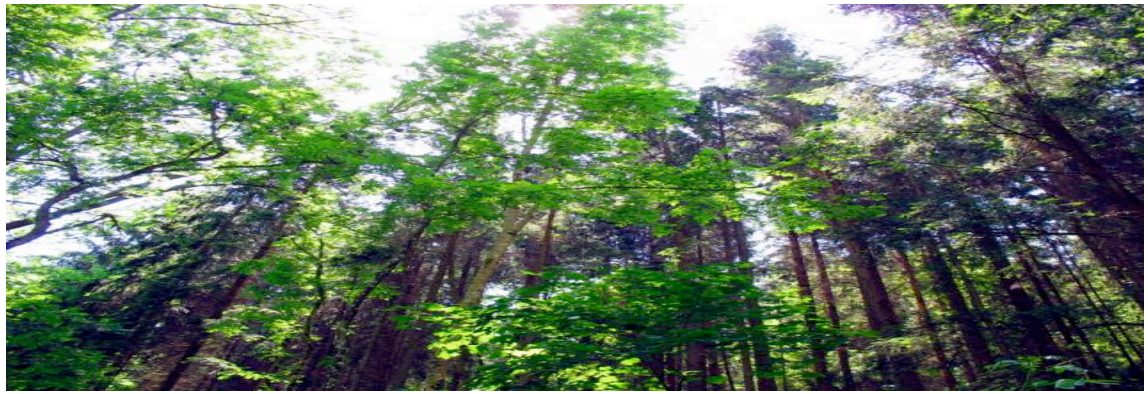
Forest-Based Translation Rule Extraction



Haitao Mi
Institute of Computing Technology

Liang Huang
University of Pennsylvania

EMNLP 2008 talk, Honolulu, Hawaii, October 2008

Translation Rule Extraction

- rule extraction is a central problem in Statistical MT
- especially in linguistically syntax-based systems
  - use parse trees ("syntax") from either or both sides
  - more informed translation thanks to syn. categories
  - but in practice worse than formal syntax only (Hiero)

source target	examples (partial)	type
tree-to-tree	Ding and Palmer (2006)	linguistic syntax
tree-to-string	Liu et al. (2006); Huang et al. (2006)	
string-to-tree	Galley et al. (2006)	
string-to-string	Chiang (2005)	formal

How to learn better rules?

- one major problem: parsing error affects rule set quality
  - *k*-best trees? limited scope; too slow cf. (Venugopal et al 2008)
  - we use packed forest of exponentially many trees
  - result: 2.5 BLEU final improvement; better than Hiero
  - experiments focused on tree-to-string systems

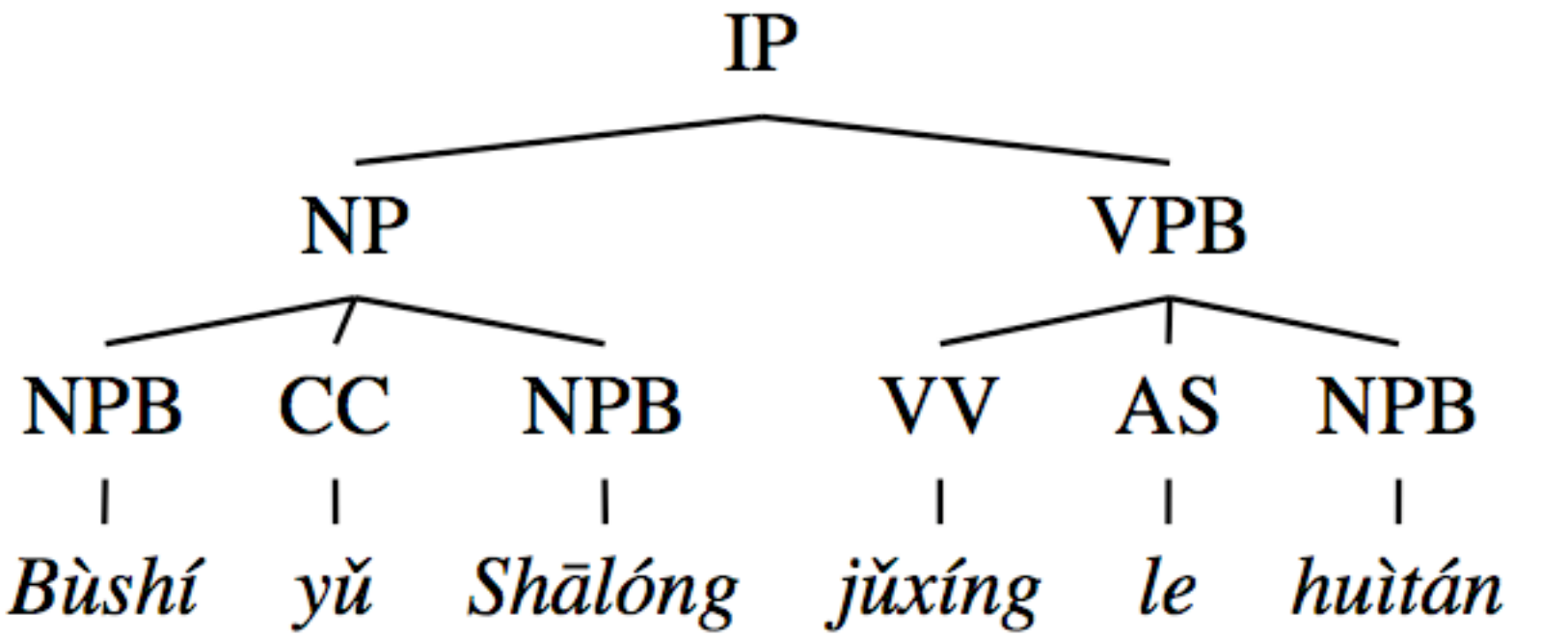
source target	examples (partial)	type
tree-to-tree	Ding and Palmer (2006)	linguistic syntax
tree-to-string	Liu et al. (2006); Huang et al. (2006)	
string-to-tree	Galley et al. (2006)	
string-to-string	Chiang (2005)	formal

Outline

- Background: Tree-based Translation
  - Basic Rule Extraction Algorithm (Galley et al., 2004)
- Forest-based Rule Extraction
- Related Work
- Experiments

Tree-based Translation

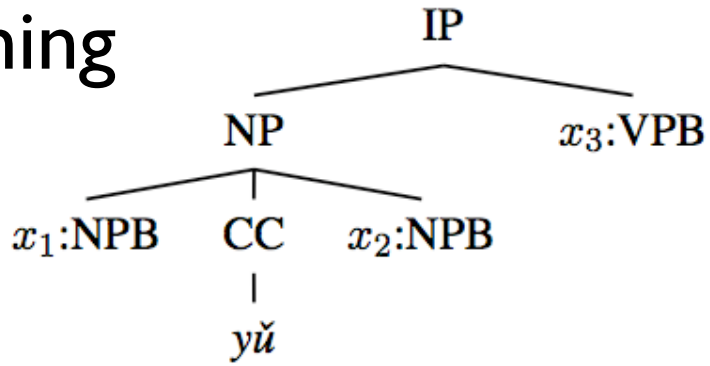
- get 1-best parse tree; then convert to English

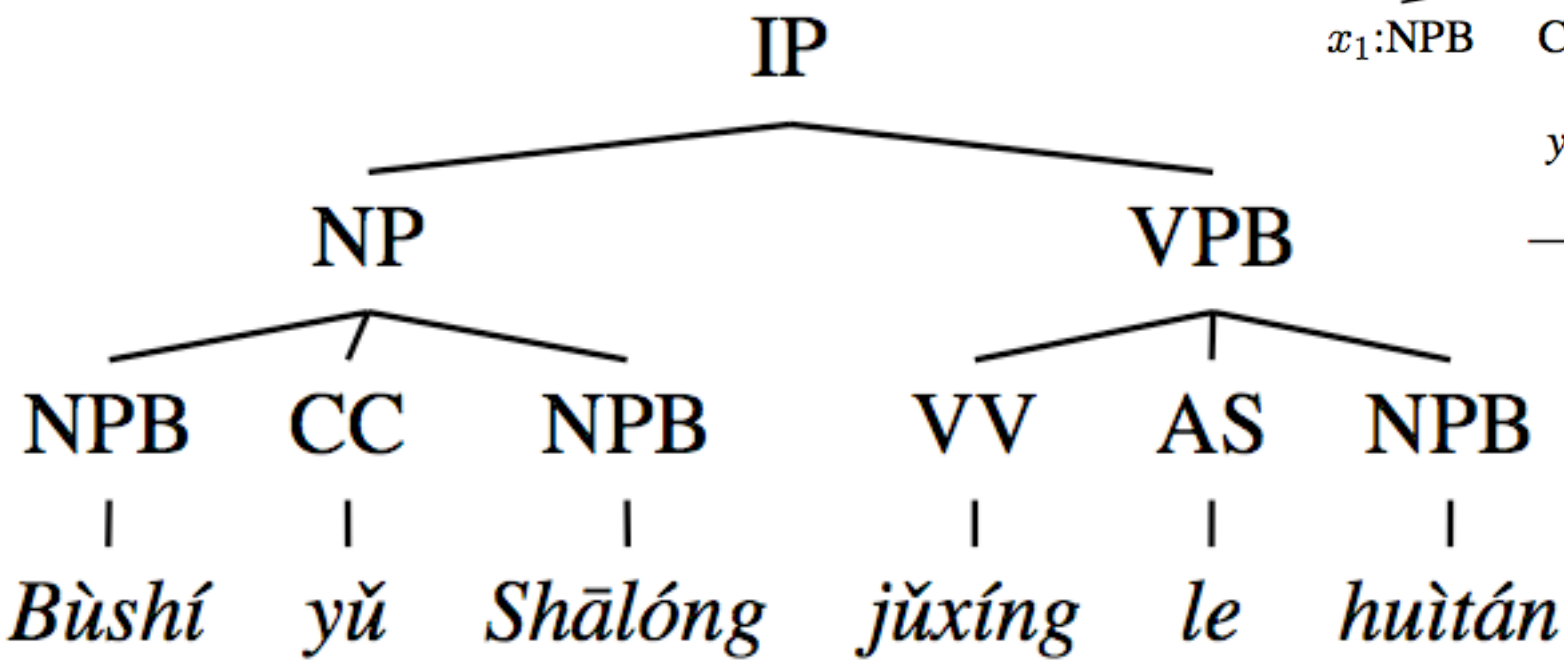


"Bush held a meeting with Sharon"

