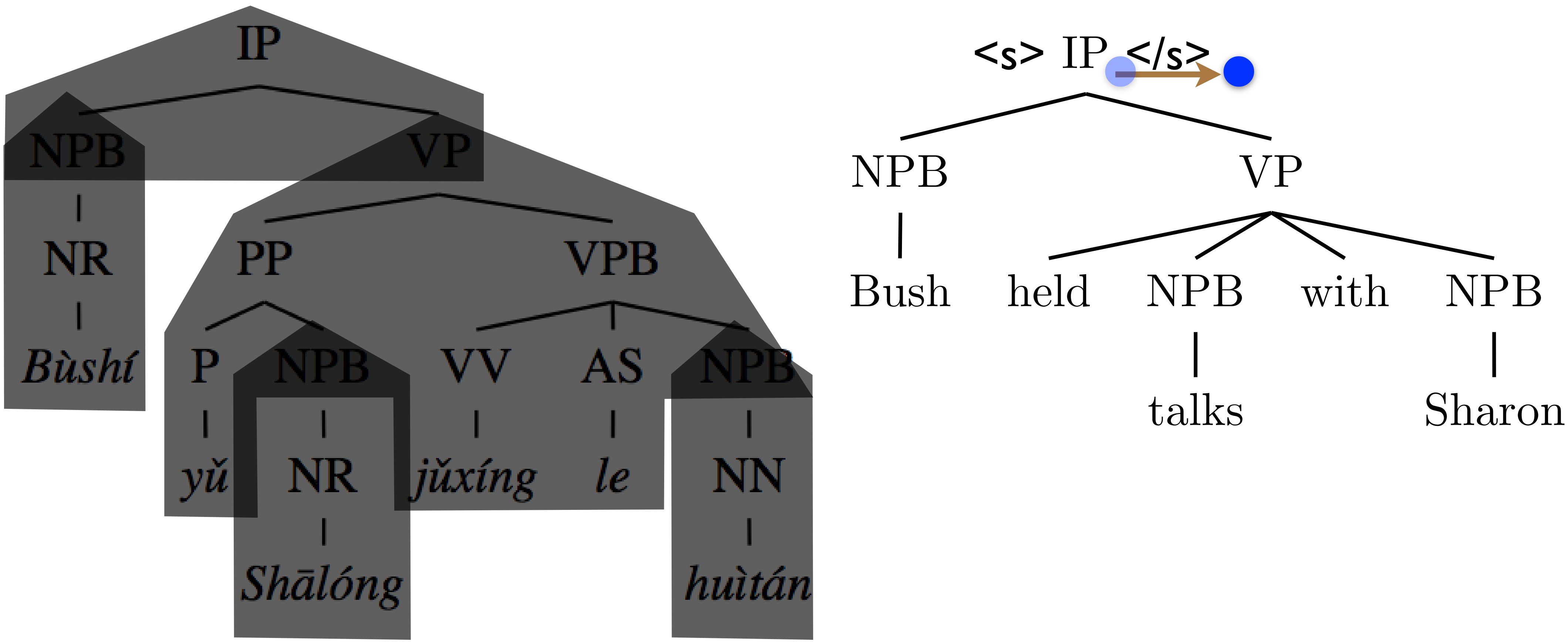


action: scan

Example Incremental Decoding

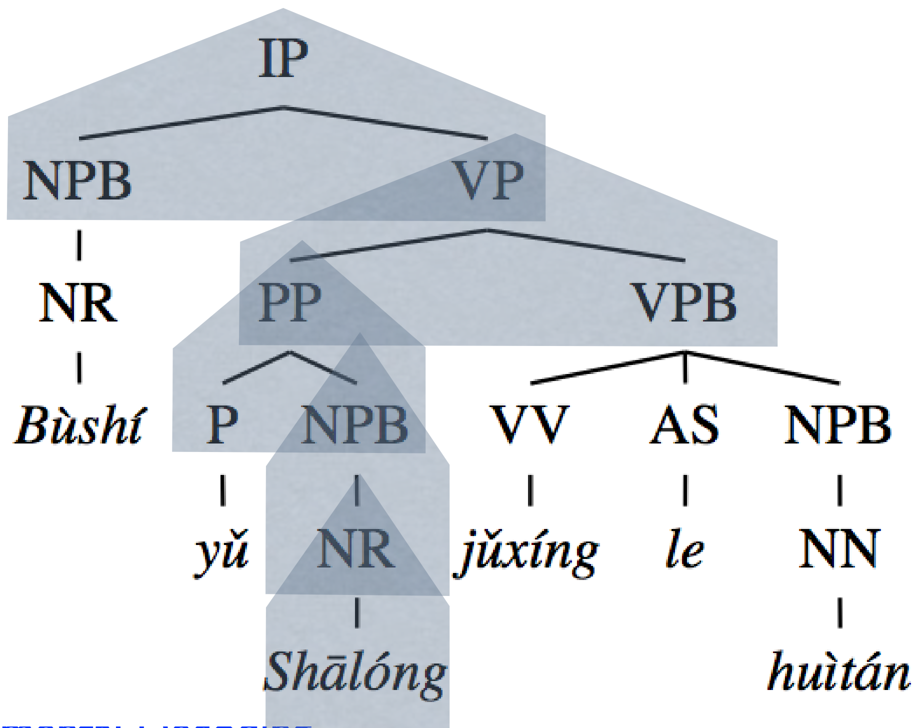
[ε→<s> IP </s>•]