Tree-based Translation

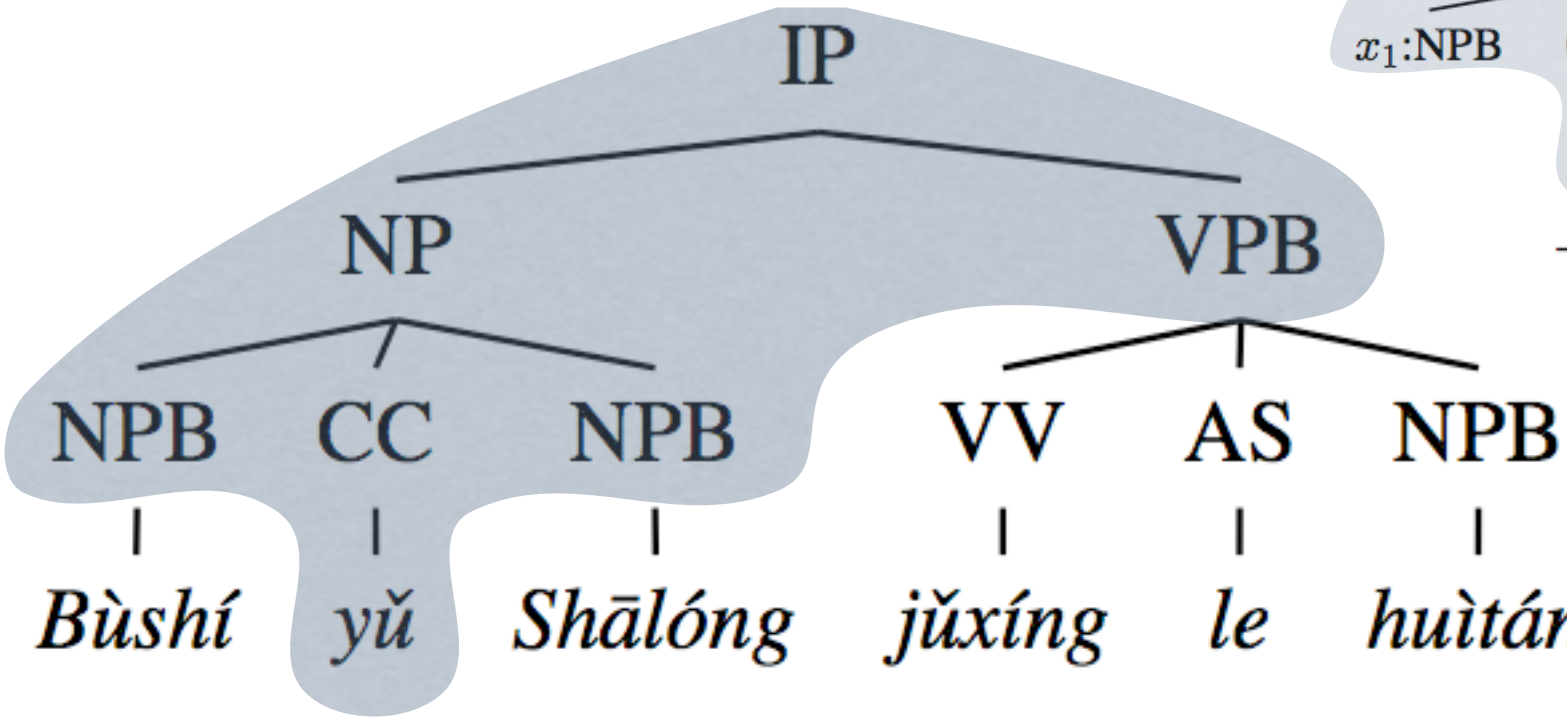
- recursive rewrite by pattern-matching

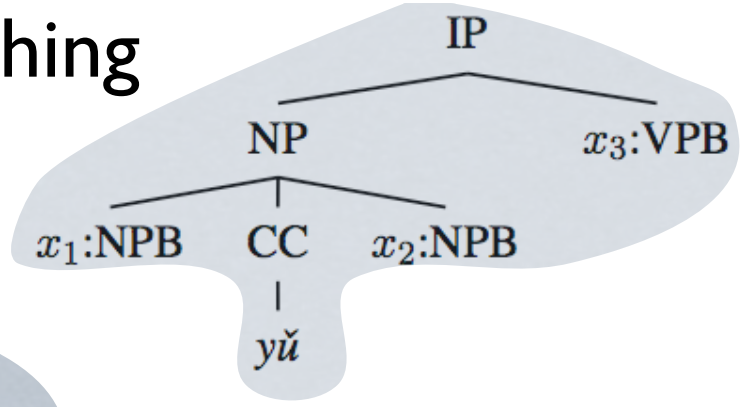




Tree-based Translation

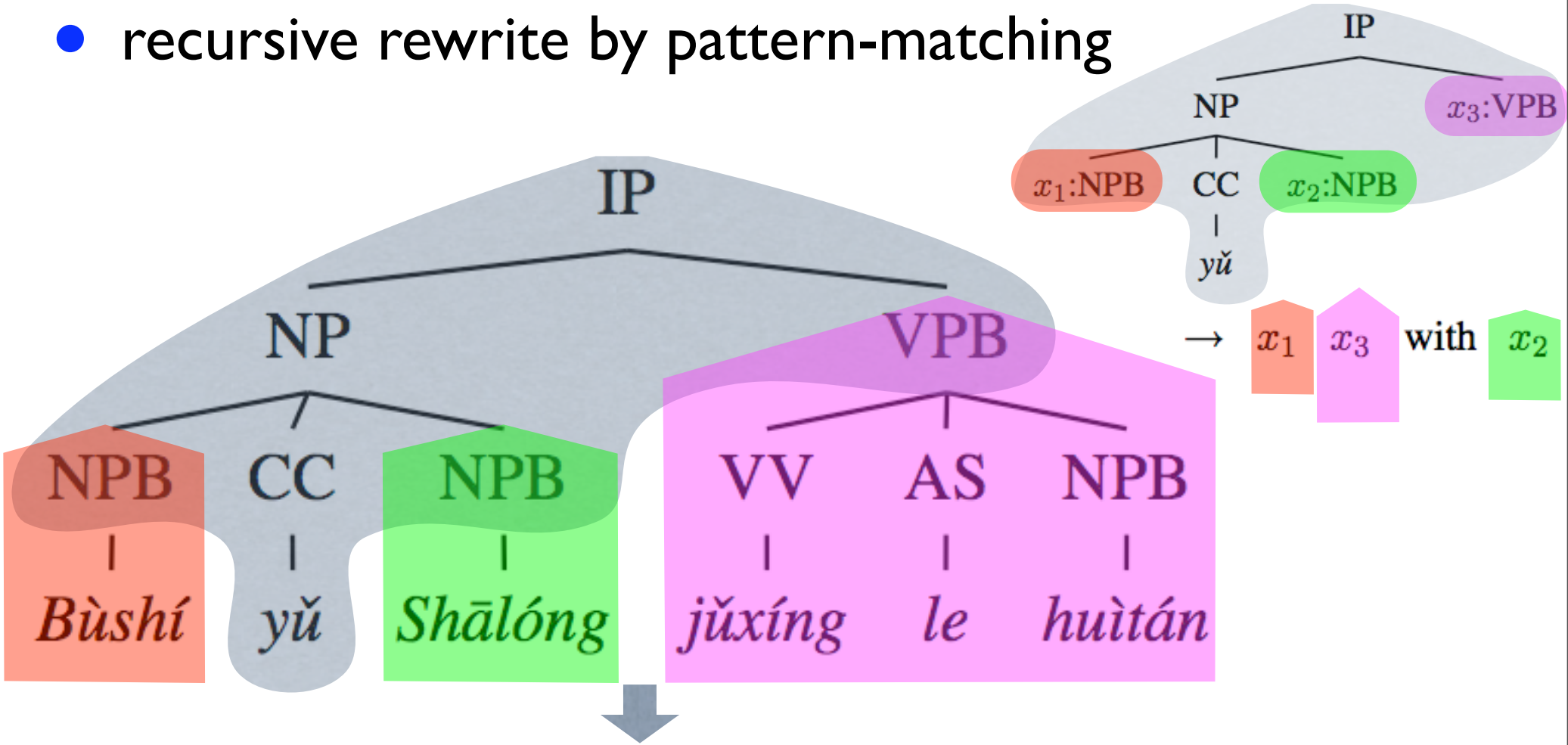
- recursive rewrite by pattern-matching





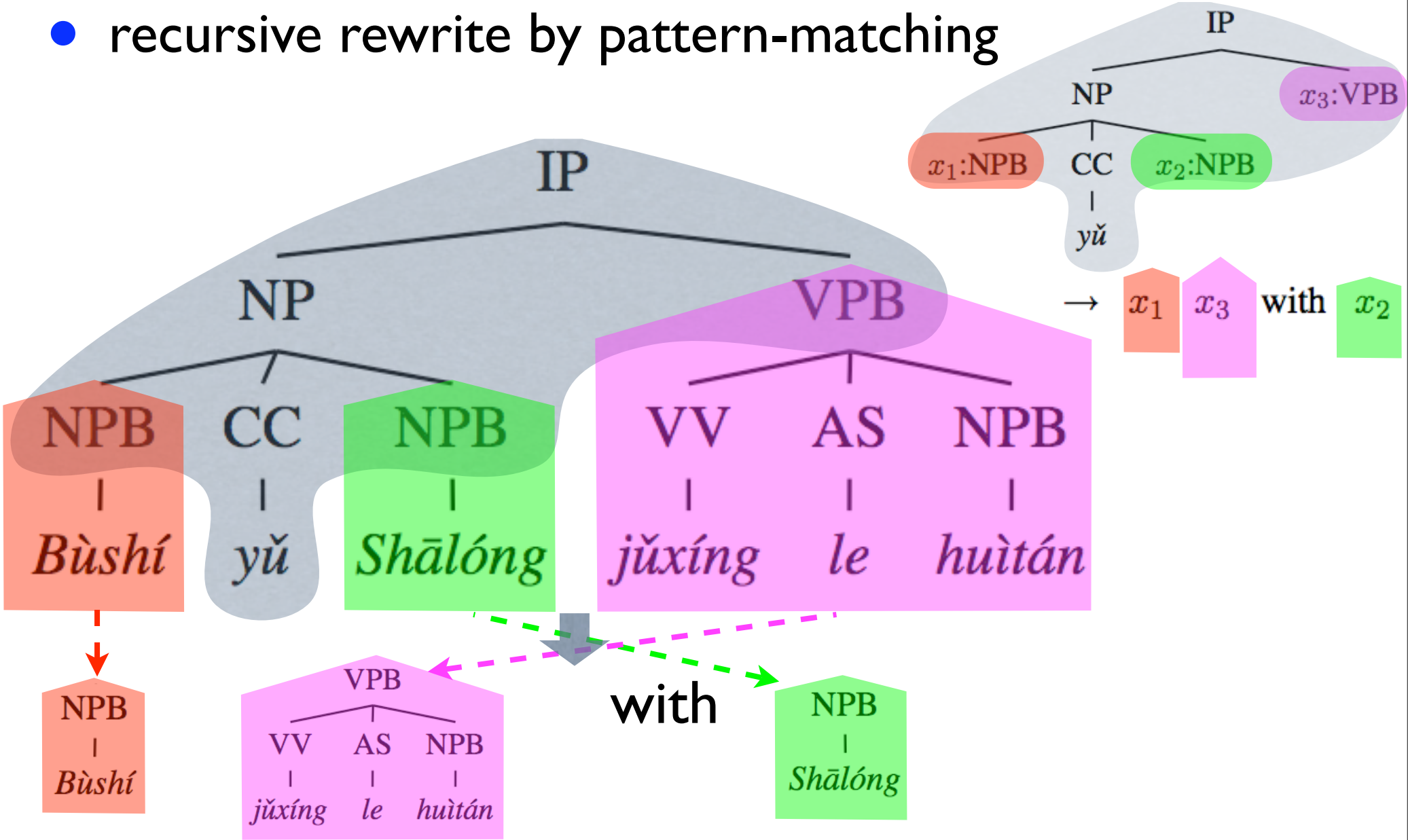
Tree-based Translation

- recursive rewrite by pattern-matching



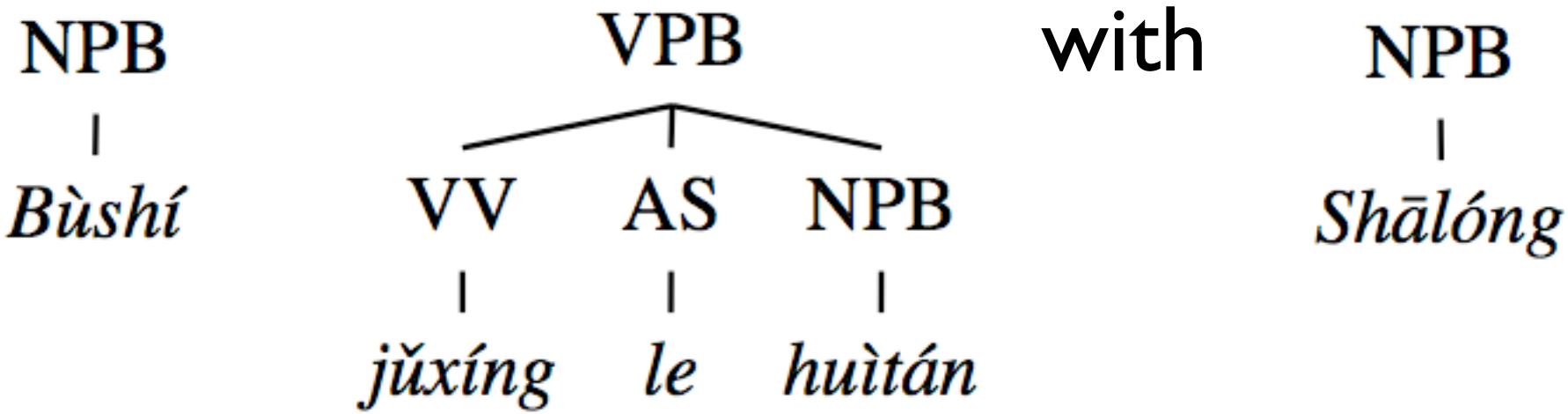
Tree-based Translation

- recursive rewrite by pattern-matching



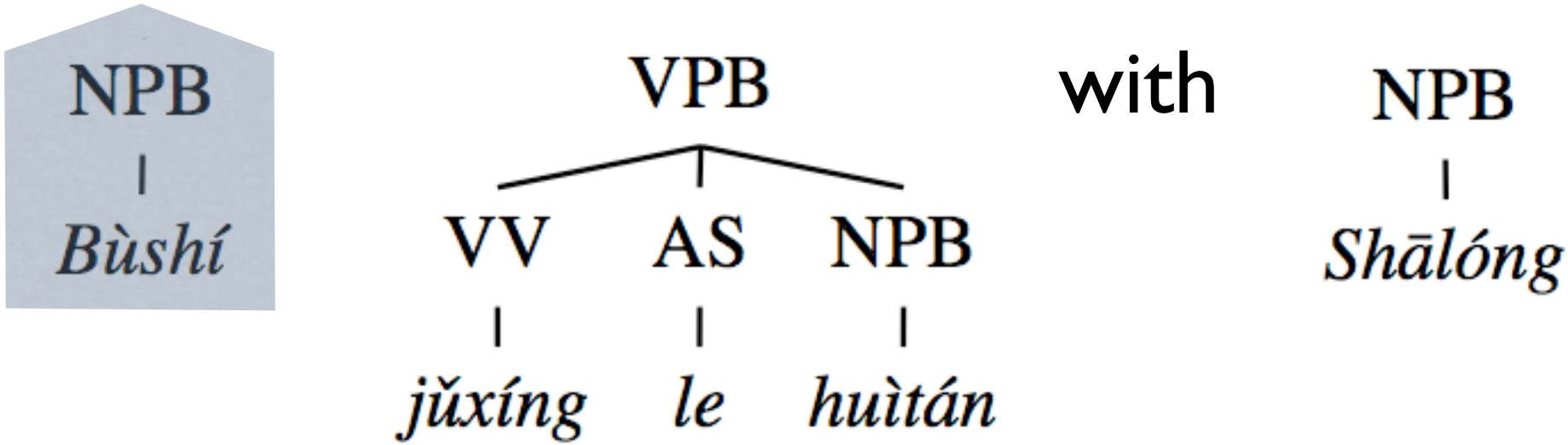
Tree-based Translation

- recursively solve unfinished subproblems



Tree-based Translation

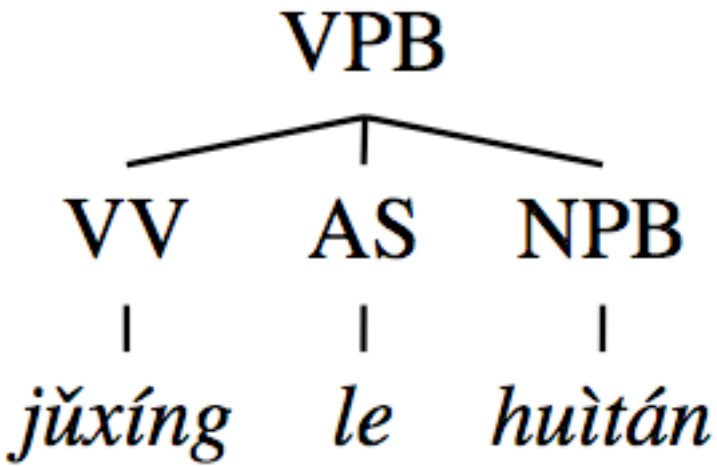
- recursively solve unfinished subproblems



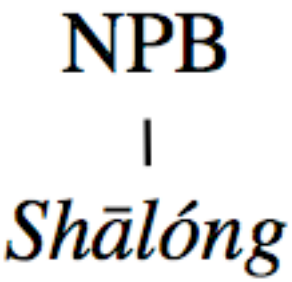
Tree-based Translation

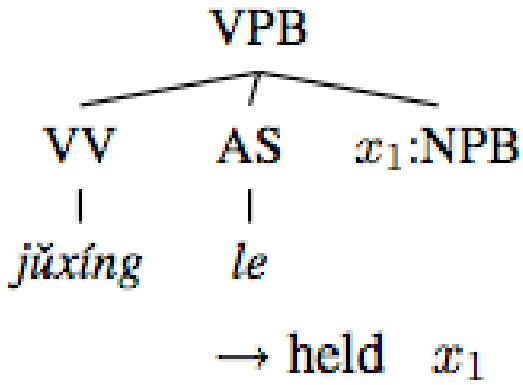
- recursively solve unfinished subproblems

Bush



with

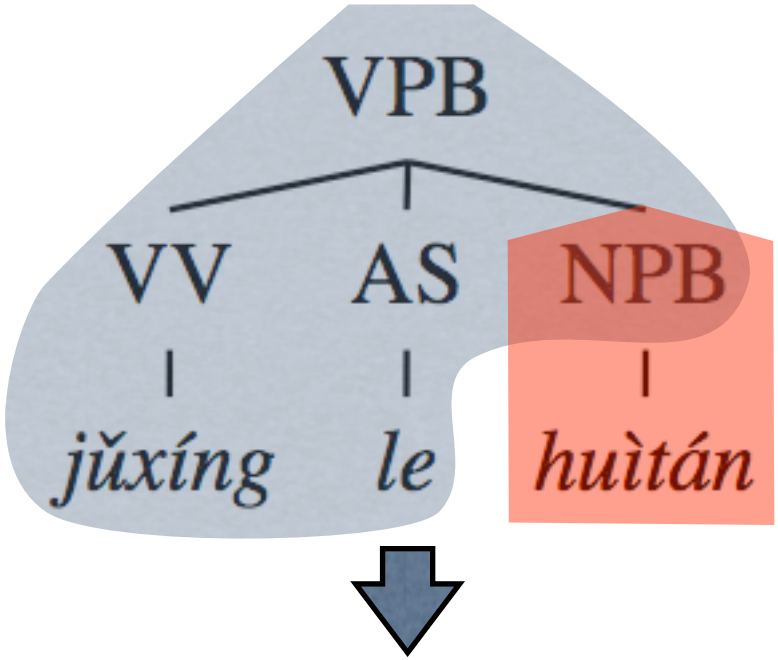




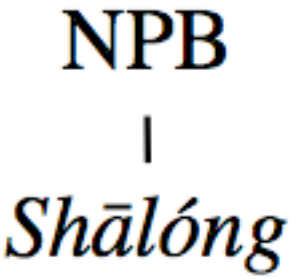
Tree-based Translation

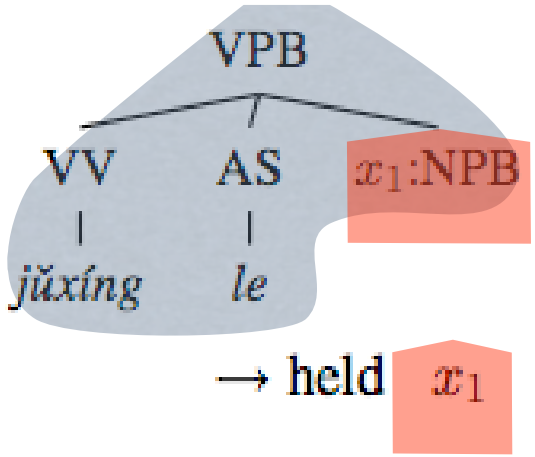
- recursively solve unfinished subproblems

Bush



with

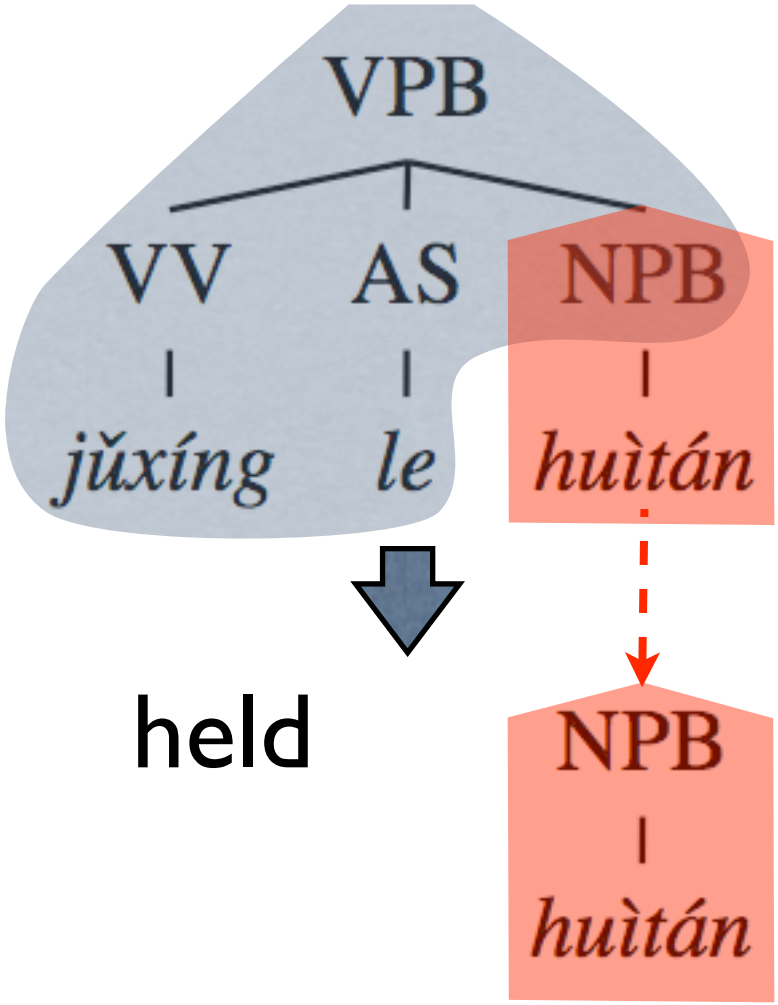




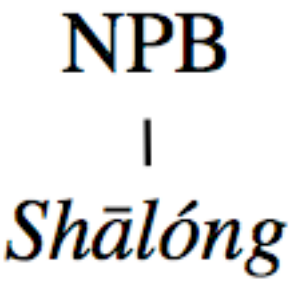
Tree-based Translation

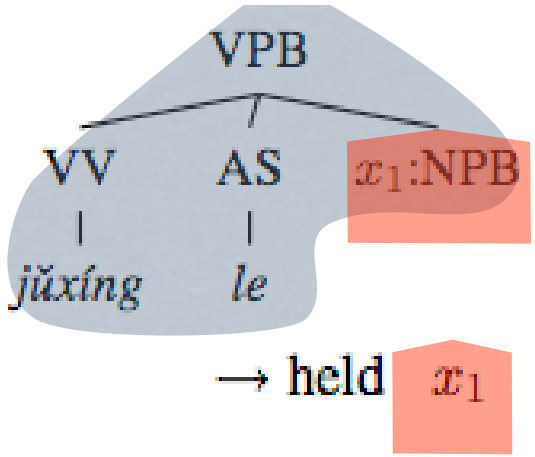
- recursively solve unfinished subproblems

Bush



with





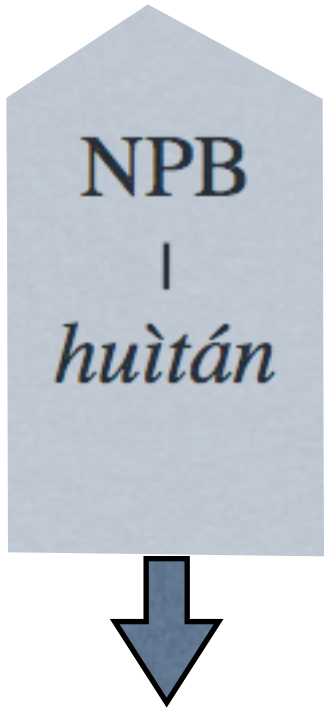
Tree-based Translation

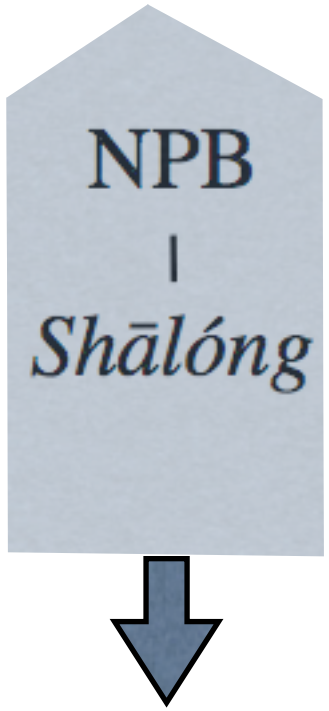
- continue pattern-matching

Bush held NPB | *huìtán* with NPB | *Shālóng*

Tree-based Translation

- continue pattern-matching

Bush held 

with 

a meeting

Sharon

Tree-based Translation

- continue pattern-matching

Bush held a meeting with Sharon

Tree-based Translation

- continue pattern-matching

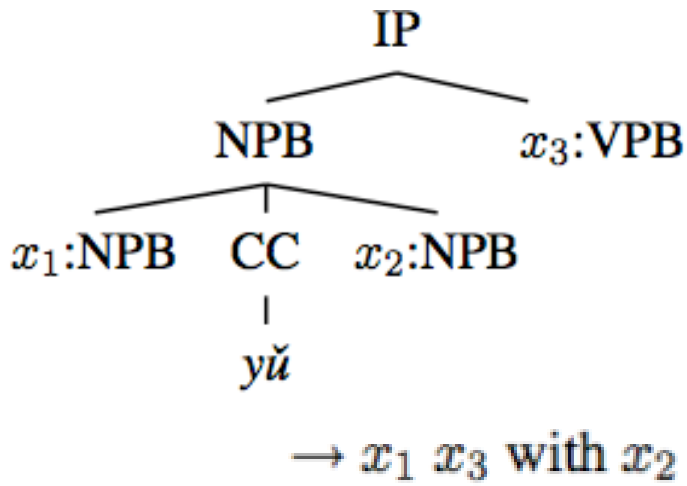
Bush held a meeting with Sharon

pros: simplicity: separate parsing and decoding (fast!)

expressive grammar,

"extended domain of locality"

...



(Galley et al. 2004; Huang, Knight, Joshi 2006)

Tree-based Translation

- continue pattern-matching

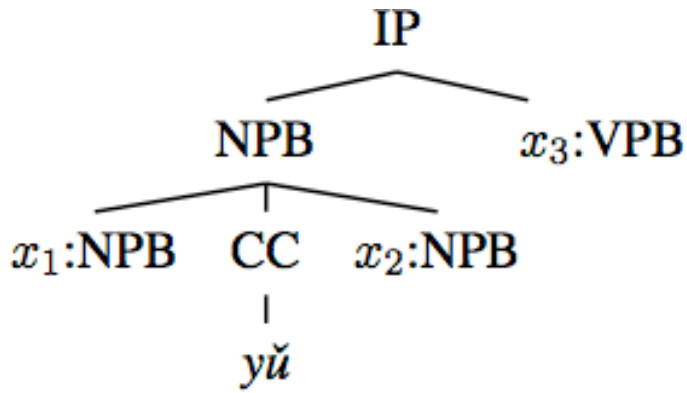
Bush held a meeting with Sharon

pros: simplicity: separate parsing and decoding (fast!)

expressive grammar,

"extended domain of locality"

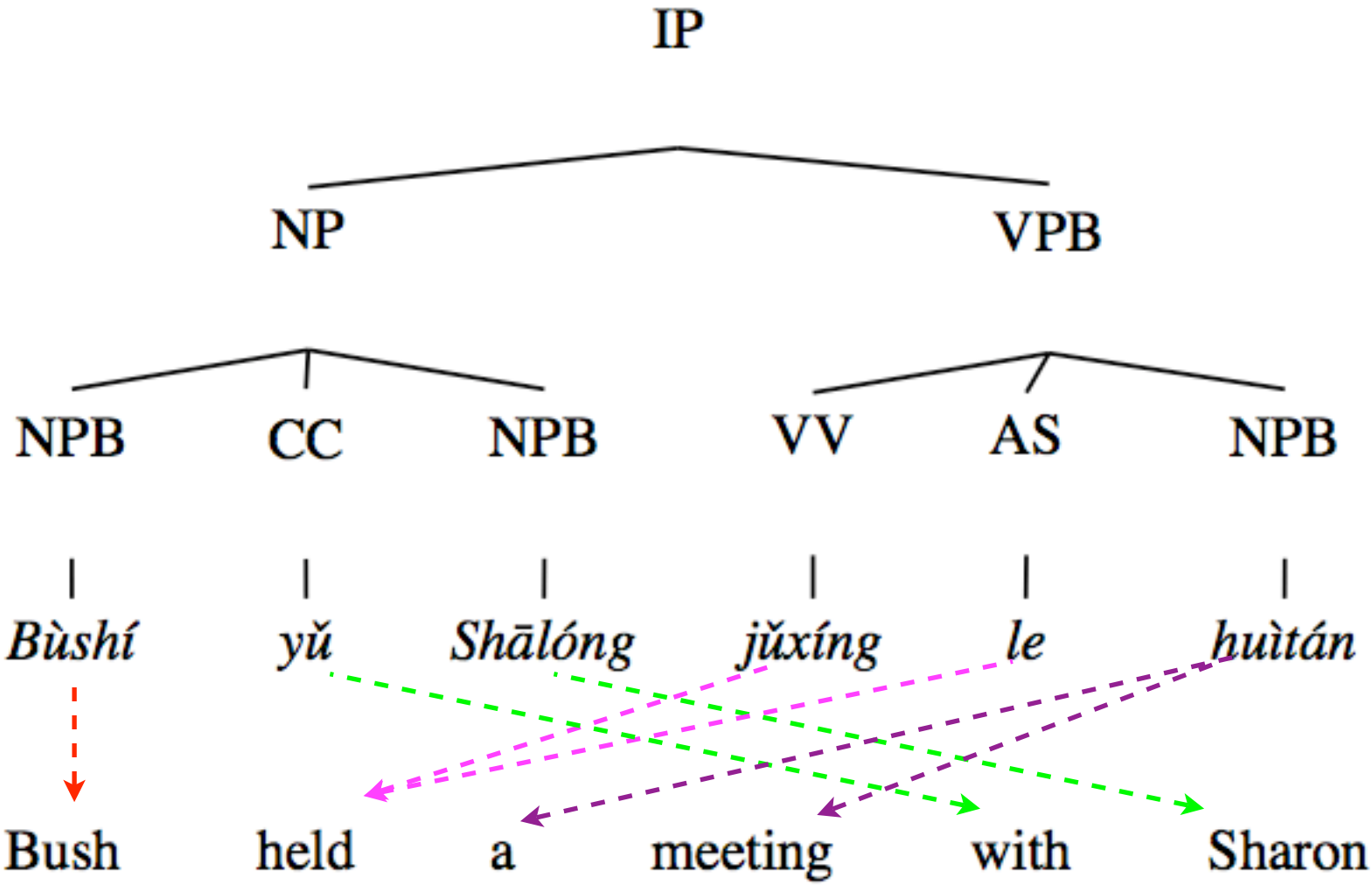
...