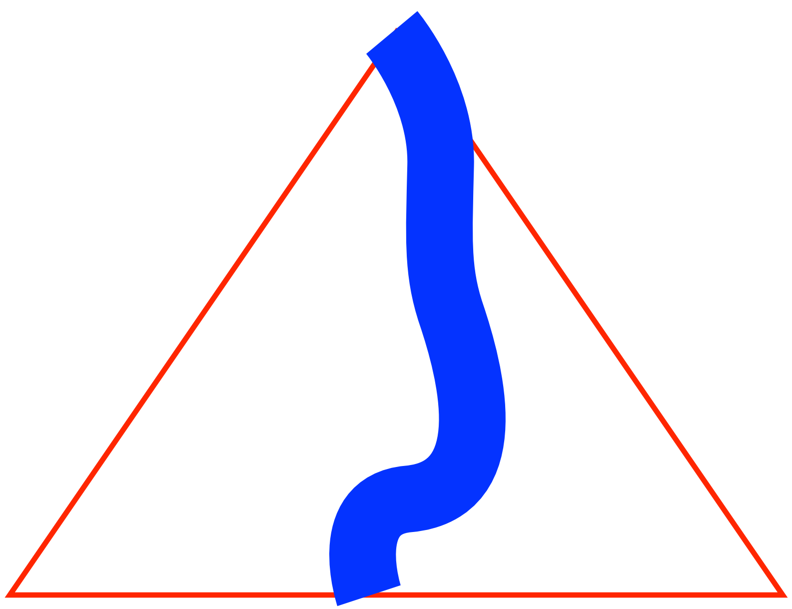
<s> Bush held talks with Sharon </s>



Complexity Analysis

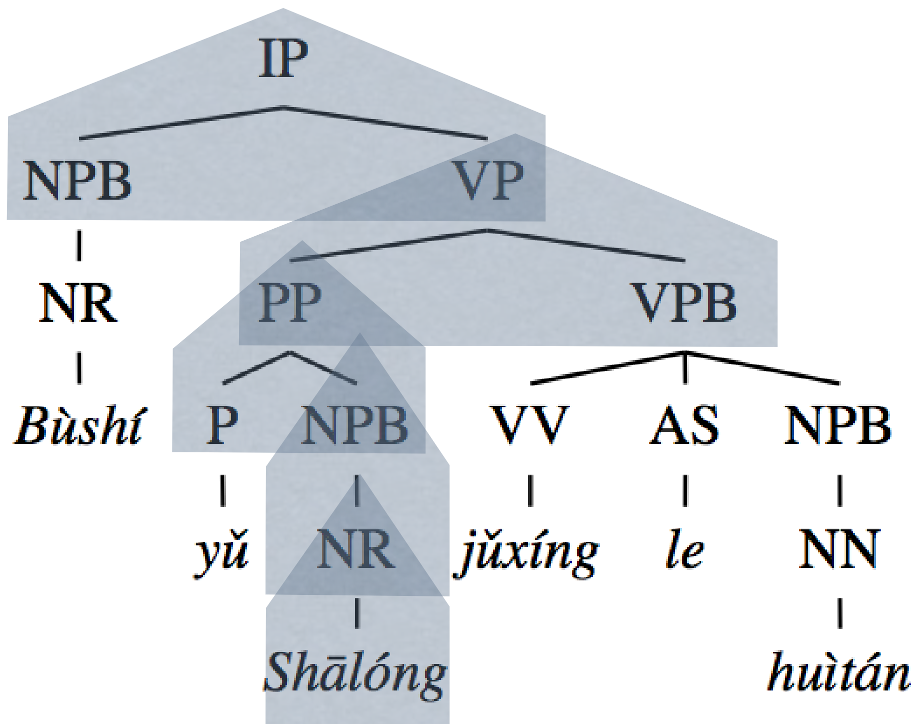
- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules => $O(c^{\log n}) = O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)

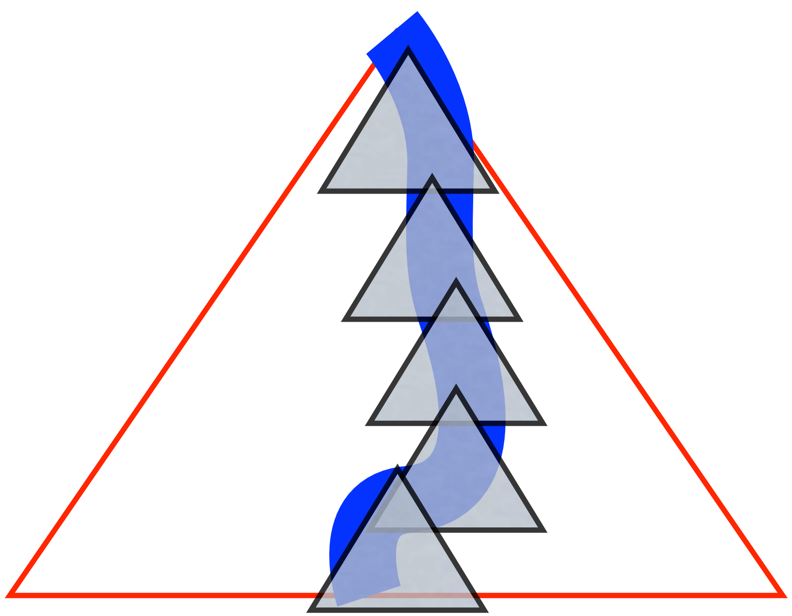




Complexity Analysis

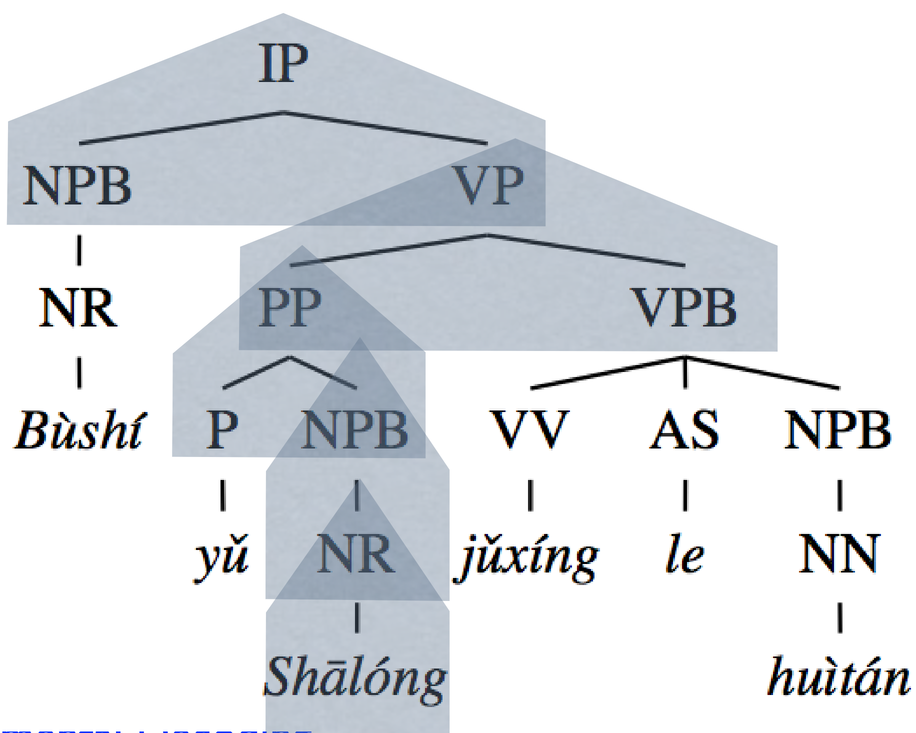
- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules => $O(c^{\log n}) = O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)



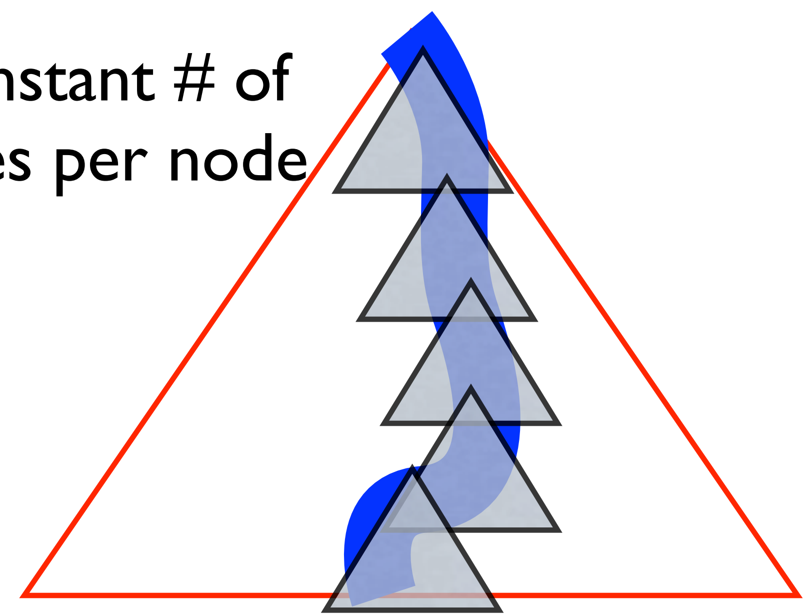


Complexity Analysis

- how many possible derivation stacks?
- exponential in root-to-leaf path length (tree depth)
- tree depth $O(\log n)$; const # rules => $O(c^{\log n}) = O(n^{\log c})$
- so avg-case complexity is polynomial (see proof in paper)