Q: where are the rules from?

Where are the rules from?

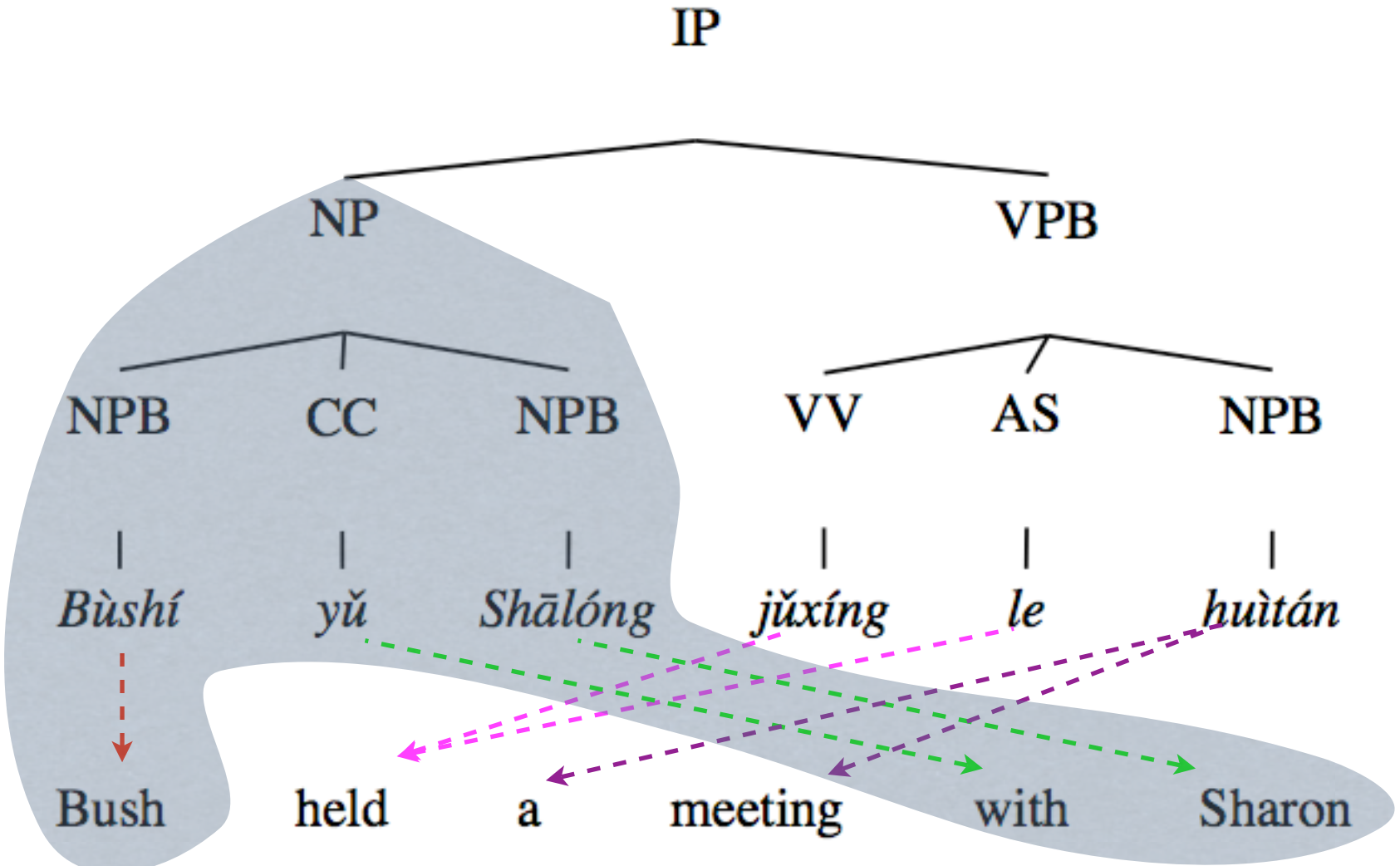
- source parse tree, target sentence, and alignment
- intuition: contiguous span



GHKM - (Galley et al 2004; 2006)

Where are the rules from?

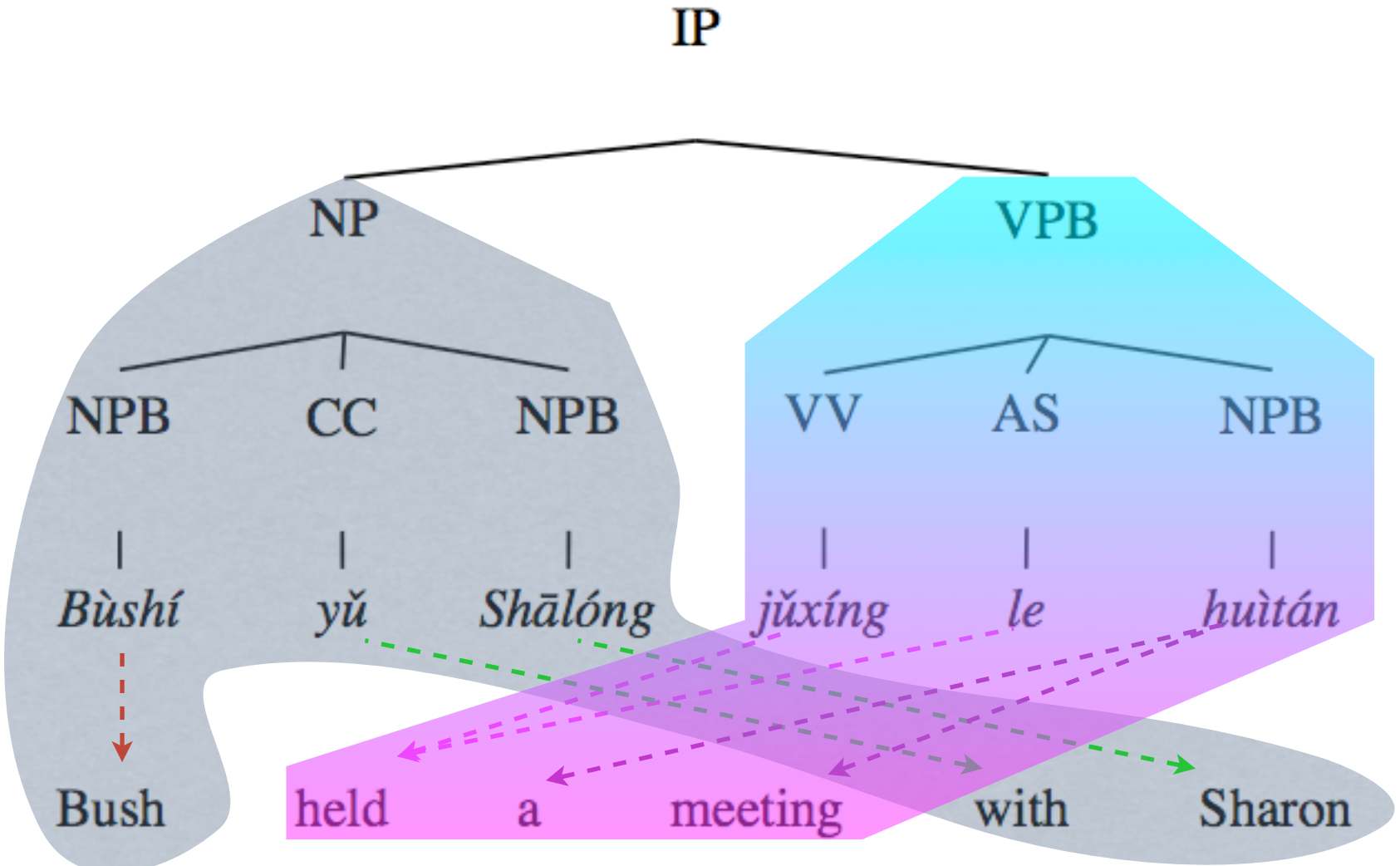
- source parse tree, target sentence, and alignment
- intuition: contiguous span



GHKM - (Galley et al 2004; 2006)

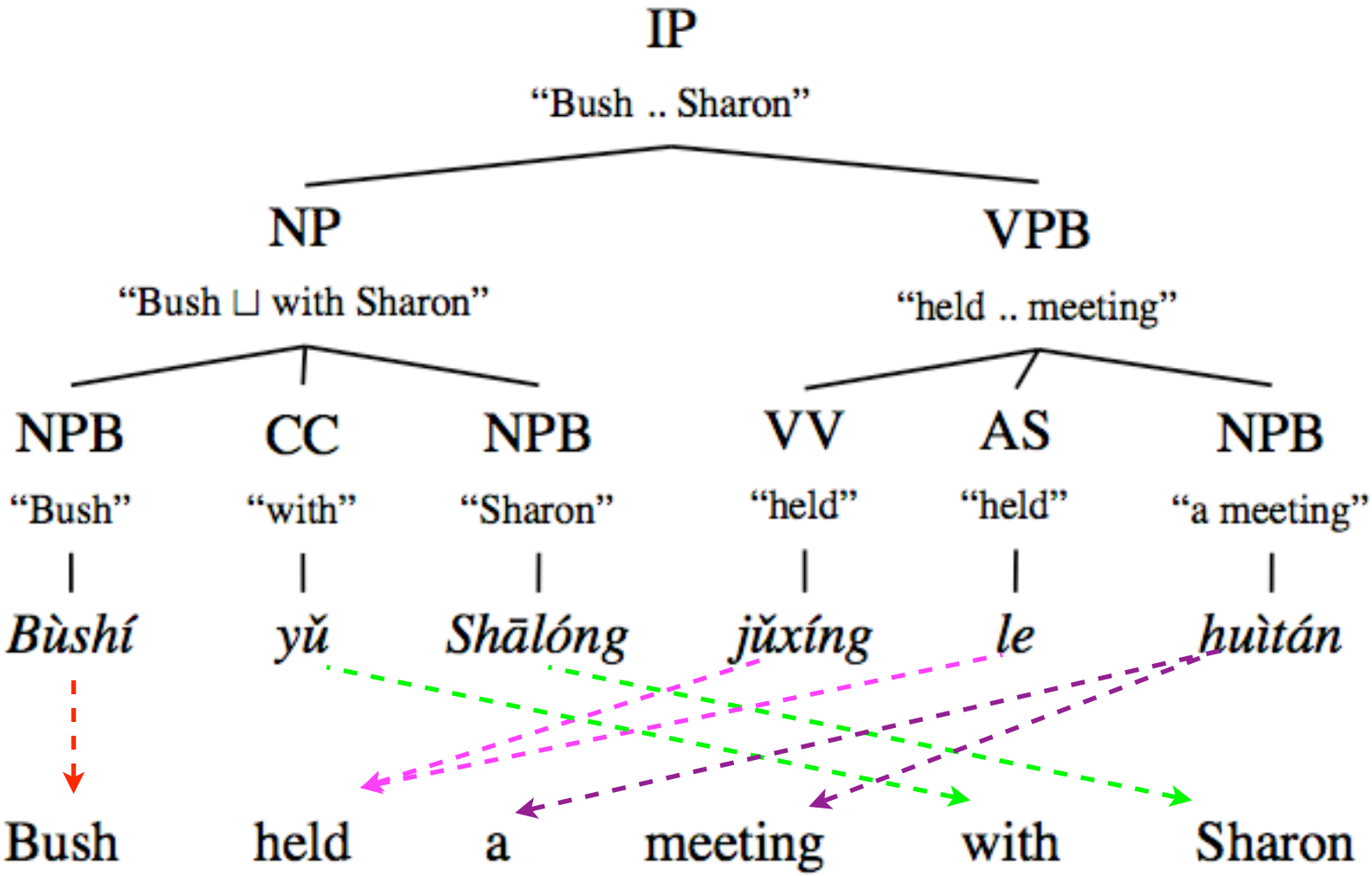
Where are the rules from?

- source parse tree, target sentence, and alignment
- intuition: contiguous span



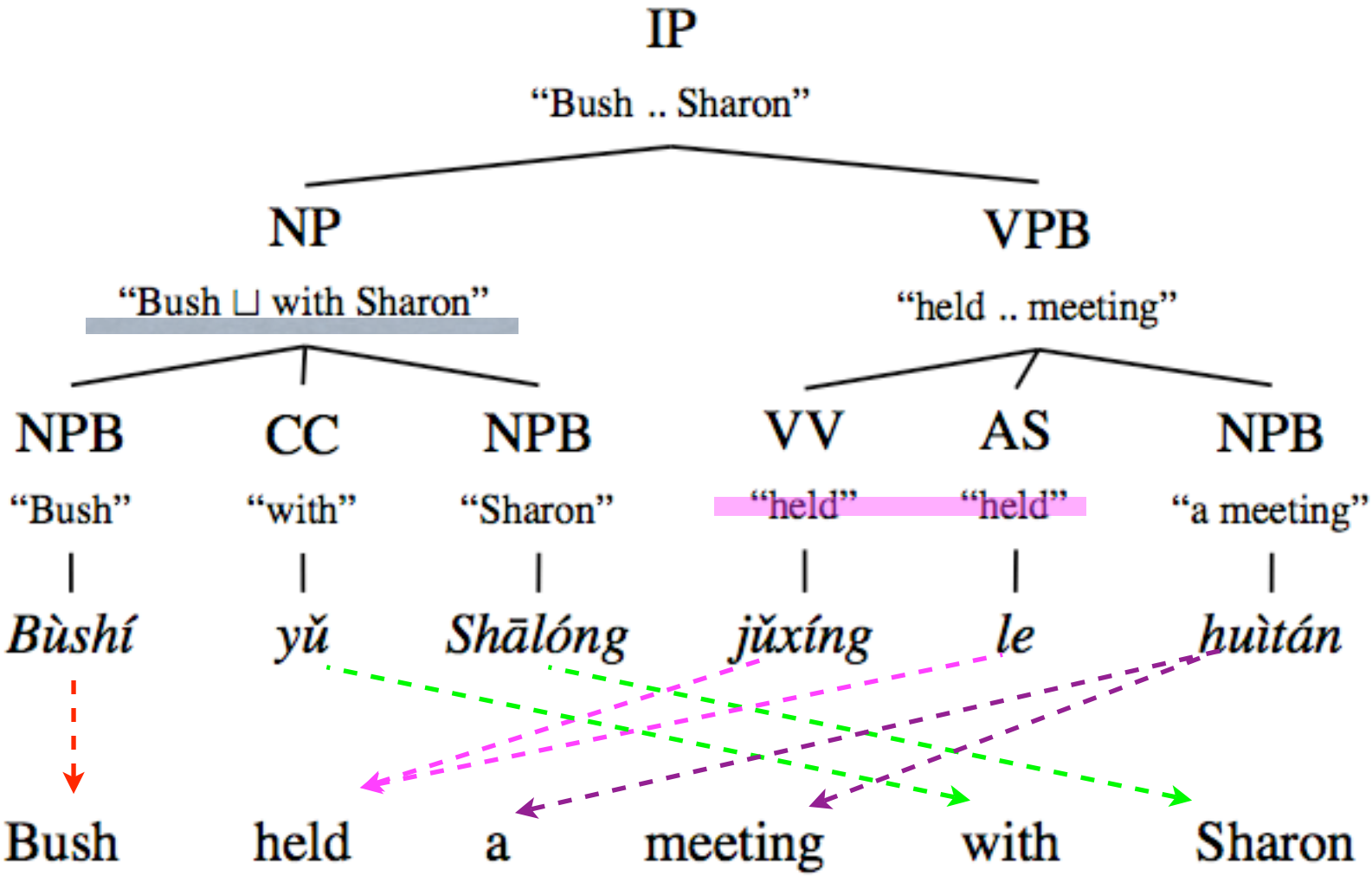
Where are the rules from?

- source parse tree, target sentence, and alignment
- compute target spans



Where are the rules from?

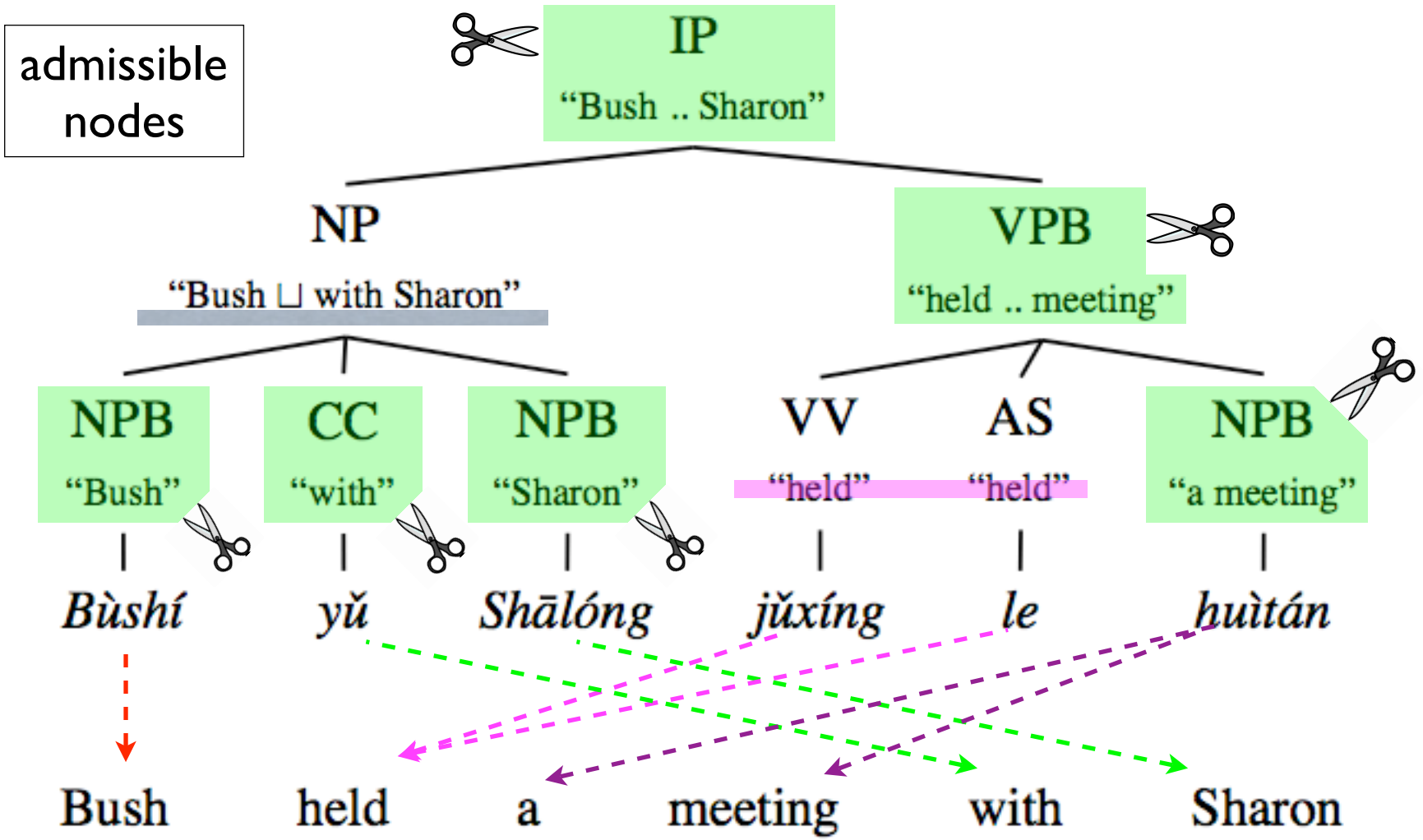
- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span



GHKM - (Galley et al 2004; 2006)