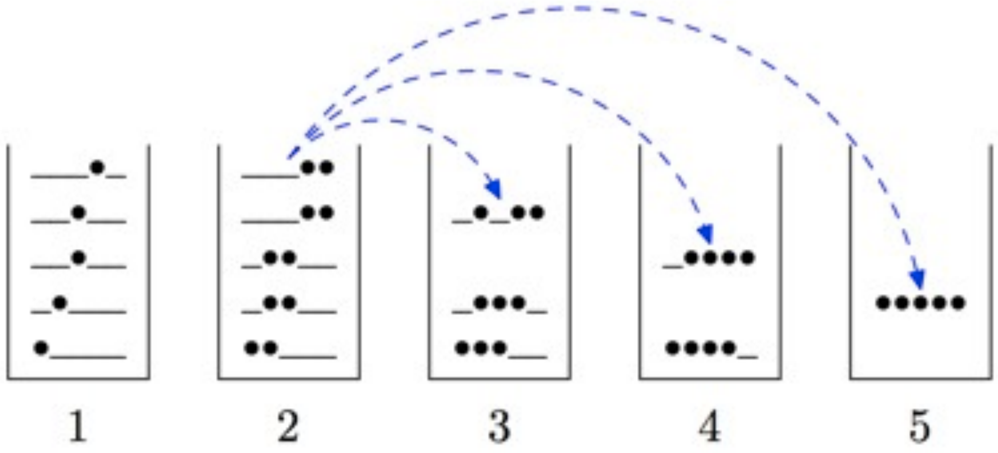


constant # of rules per node



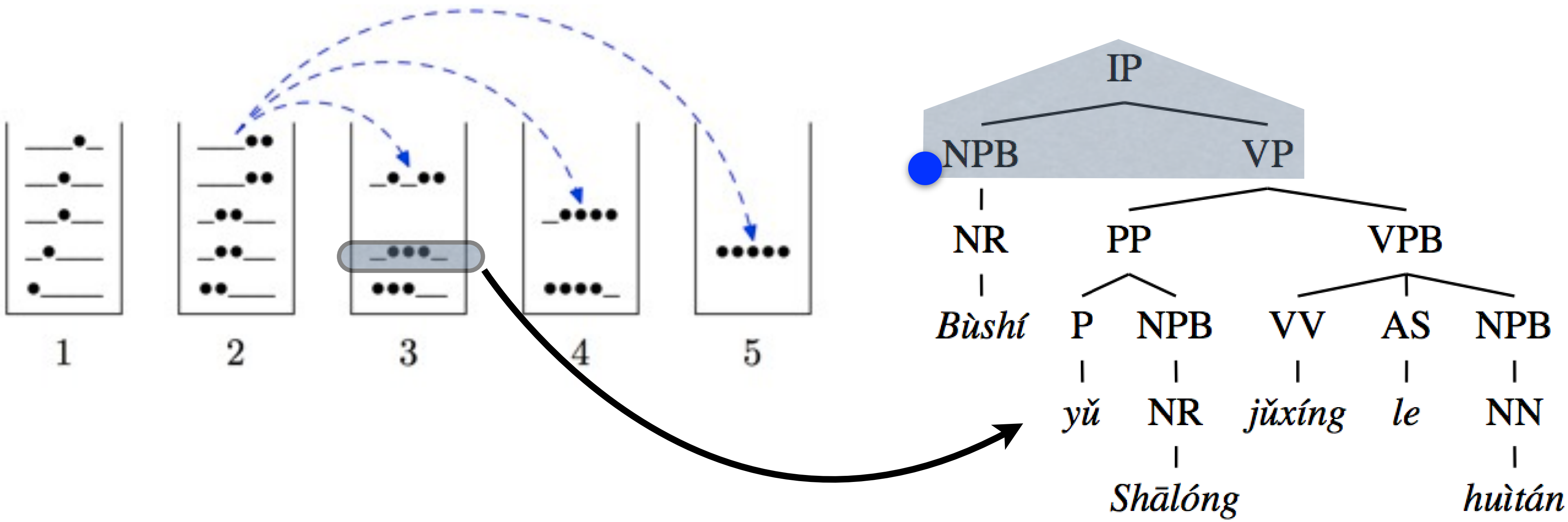
Beam Search in Practice

- very similar to phrase-based beam search
- coverage-vectors => derivation stacks
- beaming: # of Chinese tree nodes in black or grey
- assume constant # of rules per tree node: linear-time



Beam Search in Practice

- very similar to phrase-based beam search
- coverage-vectors => derivation stacks
- beaming: # of Chinese tree nodes in black or grey
- assume constant # of rules per tree node: linear-time



Related Work

- Watanabe et al (2006) presents similar incremental decoding algorithm for Hiero-style systems
  - but complexity is super-polynomial in theory
  - and quadratic in practice (just like phrase-based)
  - requires Greibach Normal Form grammar $A \rightarrow a\ B\ C\ D$
- Dyer and Resnik (2010) use two-pass decoding
  - first-pass: no LM. incremental Earley-style
  - second-pass: +LM. bottom-up CKY w/ cube pruning
- ours work: one-pass, incremental, +LM, all grammars

Experiments

Experimental Setup

- Chinese-to-English translation
  - on a Python implementation of tree-to-string system
- 1.5M sentence pairs (38M/32M words in Chn/Eng)
- dev: NIST 2006 (616 sent); test: NIST 2008 (691 sent.)
- Chinese-side parsed by Berkeley parser (Petrov & Klein, 07)
- rules extracted using GHKM algorithm (Galley et al, 04; 06)
- trigram language model trained on the English side
- feature weights tuned using MERT (Och, 03)

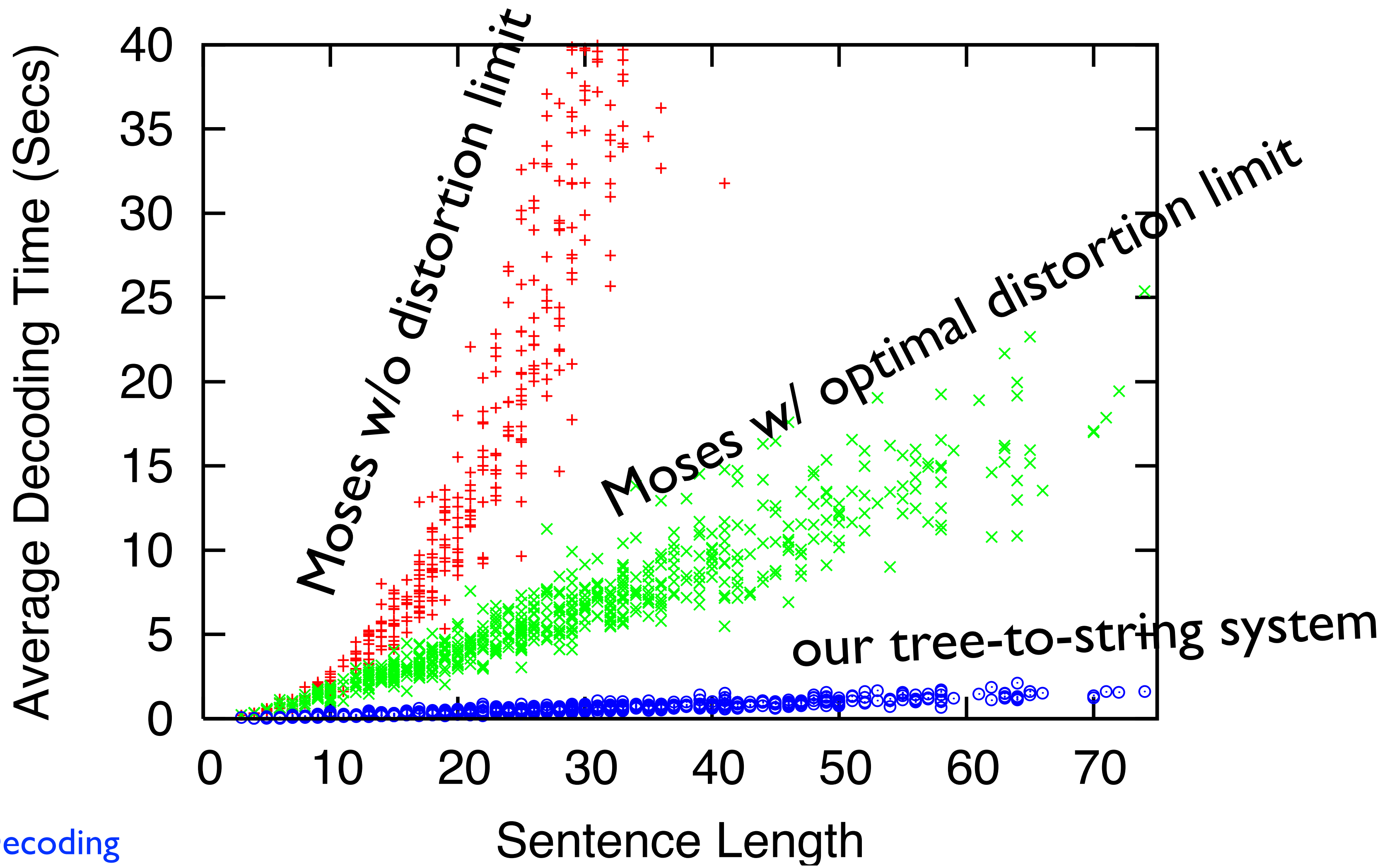
Comparison with Moses

- we train/tune Moses with various distortion limits
- our incremental tree-to-string is ~30 times faster
  - this includes parsing time (0.2s per sentence)

	BLEU	*time*
Moses (optimal distortion limit=10)	29.4	10.8s
tree-to-string: incremental (*b*=10)	29.5	0.3s
tree-to-string: incremental (*b*=50)	30.0	0.8s

Comparison with Moses

- incremental tree-to-string is linear-time in practice
- and 30 times faster than Moses (distortion limit=10)