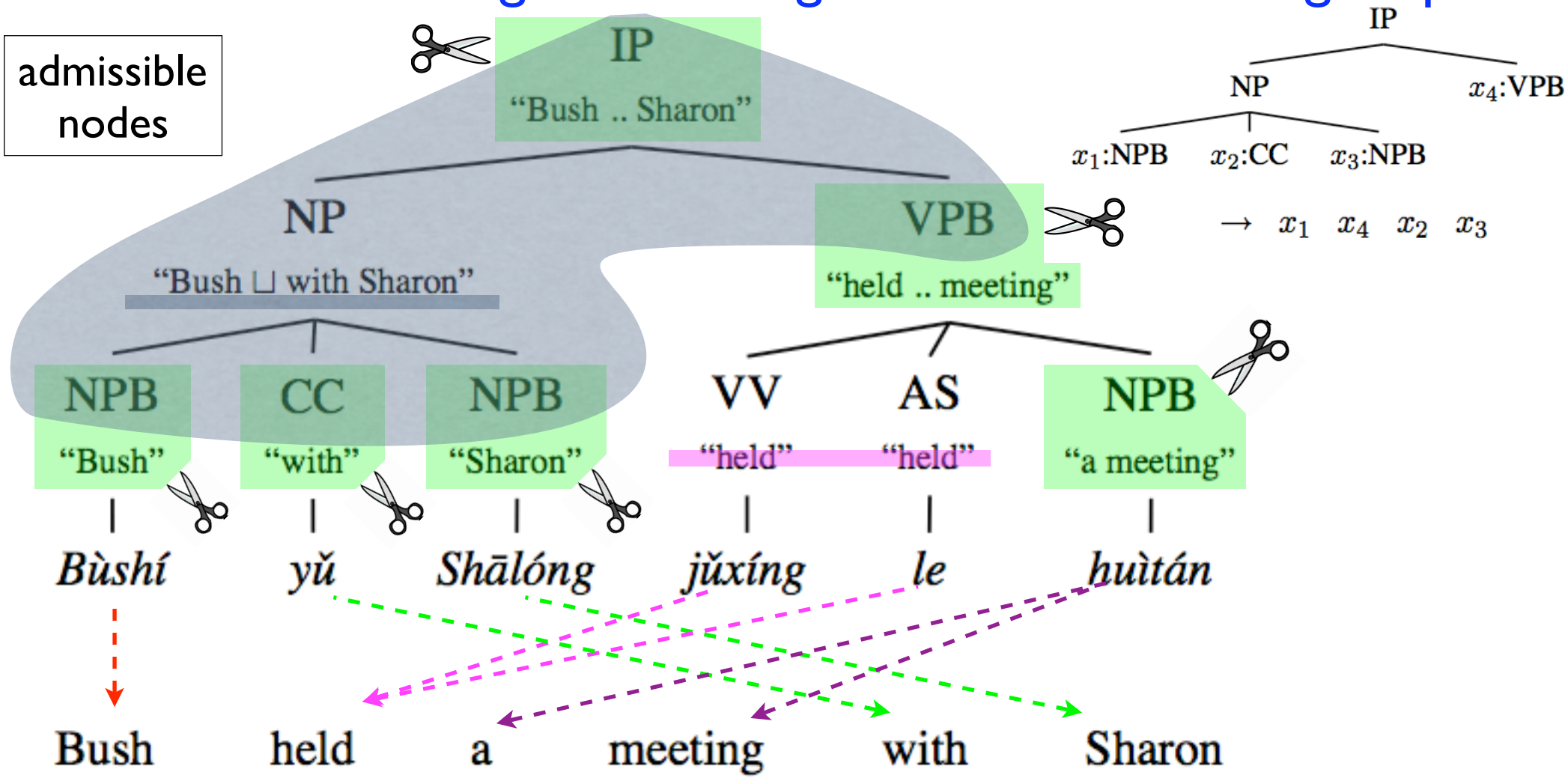
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span



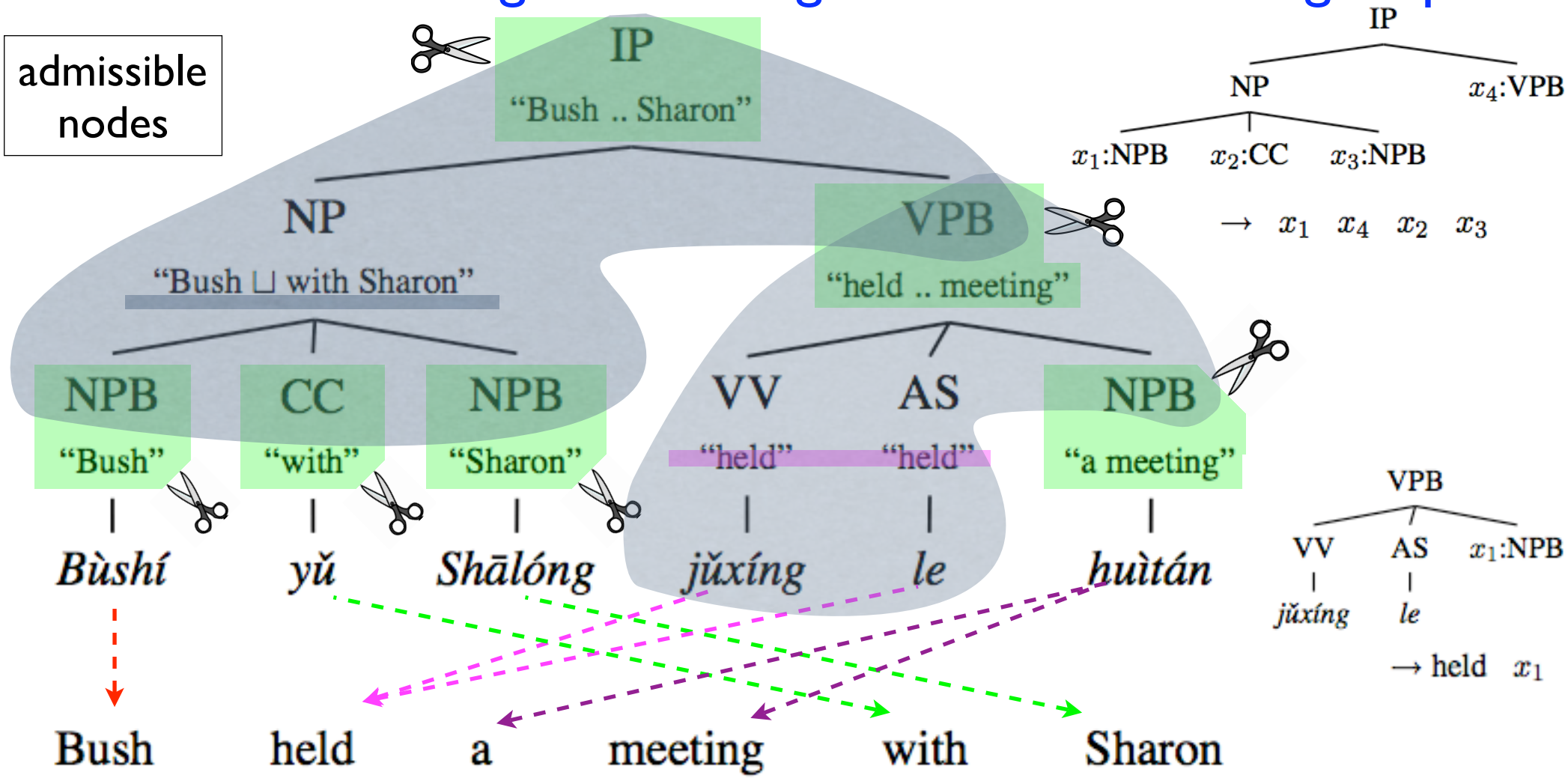
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span



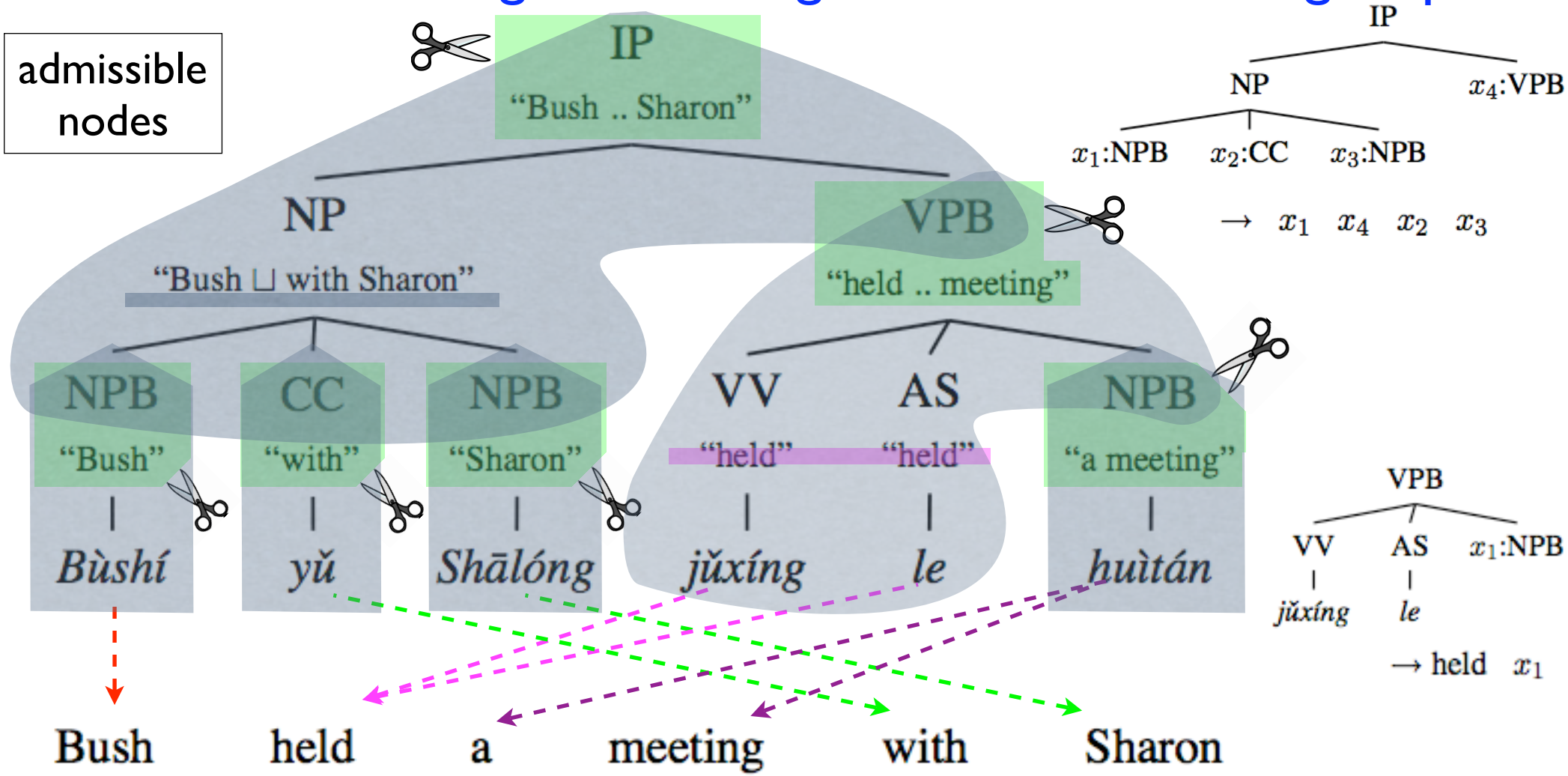
Where are the rules from?

- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span

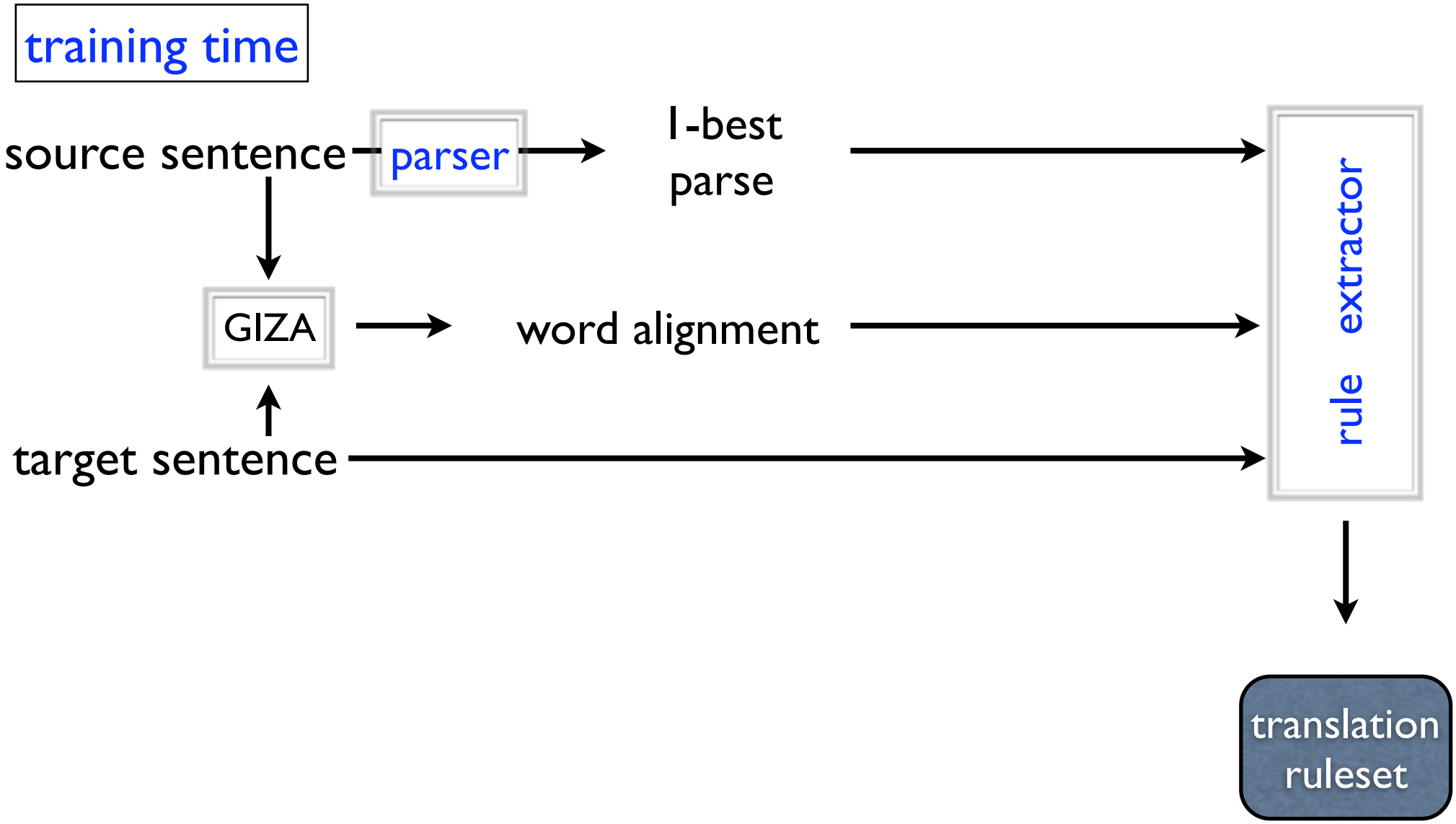


Where are the rules from?

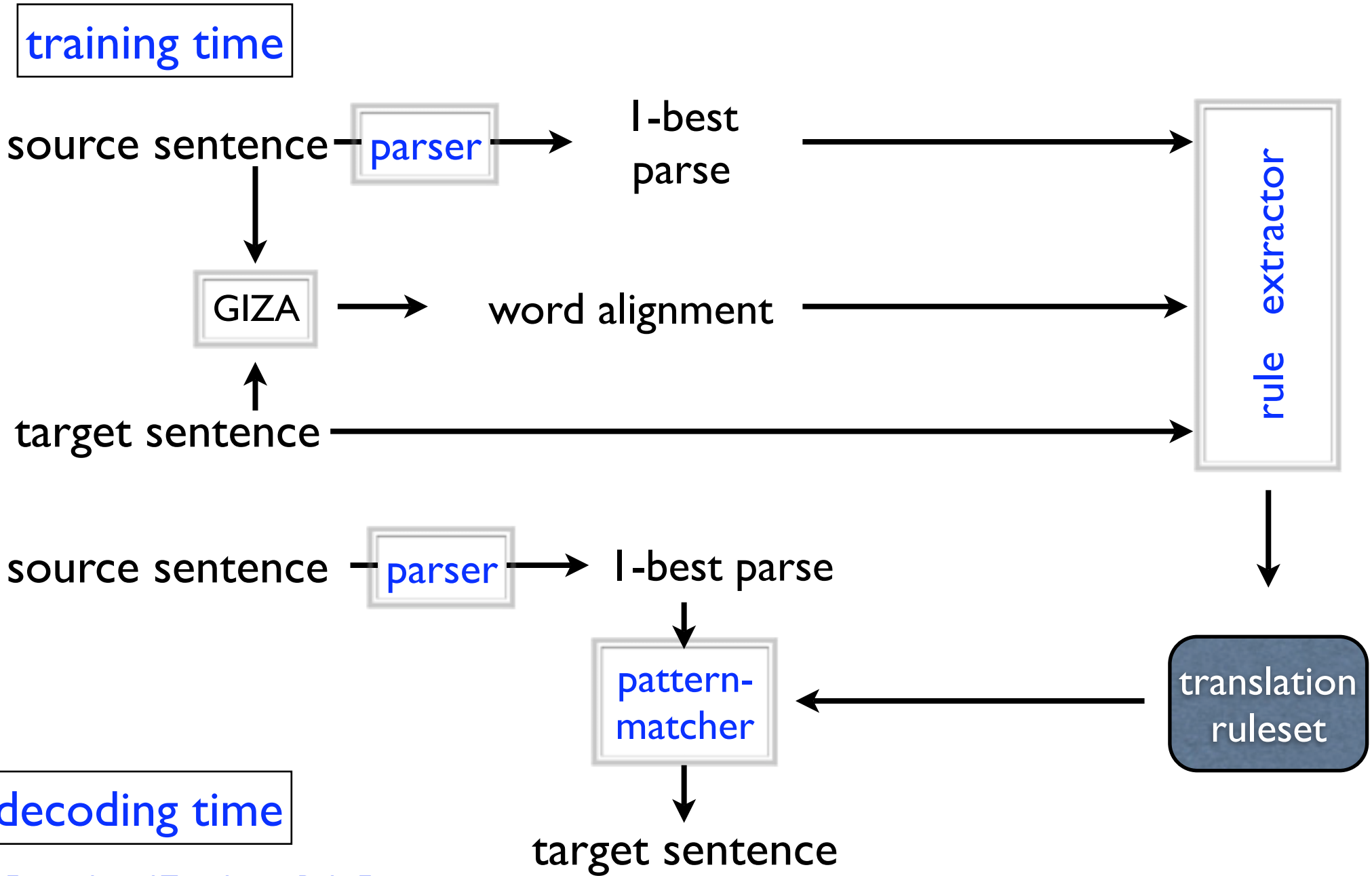
- source parse tree, target sentence, and alignment
- well-formed fragment: contiguous and faithful target-span