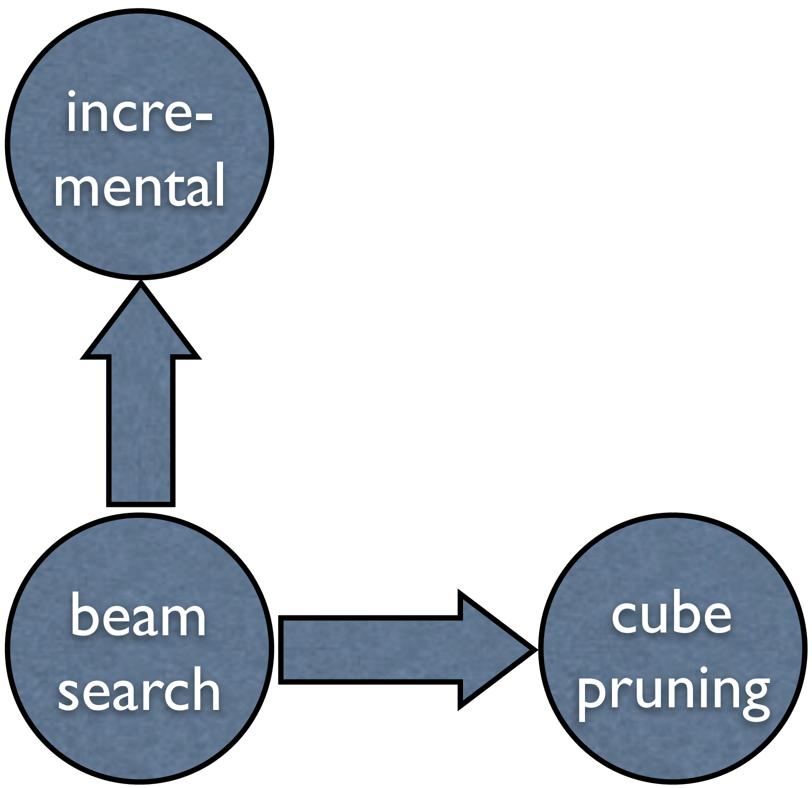
Comparison with Cube Pruning

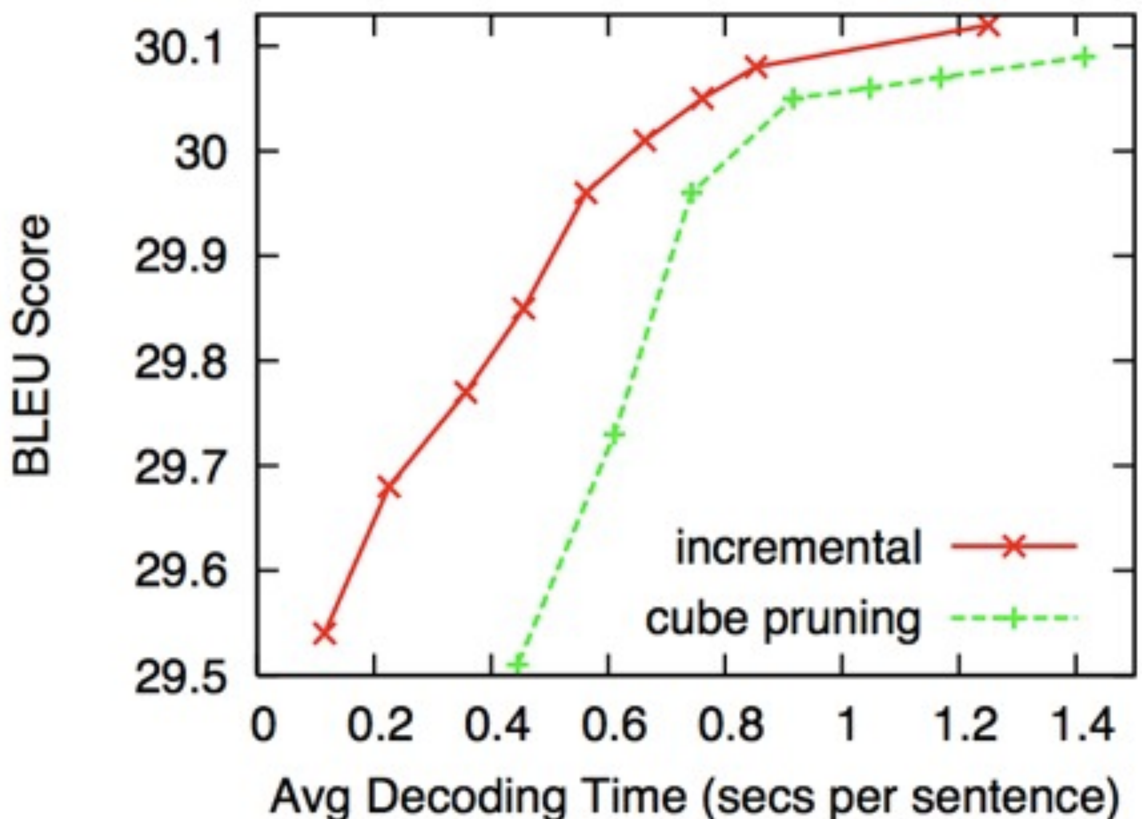
- incremental is slightly faster than cube pruning

	BLEU	*time*
Moses (optimal d_{max}=10)	29.4	10.8s
tree-to-string: incremental (b=10)	29.5	0.3s
tree-to-string: incremental (b=50)	30.0	0.8s
tree-to-string: cube pruning (b=10)	29.5	0.6s
tree-to-string: cube pruning (b=50)	30.0	1.0s