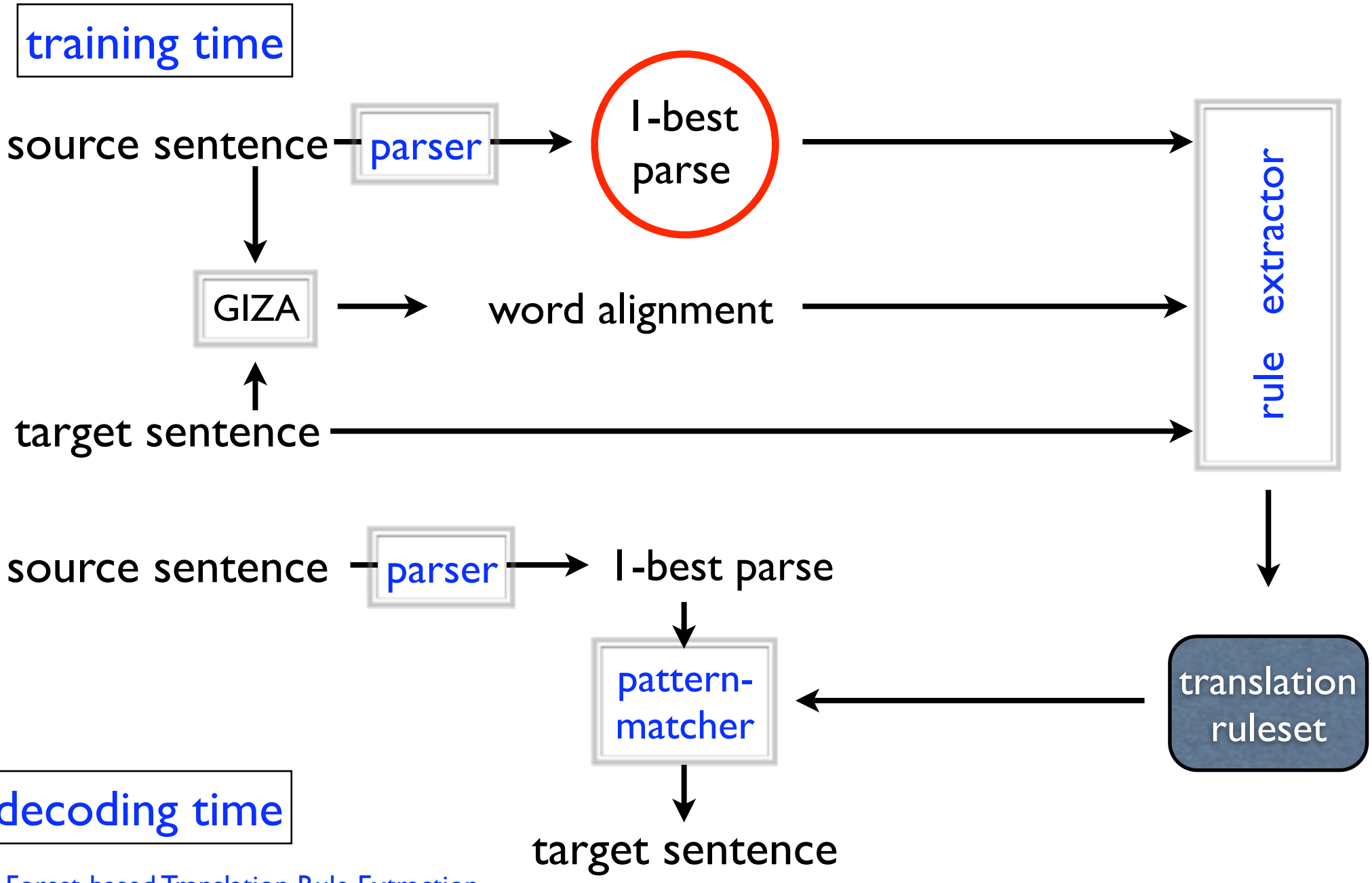
The Baseline Pipeline



The Baseline Pipeline



The Baseline Pipeline

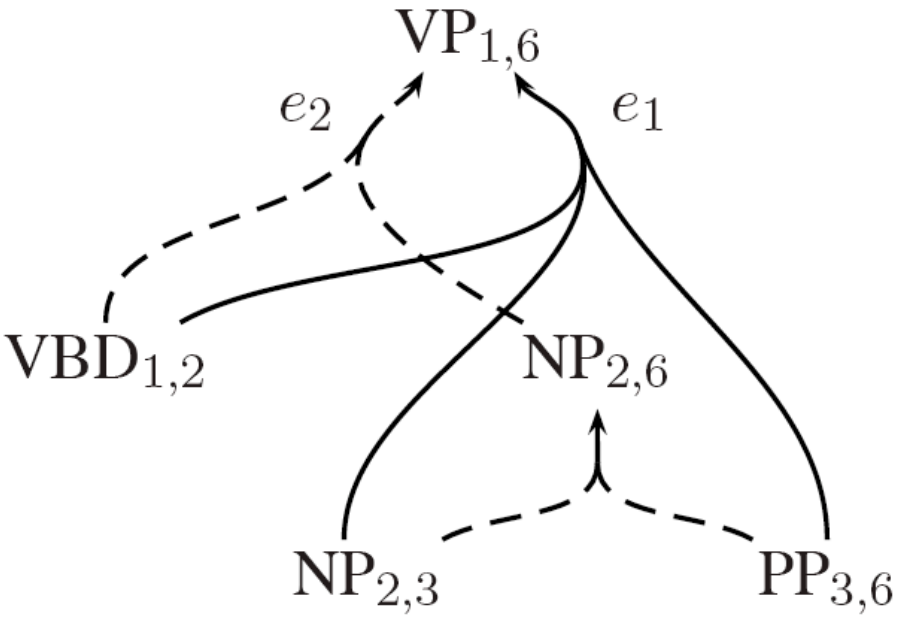


Outline

- Background: Tree-based Translation and Rule Extraction
- Forest-based Rule Extraction
  - Background: Parse Forest
  - Forest-based Extraction
  - Inside-Outside Forest Pruning
  - Fractional Rule Counts
- Related Work
- Experiments

Packed Forest

- a compact representation of many parses
  - by sharing common sub-derivations
  - polynomial-space encoding of exponentially large set

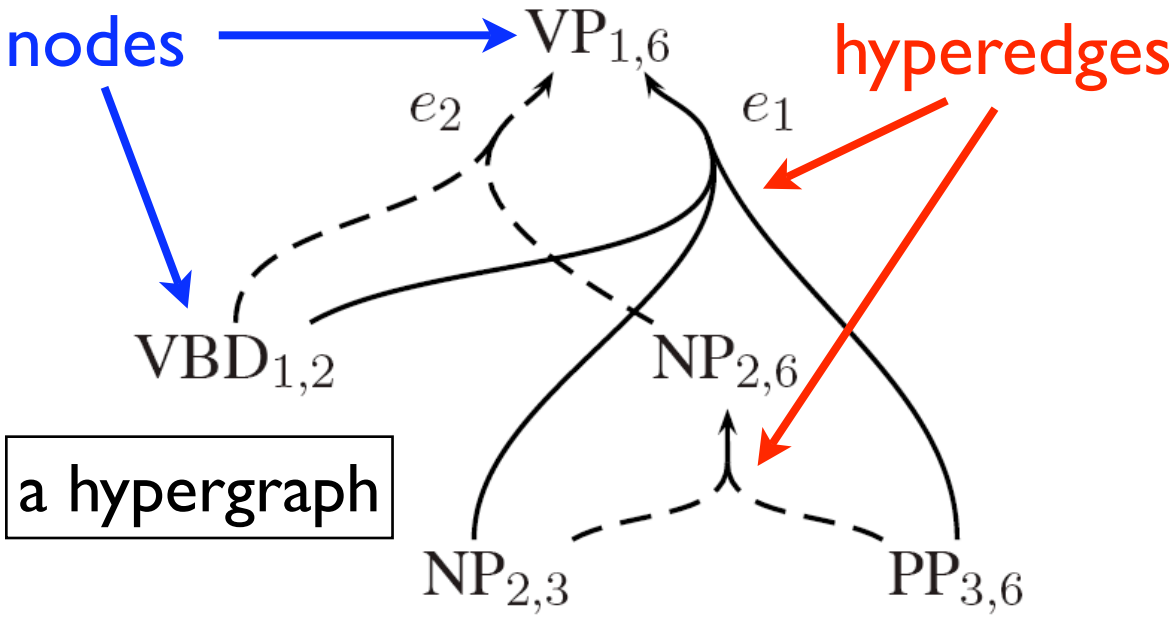


$$e_1 \quad \frac{\mathrm{VBD}_{1,2} \quad \mathrm{NP}_{2,3} \quad \mathrm{PP}_{3,6}}{\mathrm{VP}_{1,6}}$$

0 I 1 saw 2 him 3 with 4 a 5 mirror 6

Packed Forest

- a compact representation of many parses
  - by sharing common sub-derivations
  - polynomial-space encoding of exponentially large set

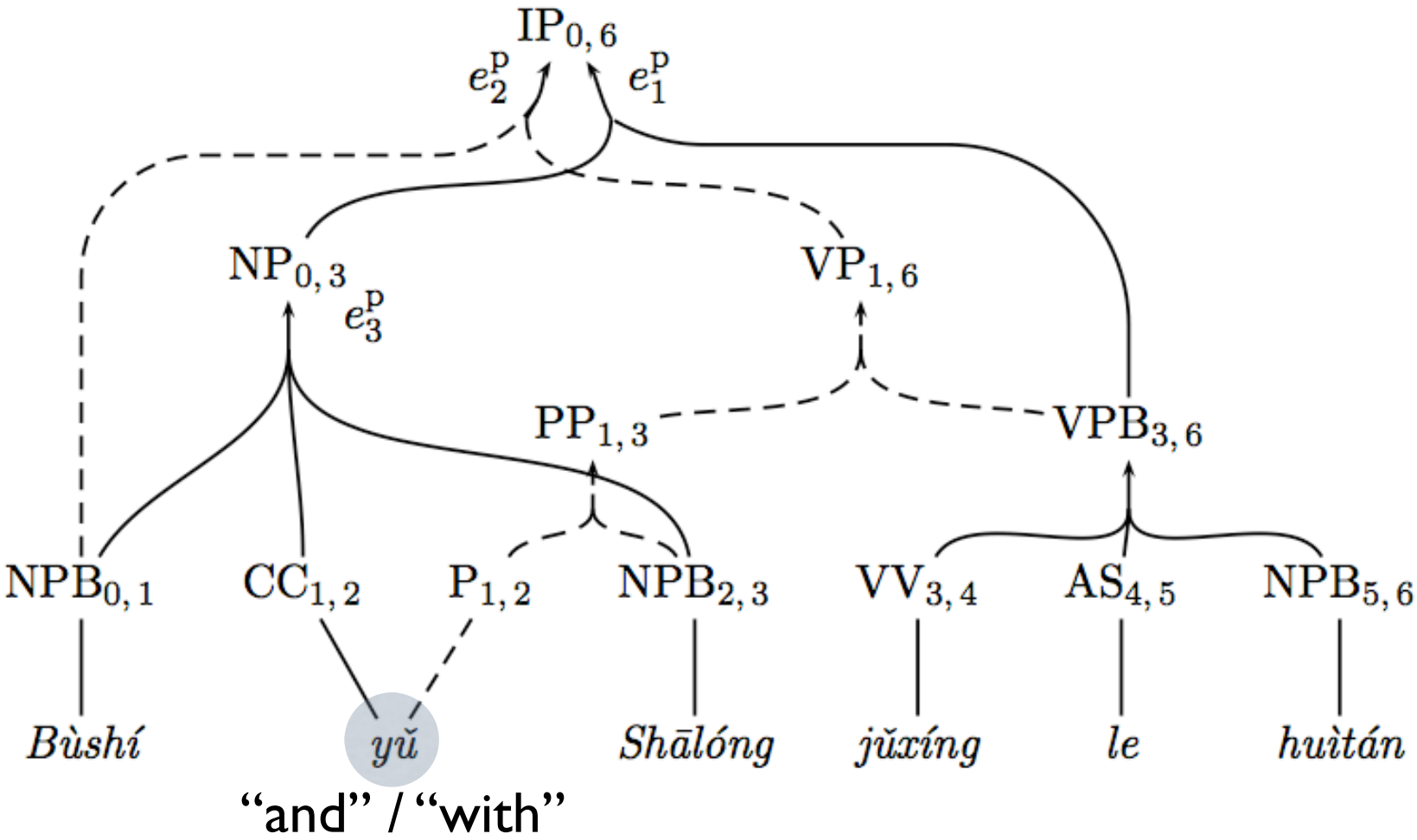


$$e_1 \quad \frac{\mathrm{VBD}_{1,2} \quad \mathrm{NP}_{2,3} \quad \mathrm{PP}_{3,6}}{\mathrm{VP}_{1,6}}$$

$_0$ I $_1$ saw $_2$ him $_3$ with $_4$ a $_5$ mirror $_6$

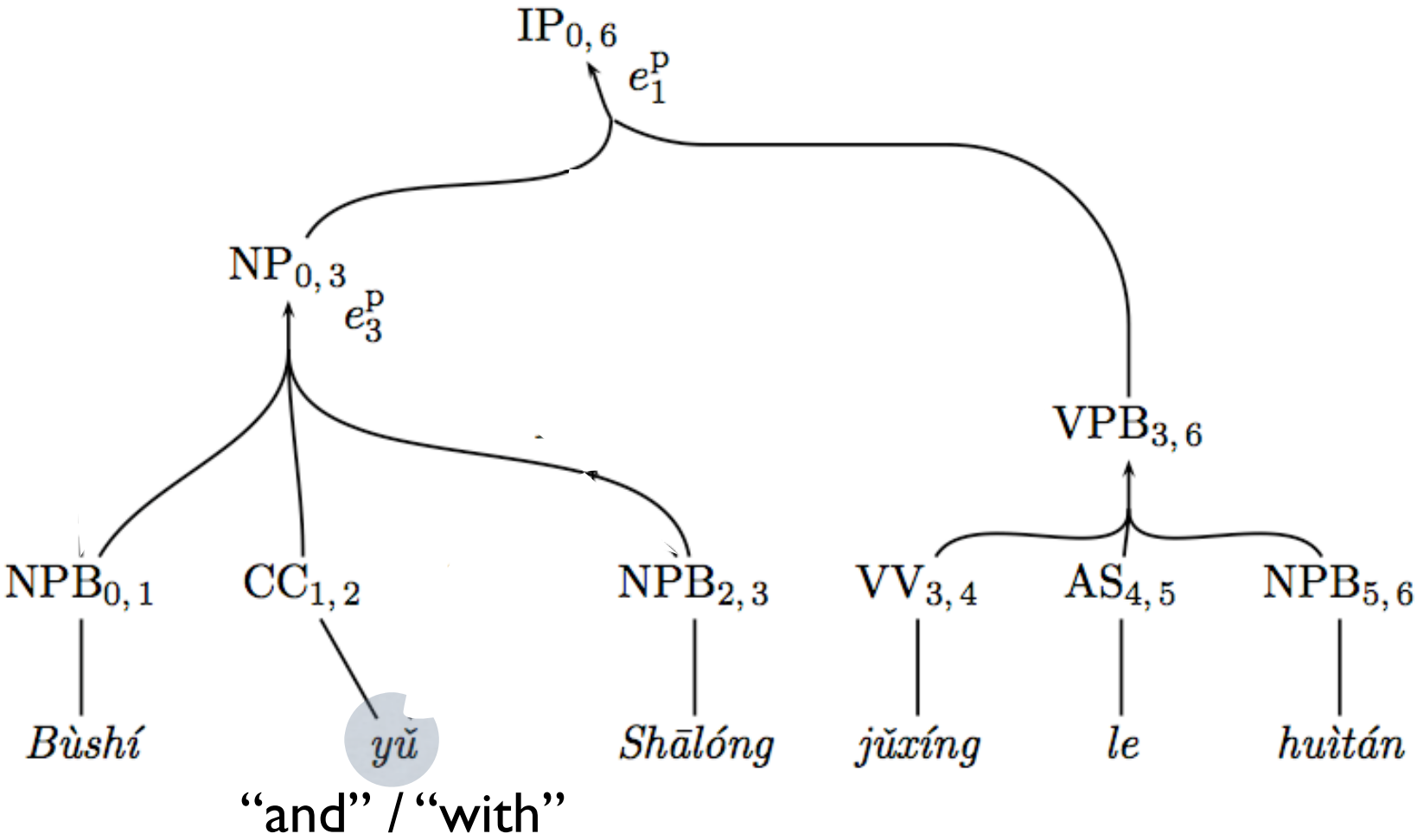
Chinese Forest

- parse the input into a forest instead of 1-best tree
- Chinese *yu* can be either a CC ("and") or P ("with")



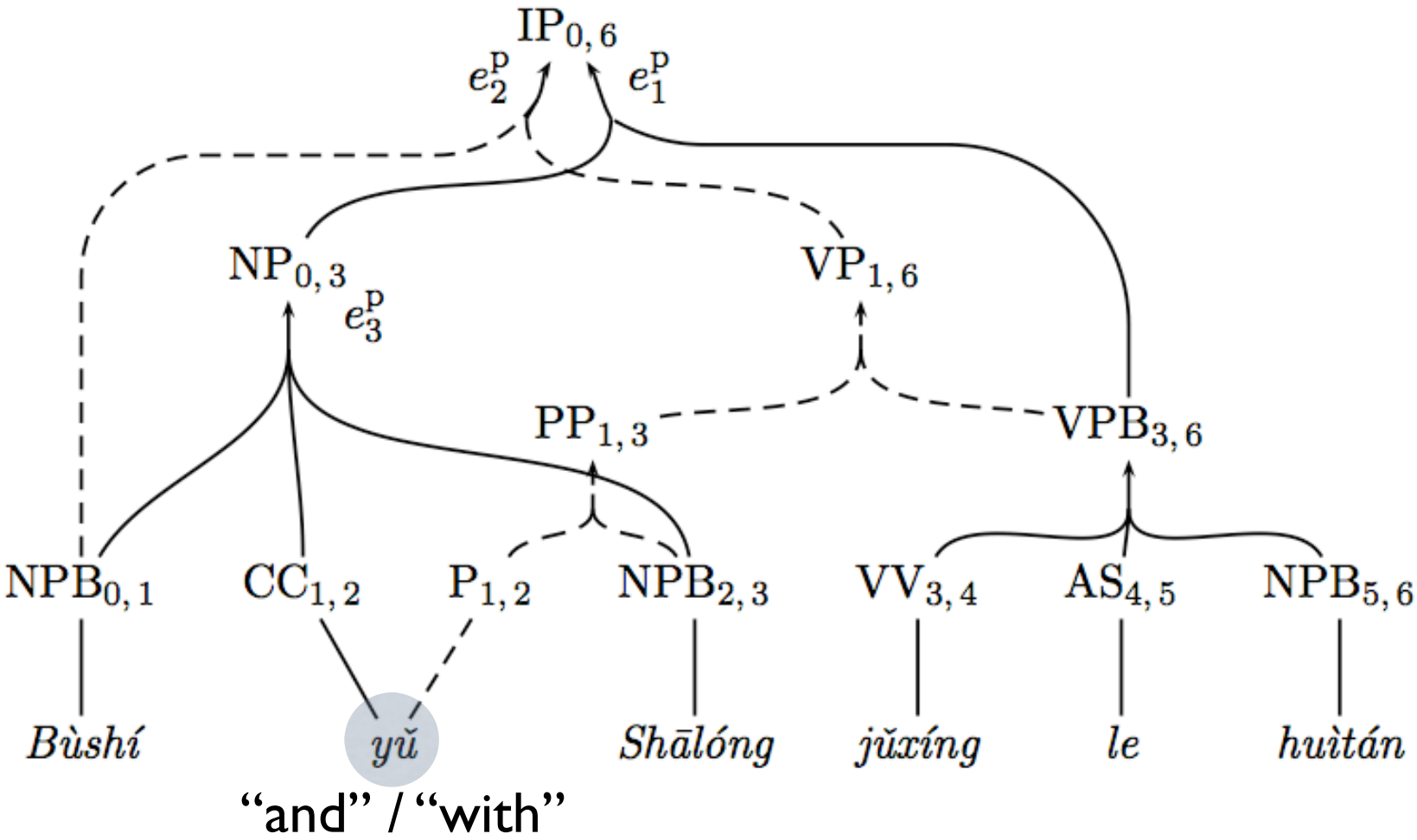
Chinese Forest

- parse the input into a forest instead of 1-best tree
- Chinese *yu* can be either a CC ("and") or P ("with")



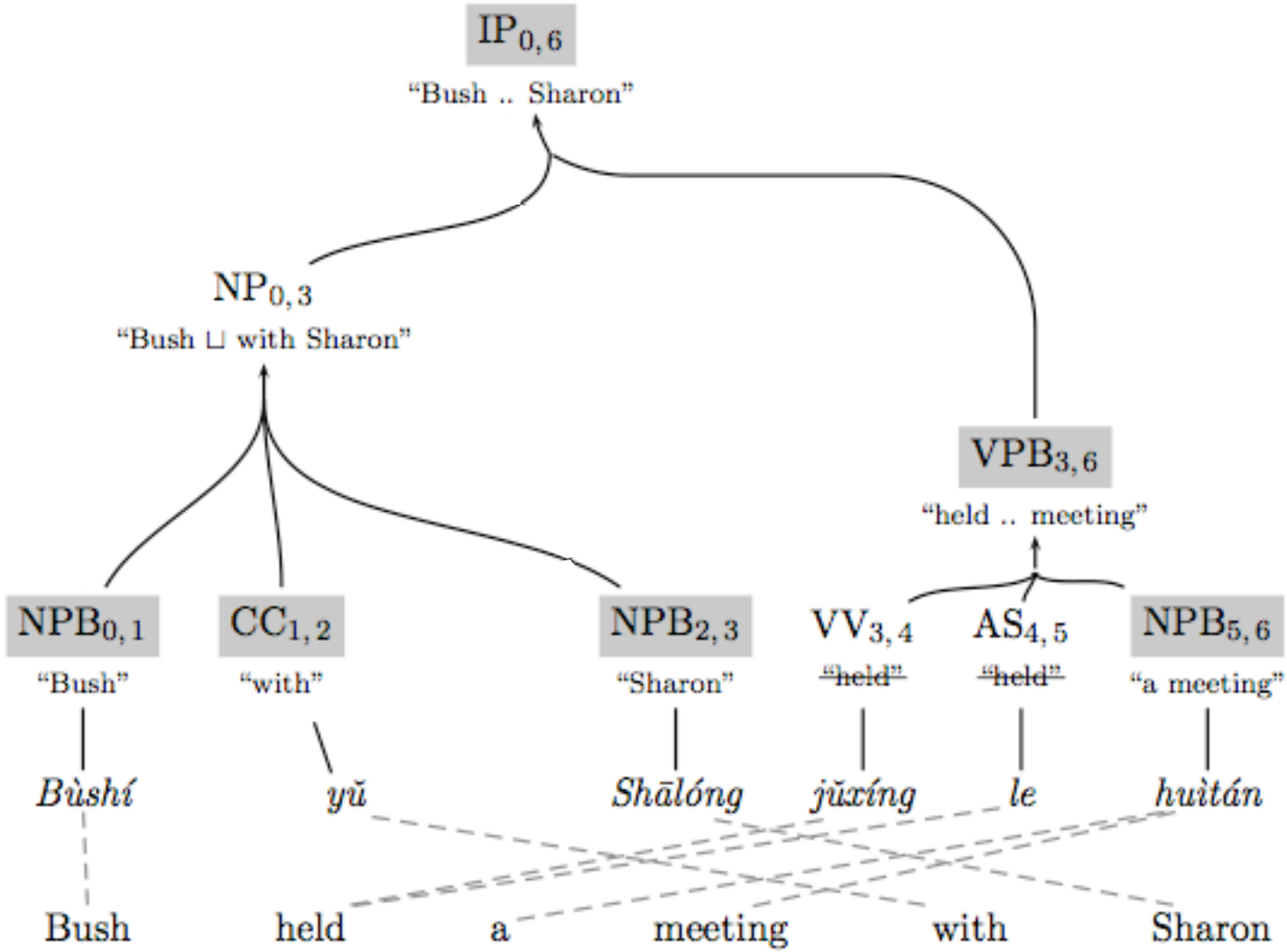
Chinese Forest

- parse the input into a forest instead of 1-best tree
- Chinese *yu* can be either a CC ("and") or P ("with")