Comparison with Cube Pruning

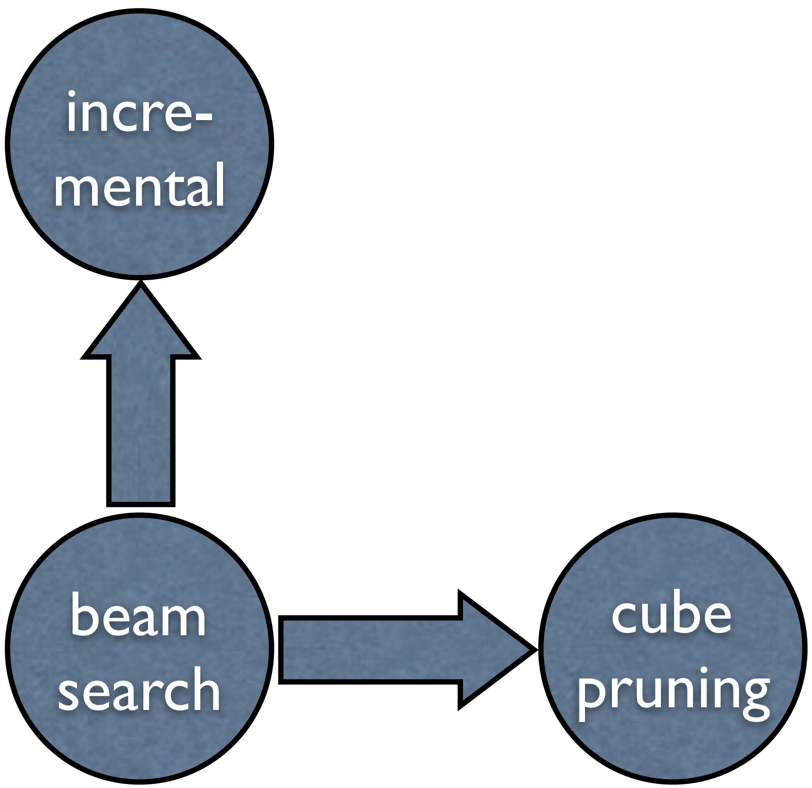
- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
  - we envision their combination will be even faster