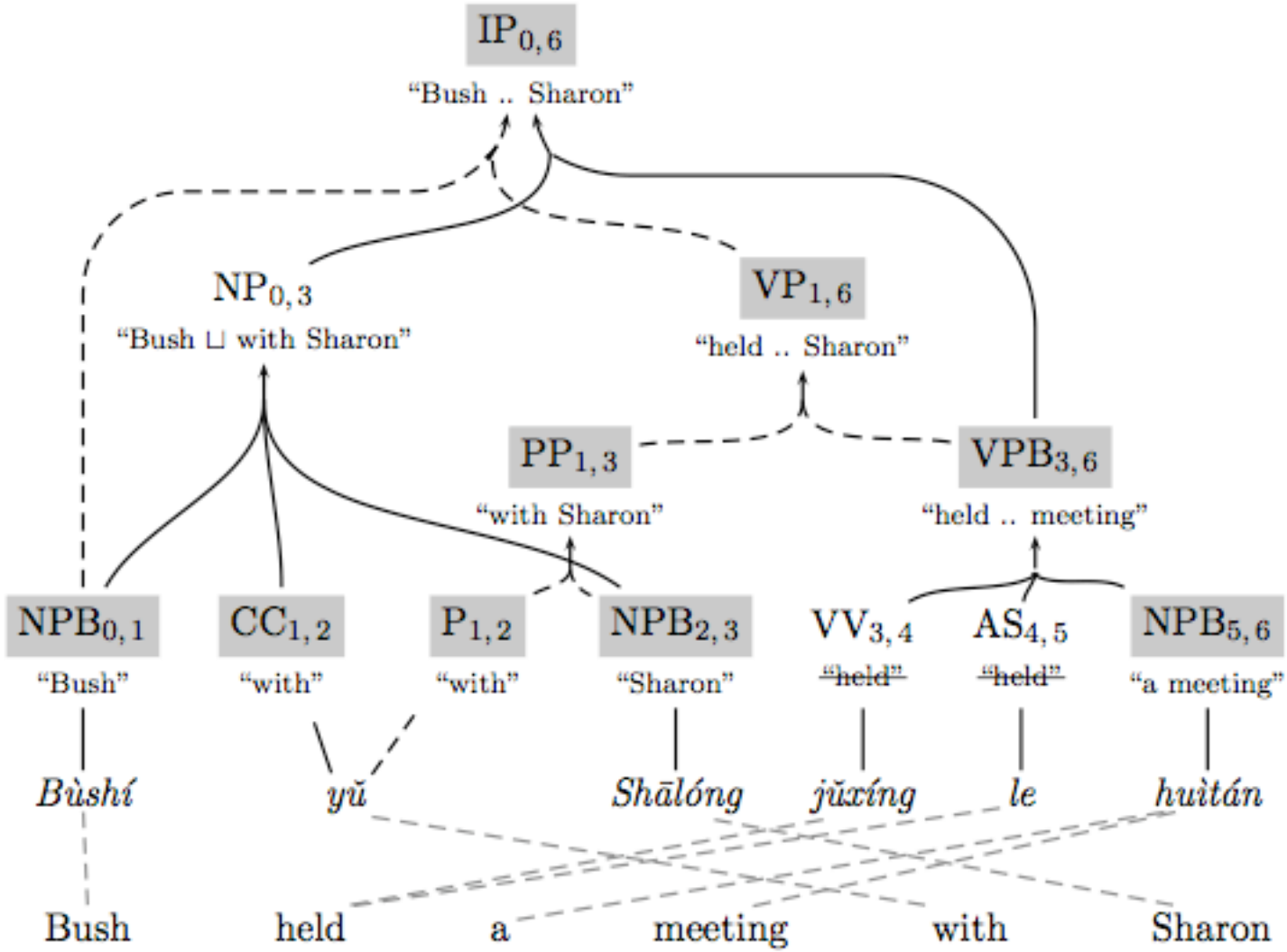
Forest-based Rule Extraction

- same at "where to cut"; different at "how to cut"



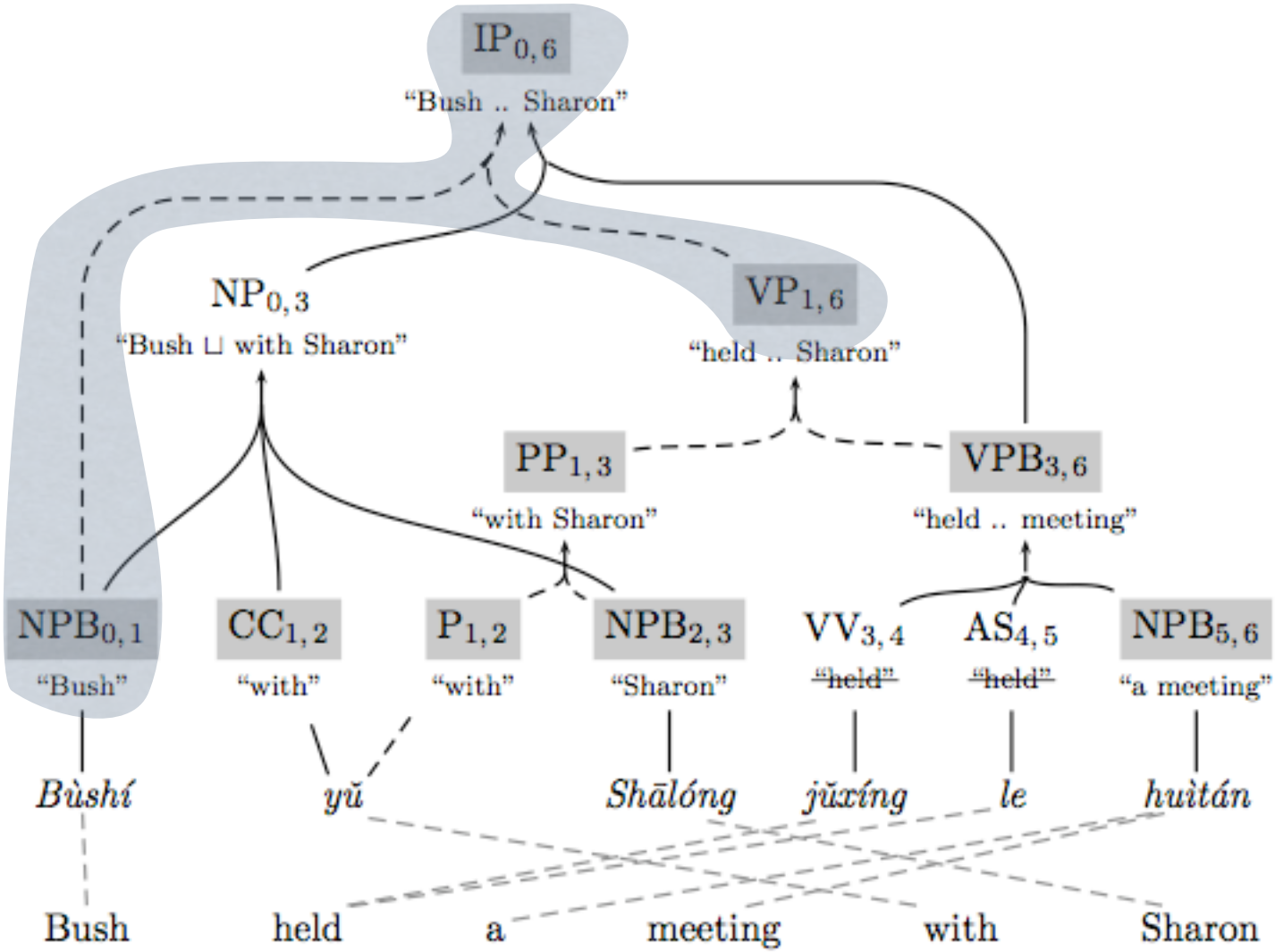
Forest-based Rule Extraction

- same at "where to cut"; different at "how to cut"



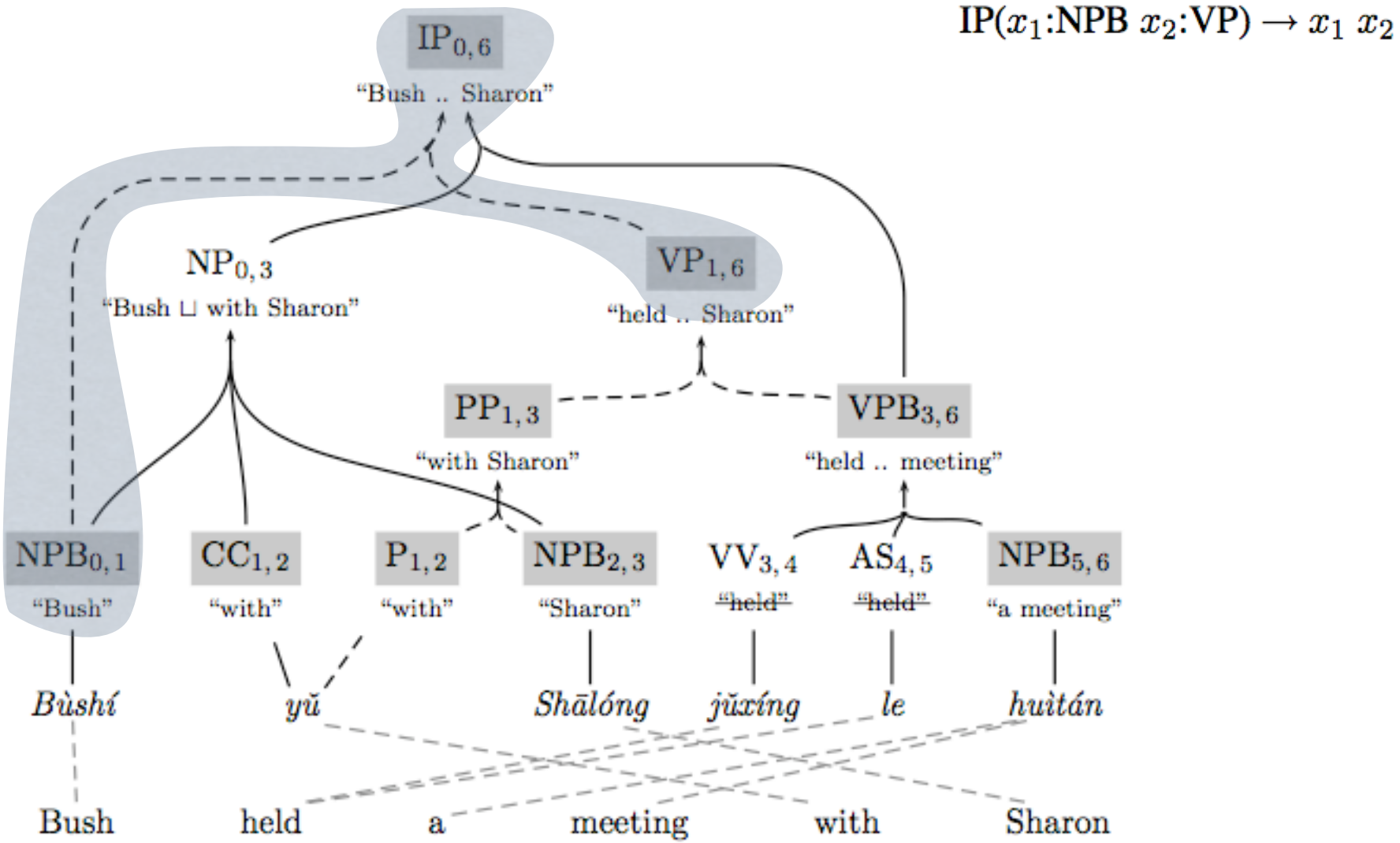
Forest-based Rule Extraction

- same at "where to cut"; different at "how to cut"



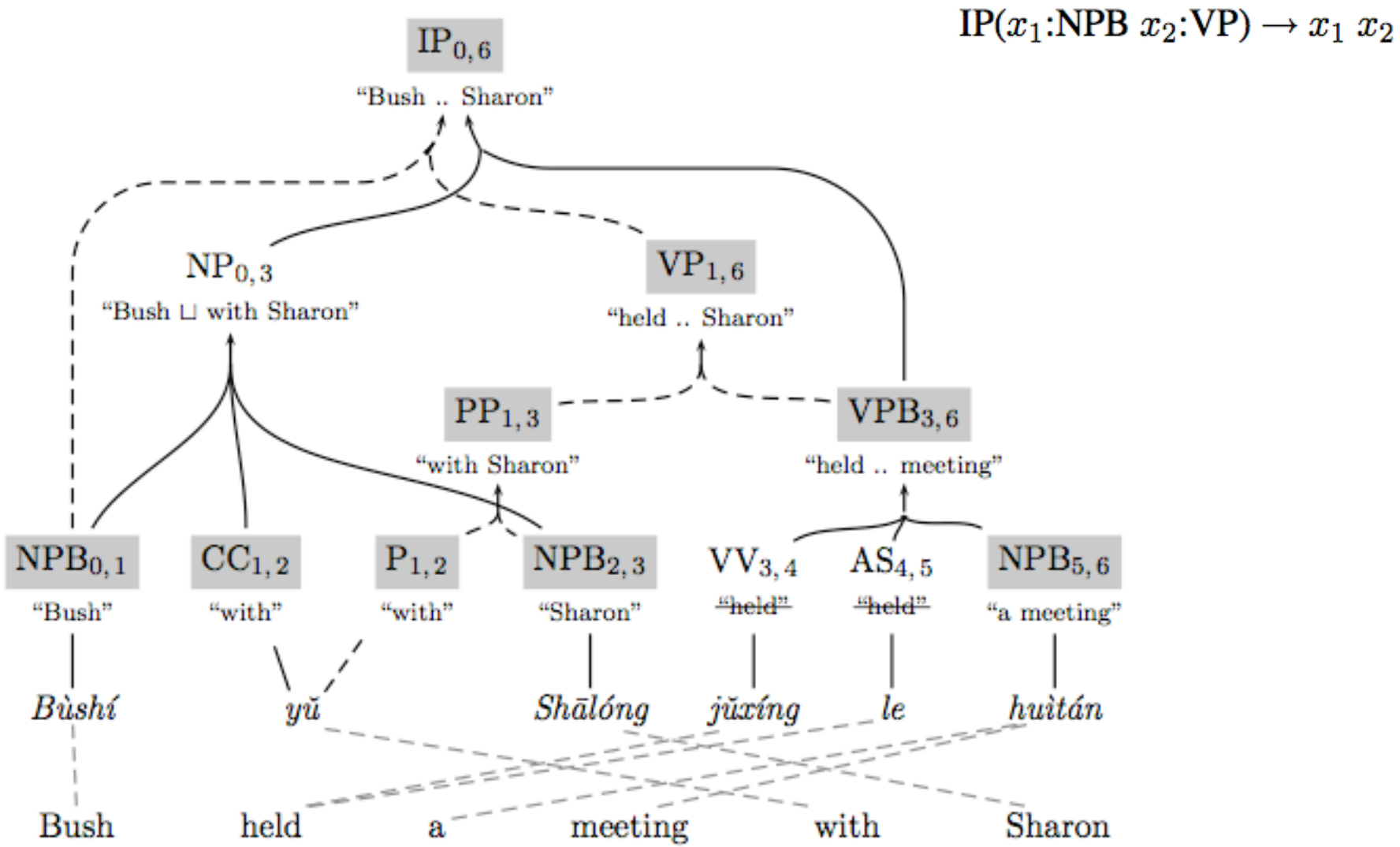
Forest-based Rule Extraction

- same at "where to cut"; different at "how to cut"



Forest-based Rule Extraction

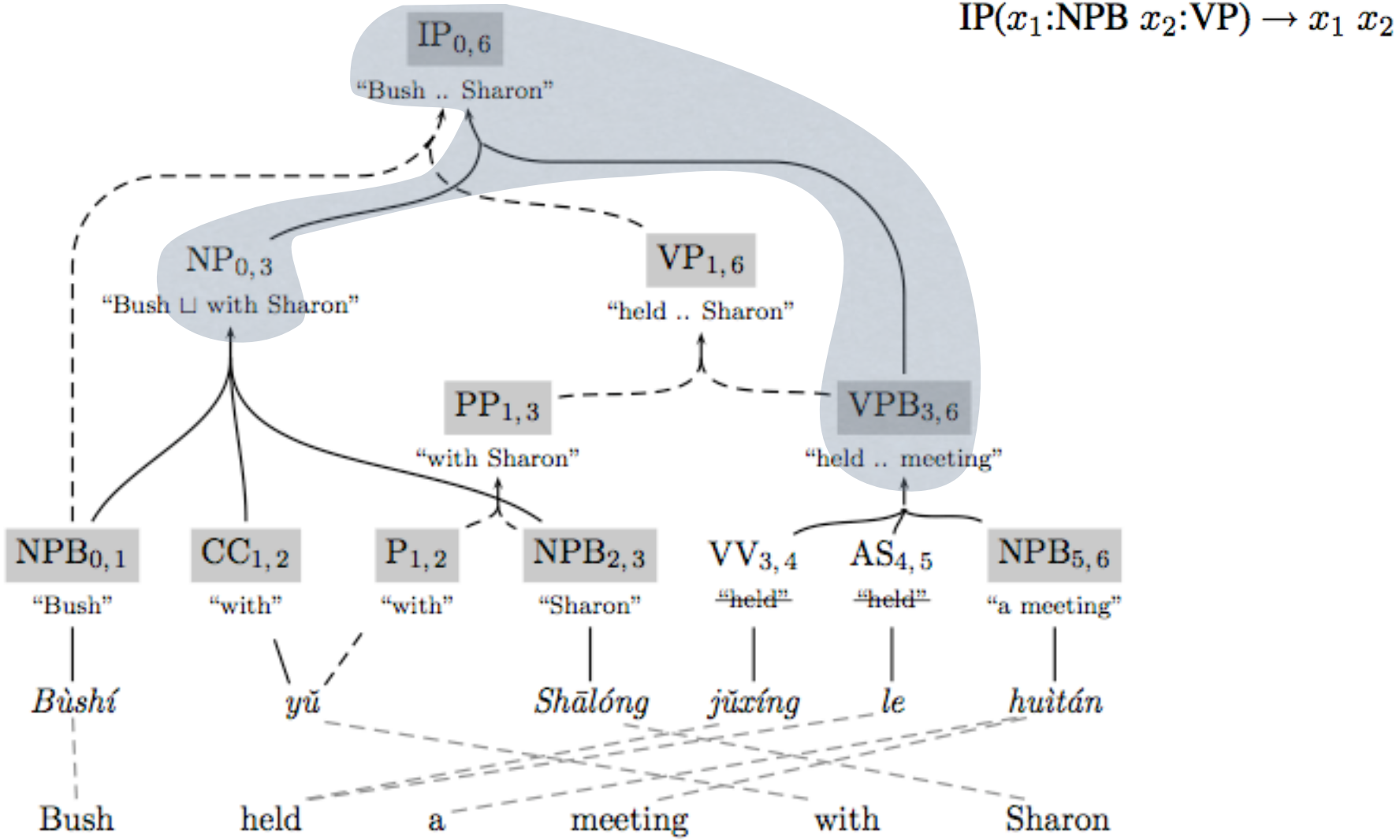
- same at "where to cut"; different at "how to cut"



 also in (Wang, Knight, Marcu, 2007); see related work

Forest-based Rule Extraction

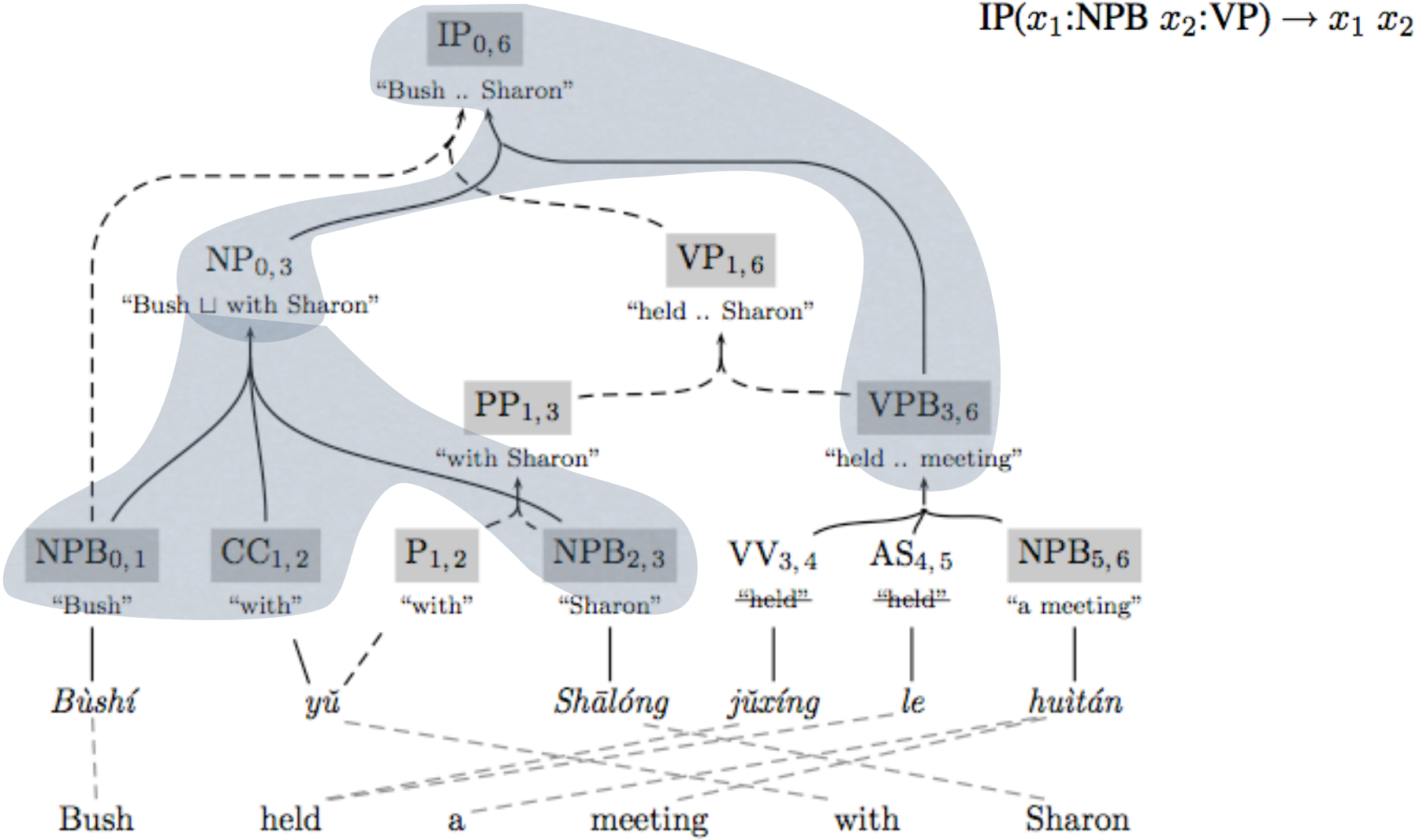
- same at "where to cut"; different at "how to cut"



 also in (Wang, Knight, Marcu, 2007); see related work

Forest-based Rule Extraction

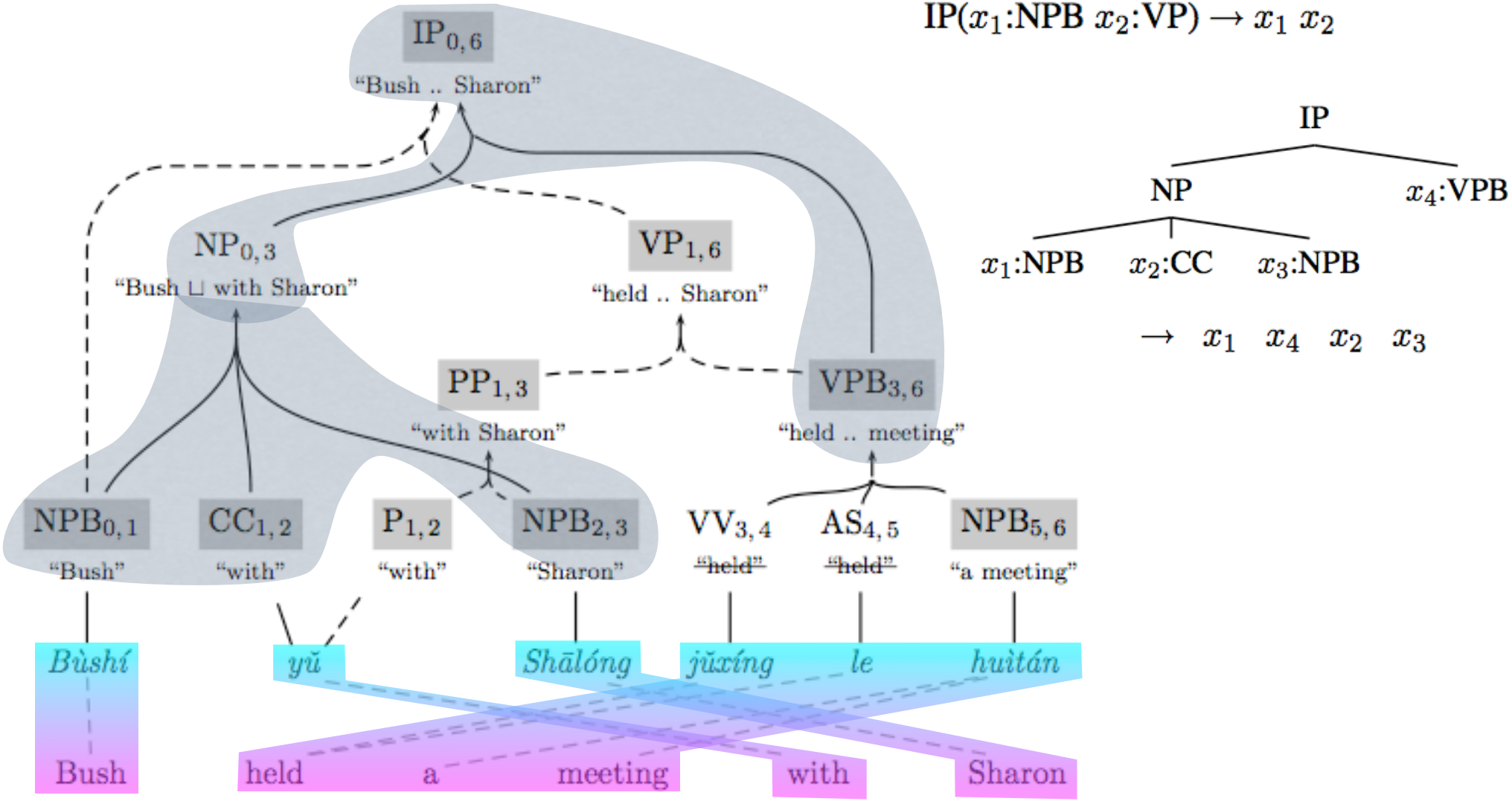
- same at "where to cut"; different at "how to cut"



 also in (Wang, Knight, Marcu, 2007); see related work

Forest-based Rule Extraction

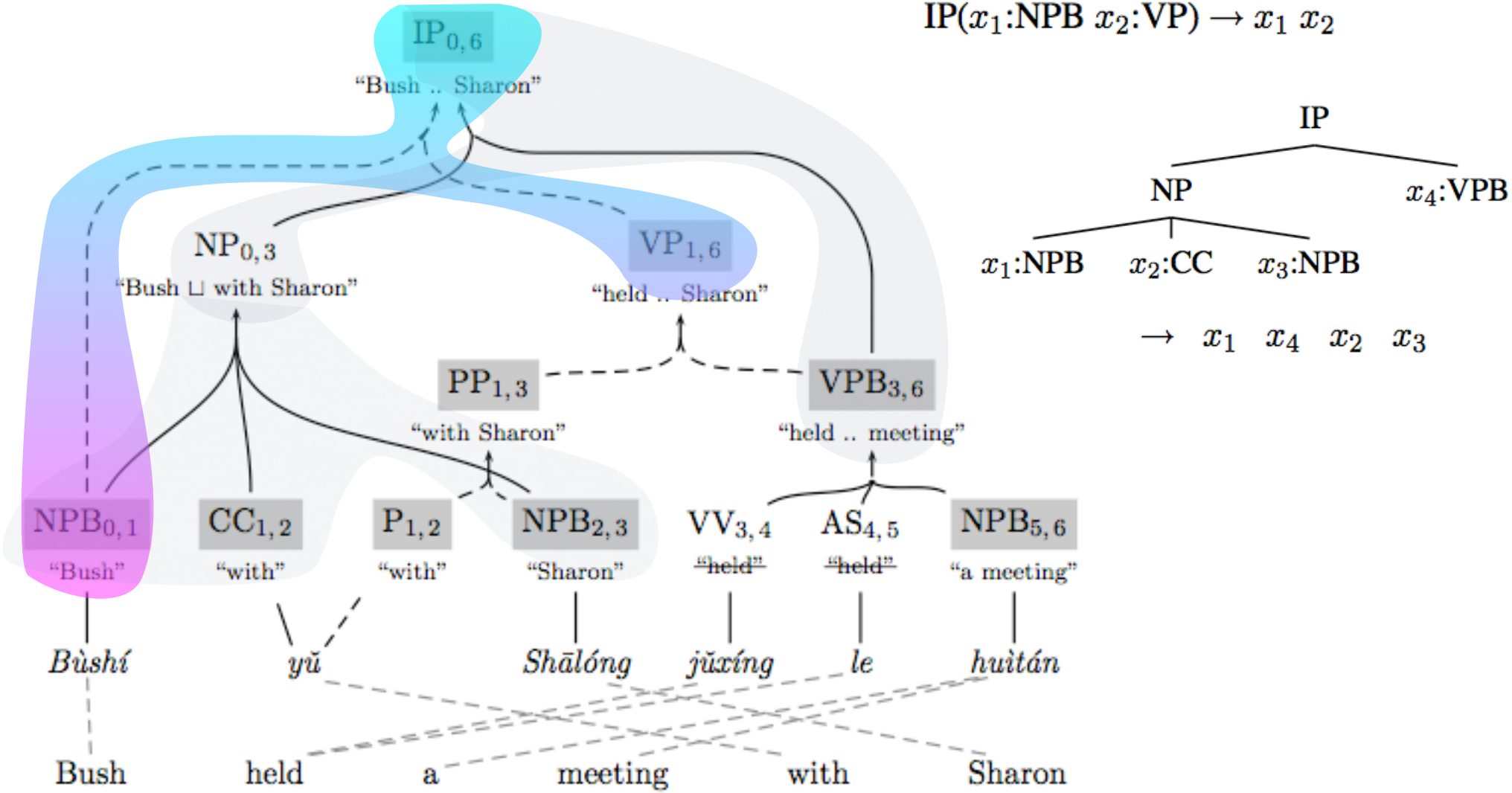
- same at "where to cut"; different at "how to cut"



 also in (Wang, Knight, Marcu, 2007); see related work

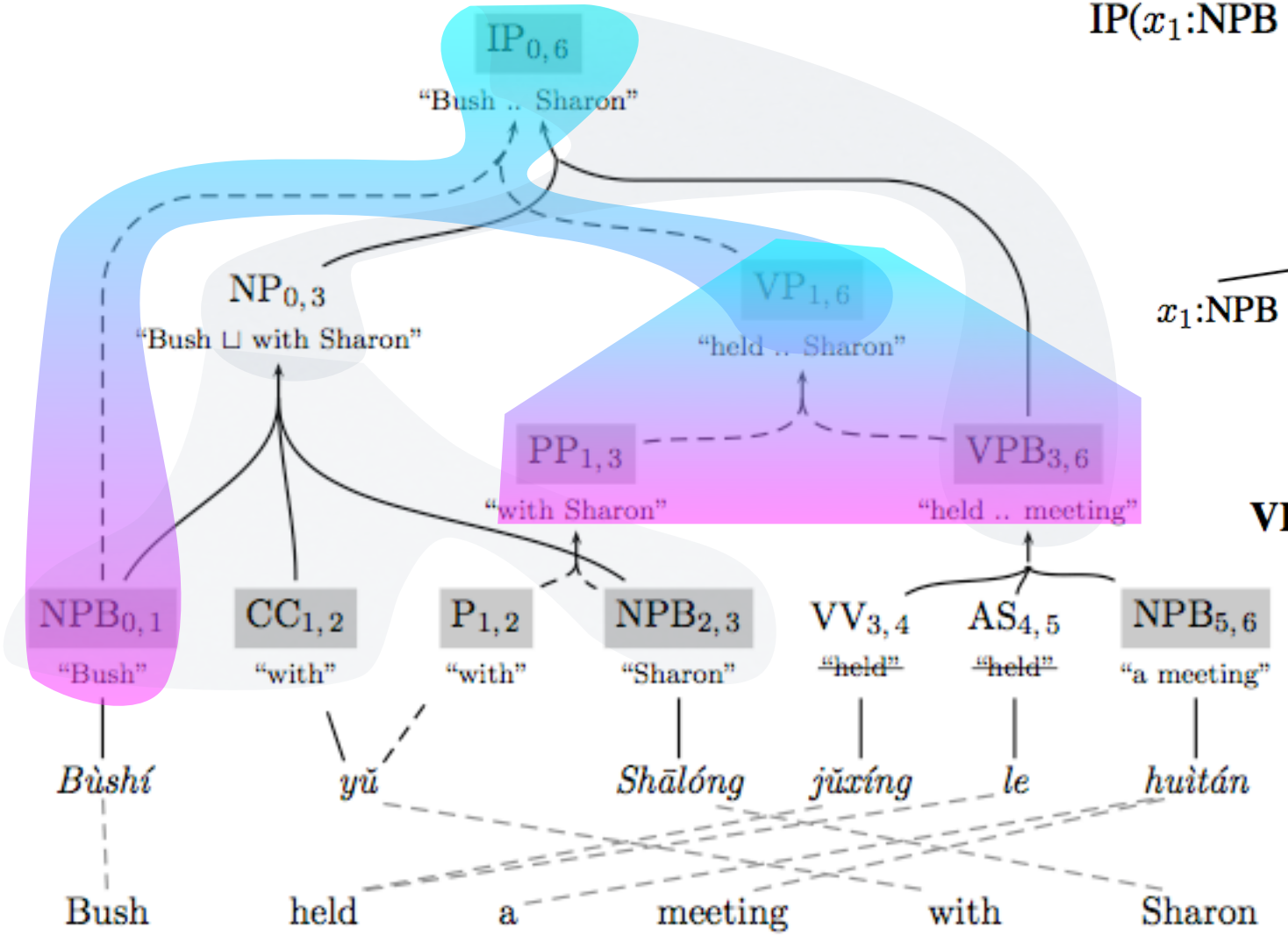
Forest-based Rule Extraction

- forest can extract smaller chunks of rules

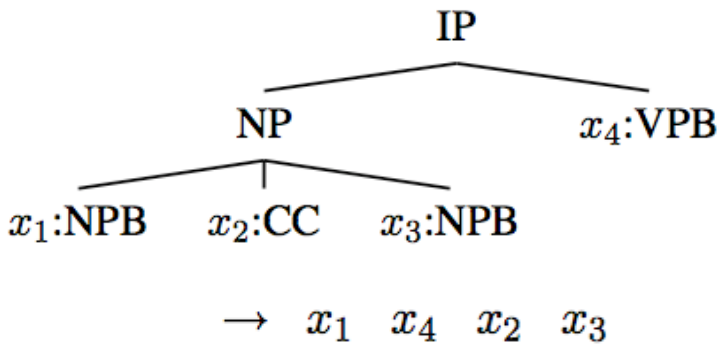


Forest-based Rule Extraction

- forest can extract smaller chunks of rules



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

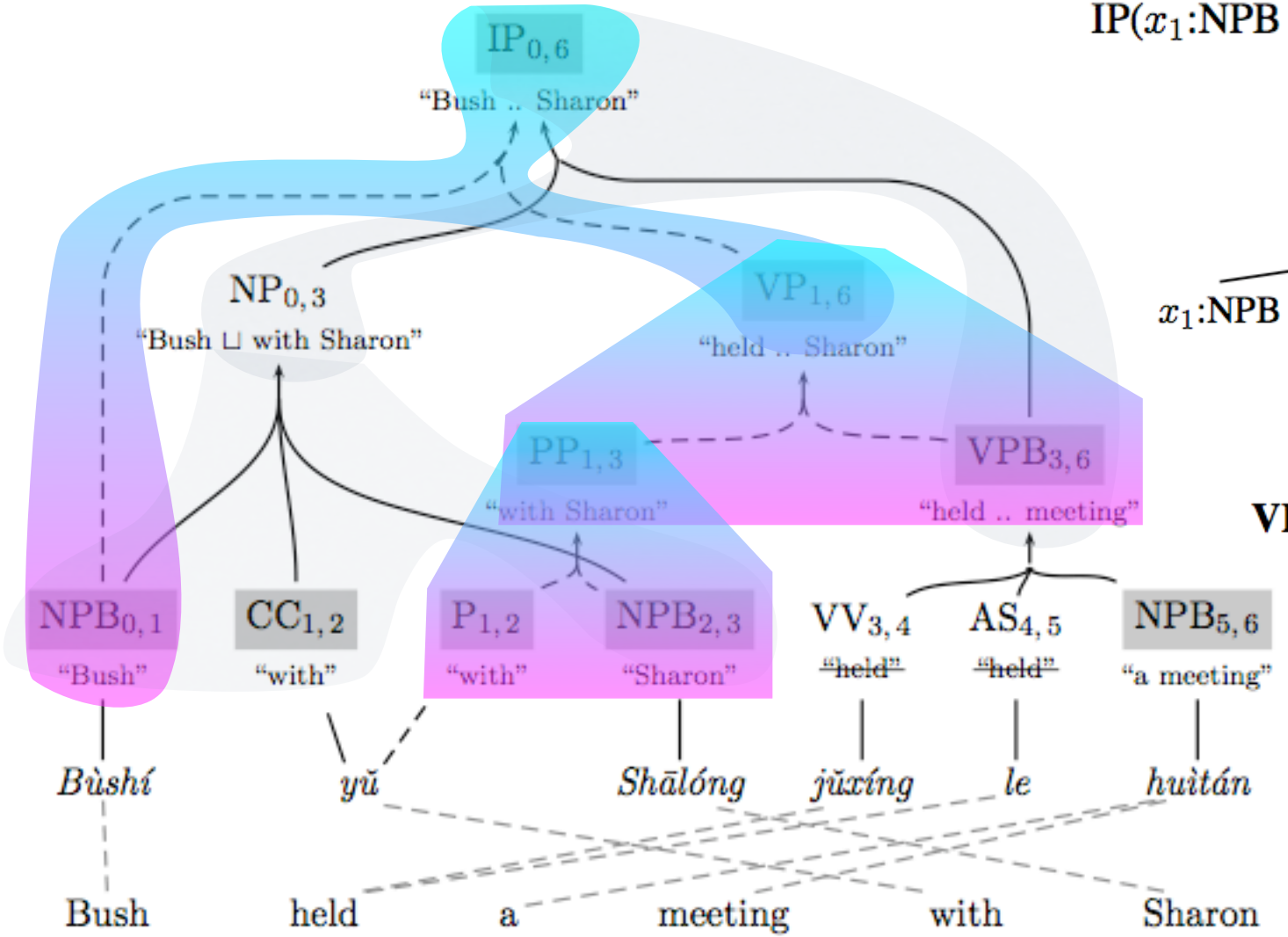


$\textbf{VP}\ (x_1\text{:PP}\ \ x_2\text{:VPB}) \rightarrow x_2\ x_1$

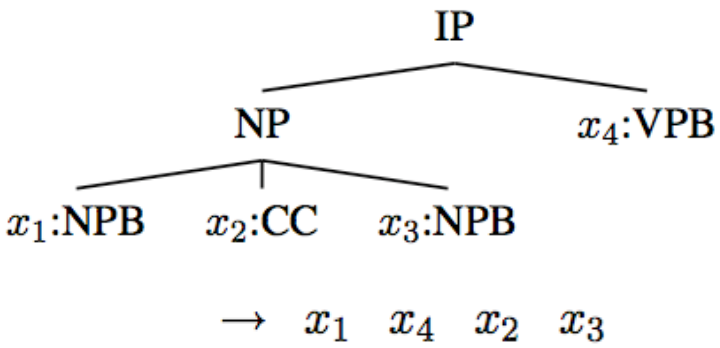
 also in (Wang, Knight, Marcu, 2007); see related work