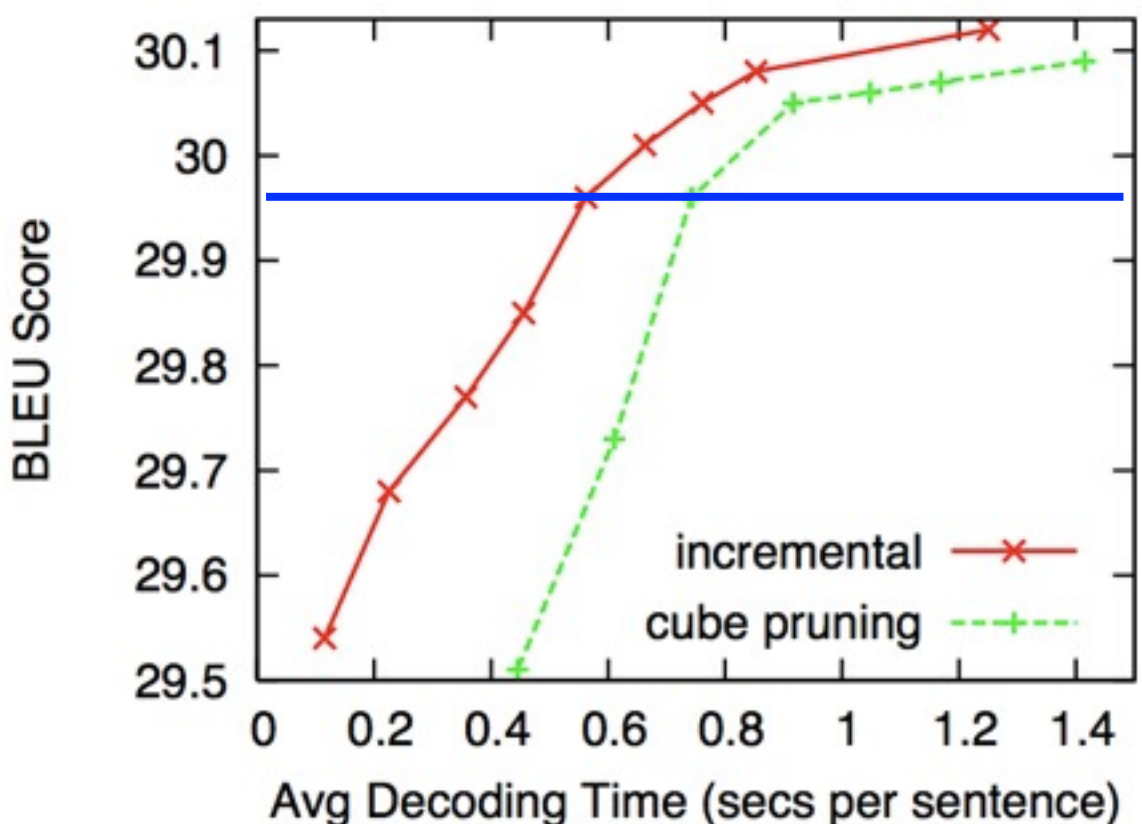




Comparison with Cube Pruning

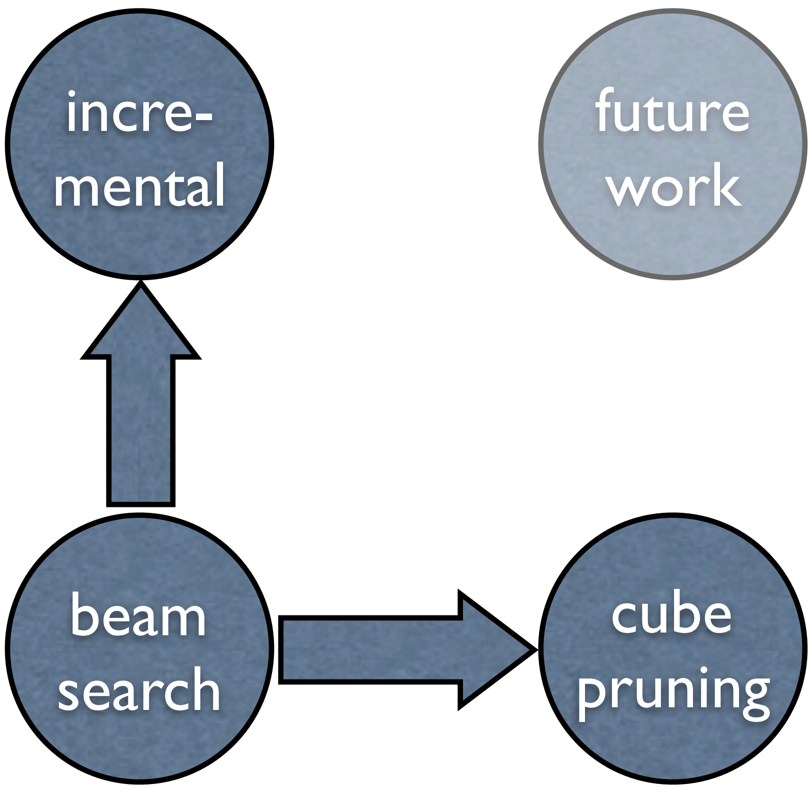
- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
  - we envision their combination will be even faster

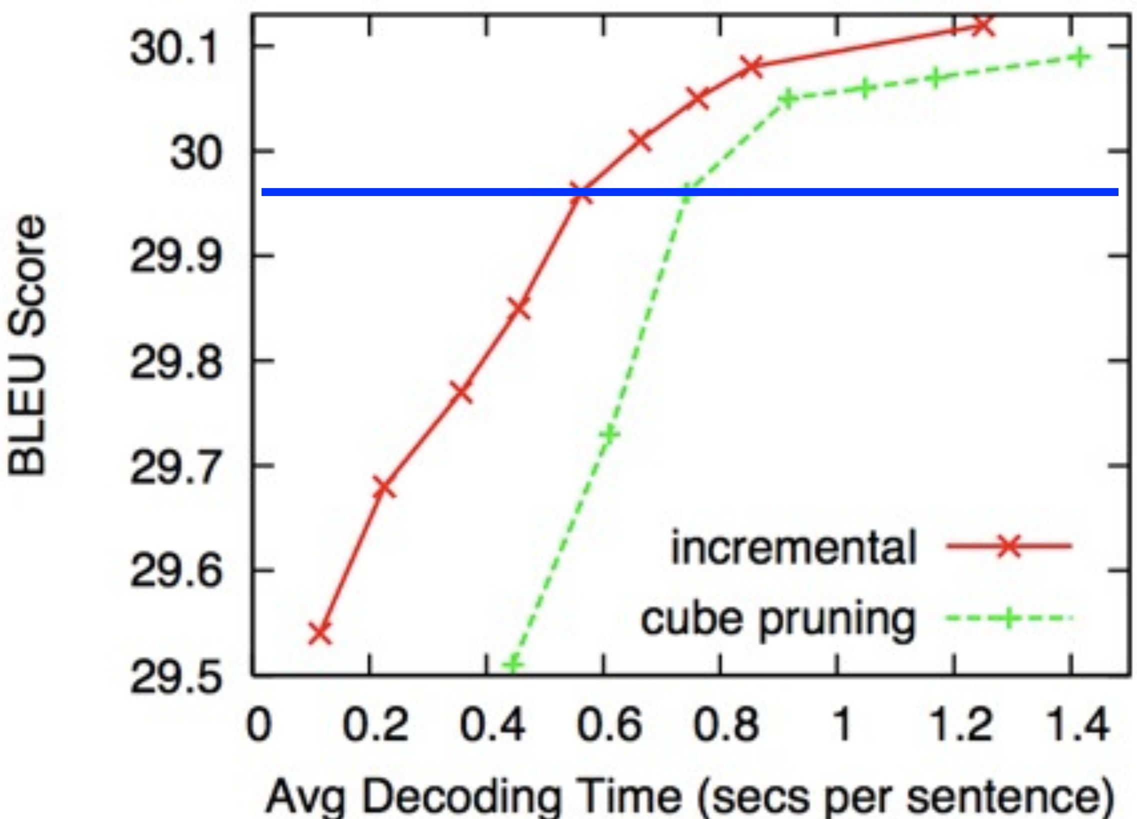




Comparison with Cube Pruning

- incremental is slightly faster than cube pruning
- note they are very different (orthogonal) techniques
  - we envision their combination will be even faster





Conclusion and Future Work

	in theory	*in practice*
phrase-based	exponential	quadratic
tree-to-string	polynomial	linear

- an incremental algorithm for tree-to-string translation
- linear-time in practice, and 30 times faster than Moses
- very different from cube pruning
  - cube pruning applies to phrase-based also (Huang/Chiang, 07)
  - future work 1: combine cube pruning w/ incremental
- future work 2: extend to other syntax-based models

非常 感谢！

非常 感谢！

Thank you very much ！

Tree Depth: Mean and Variance

- logarithmic mean and variance of tree-depth
- (needed for avg.-case polynomial-time complexity)