Forest-based Rule Extraction

- forest can extract smaller chunks of rules



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

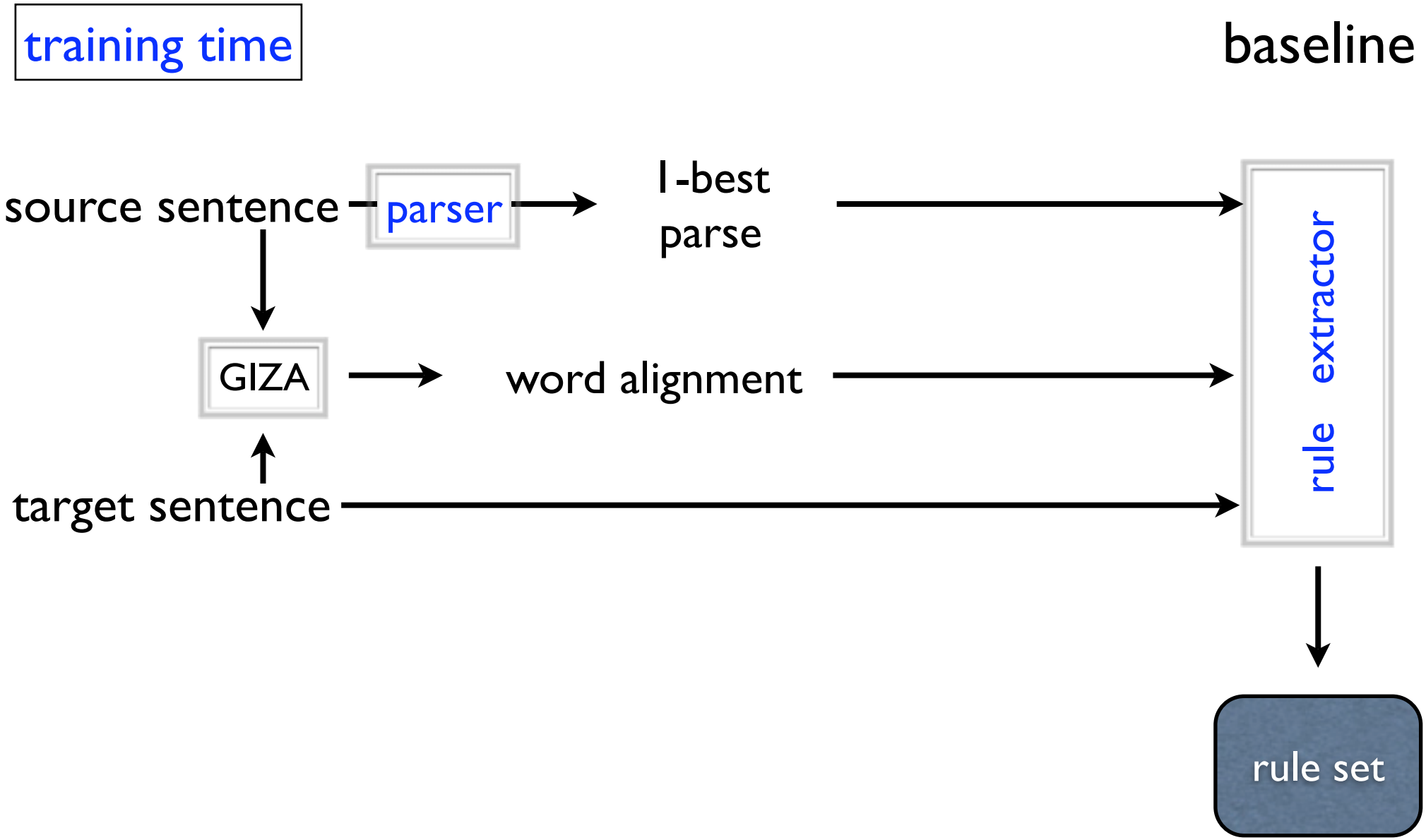


$\textbf{VP}\ (x_1\text{:PP}\ \ x_2\text{:VPB}) \rightarrow x_2\ x_1$

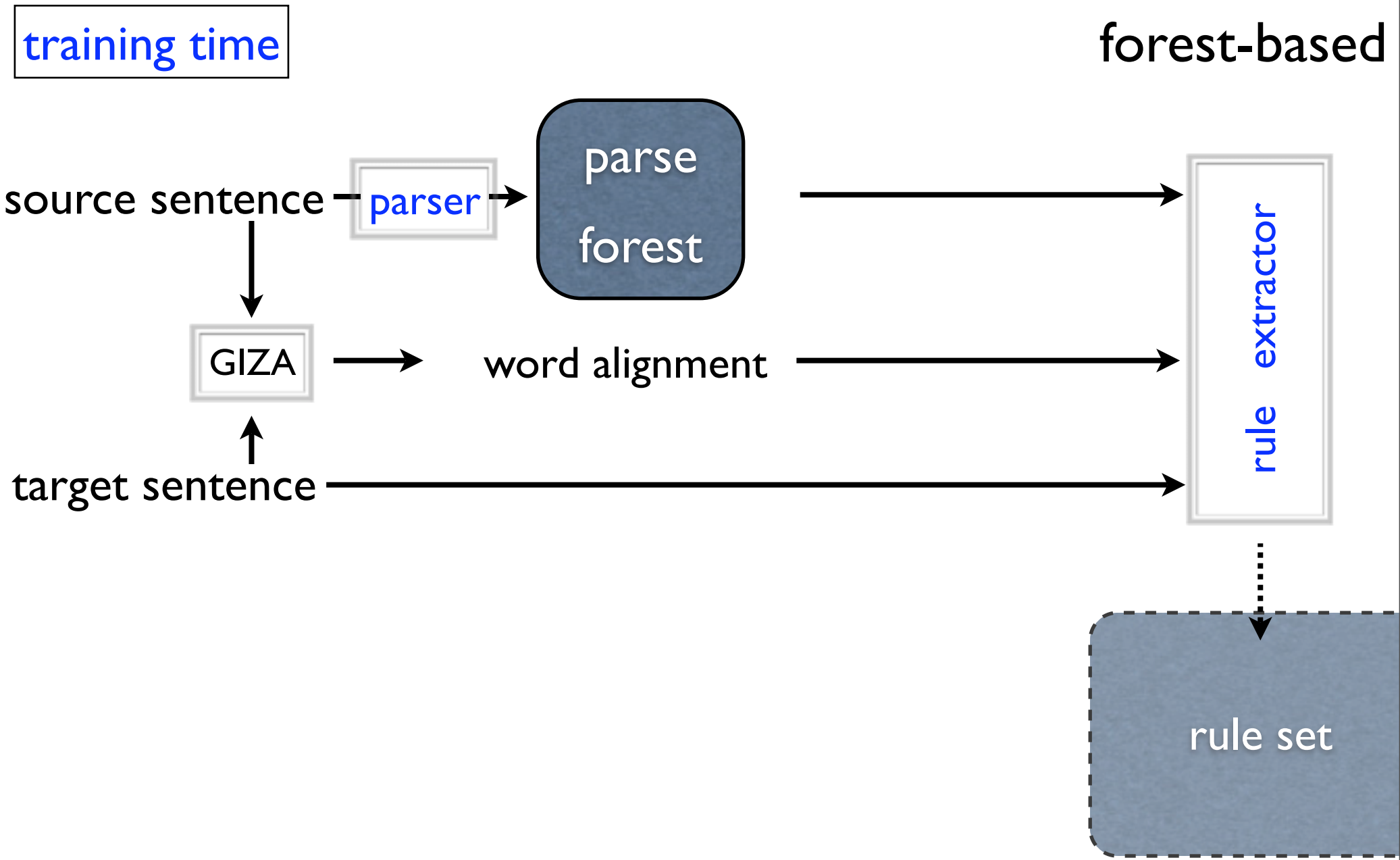
$\textbf{PP}\ (x_1\text{:P}\ \ x_2\text{:NPB}) \rightarrow x_1\ x_2$

 also in (Wang, Knight, Marcu, 2007); see related work

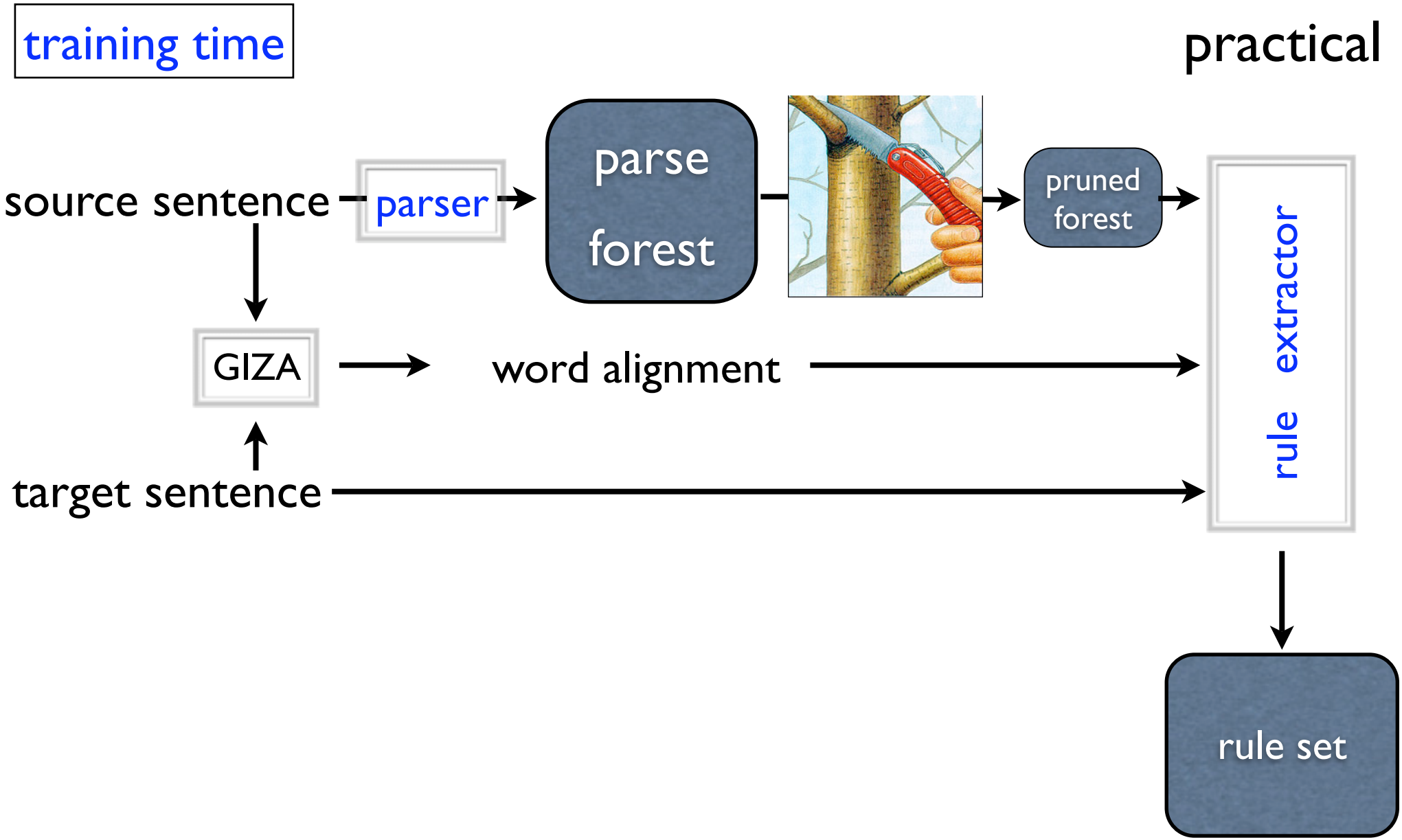
Rule Extraction Pipeline



Rule Extraction Pipeline



Rule Extraction Pipeline



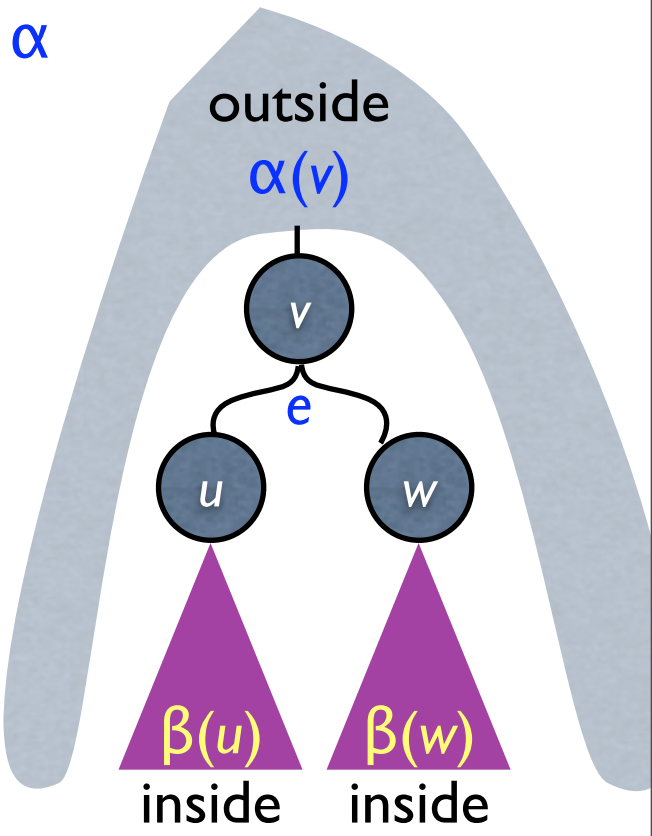
Inside-Outside Forest Pruning

- prune *unpromising* hyperedges
  - cost of best derivation that traverses e
- inside-outside, (max) marginal probs
  - first compute Viterbi inside β, outside α
- merit $\alpha\beta(e) = \alpha(v) \cdot p(e) \cdot \beta(u)\ \beta(w)$
  - similar to "expected count" in EM
- prune away a hyperedge e if

  $\alpha\beta(e) / \beta(\text{TOP}) > p$

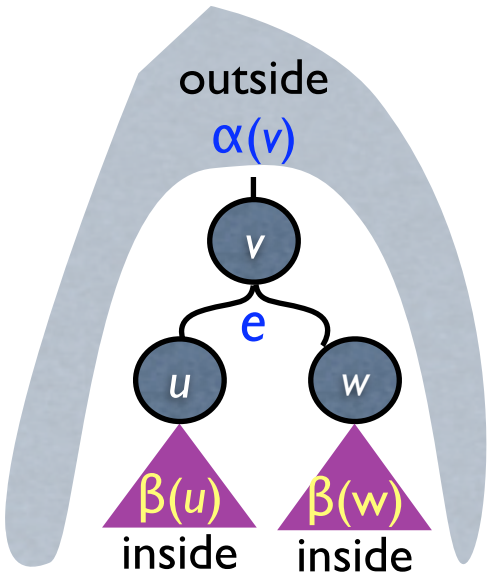
  for some threshold p

  (amount of deviation from 1-best derivation)



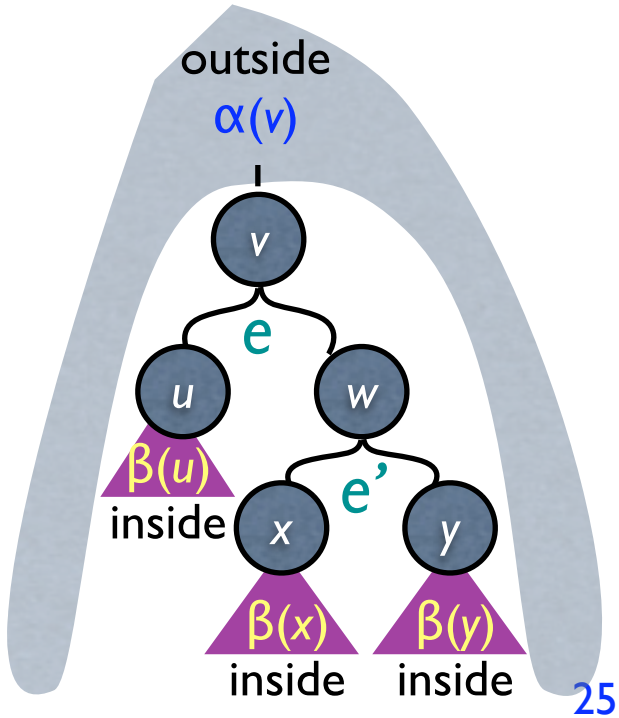
Fractional Rule Counts

- tree-based: every rule extracted gets a unit count
- forest-based: should penalize rules extracted from non 1-best parses
  - each rule gets a fractional count based on parse hyperedges
  - same idea as forest pruning: inside-outside merit

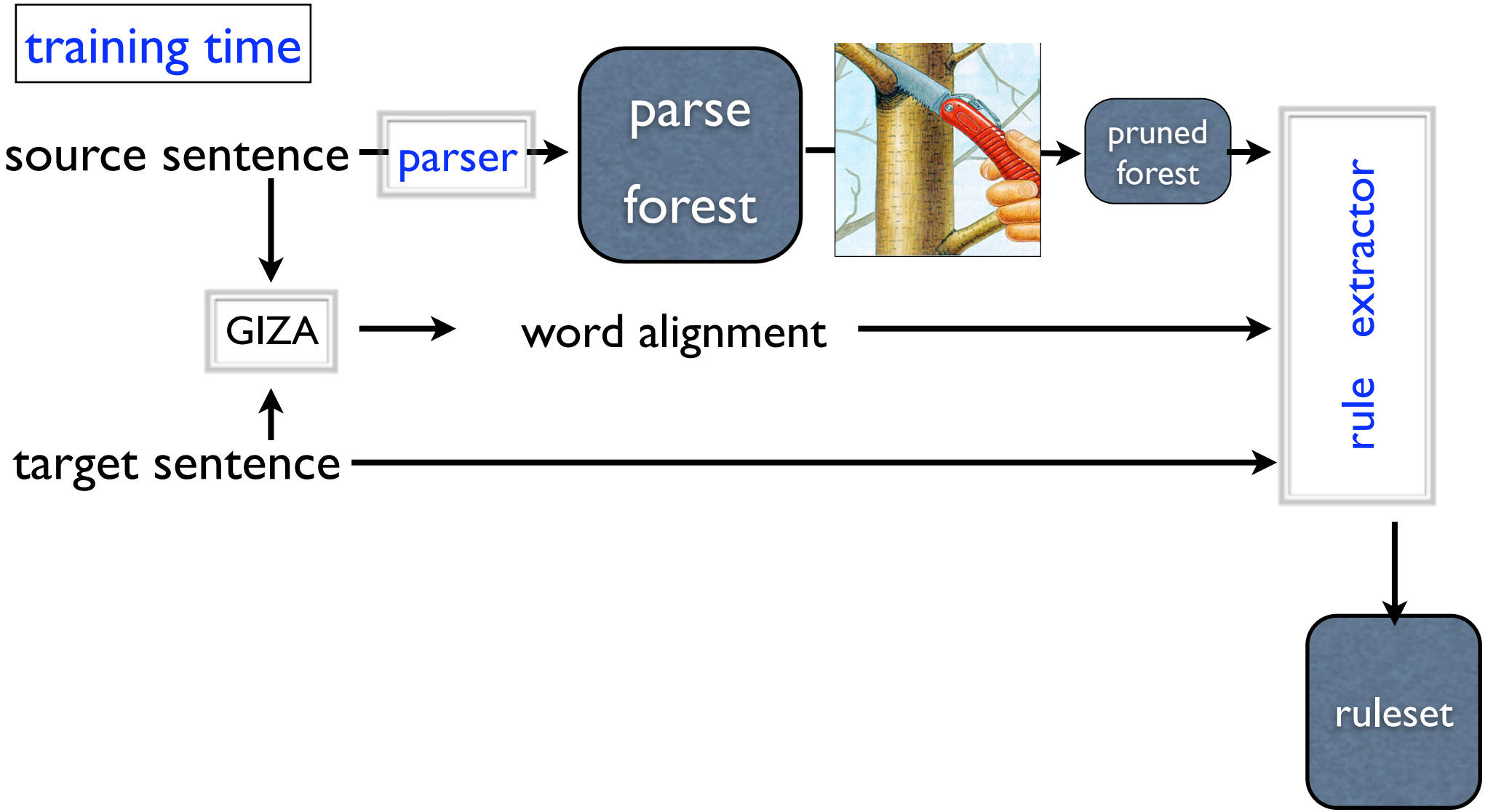


$$\alpha\beta(r) = \alpha\beta(\{e, e'\}) = \alpha(v) \cdot p(e)\, p(e') \cdot \beta(u)\beta(x)\beta(y)$$

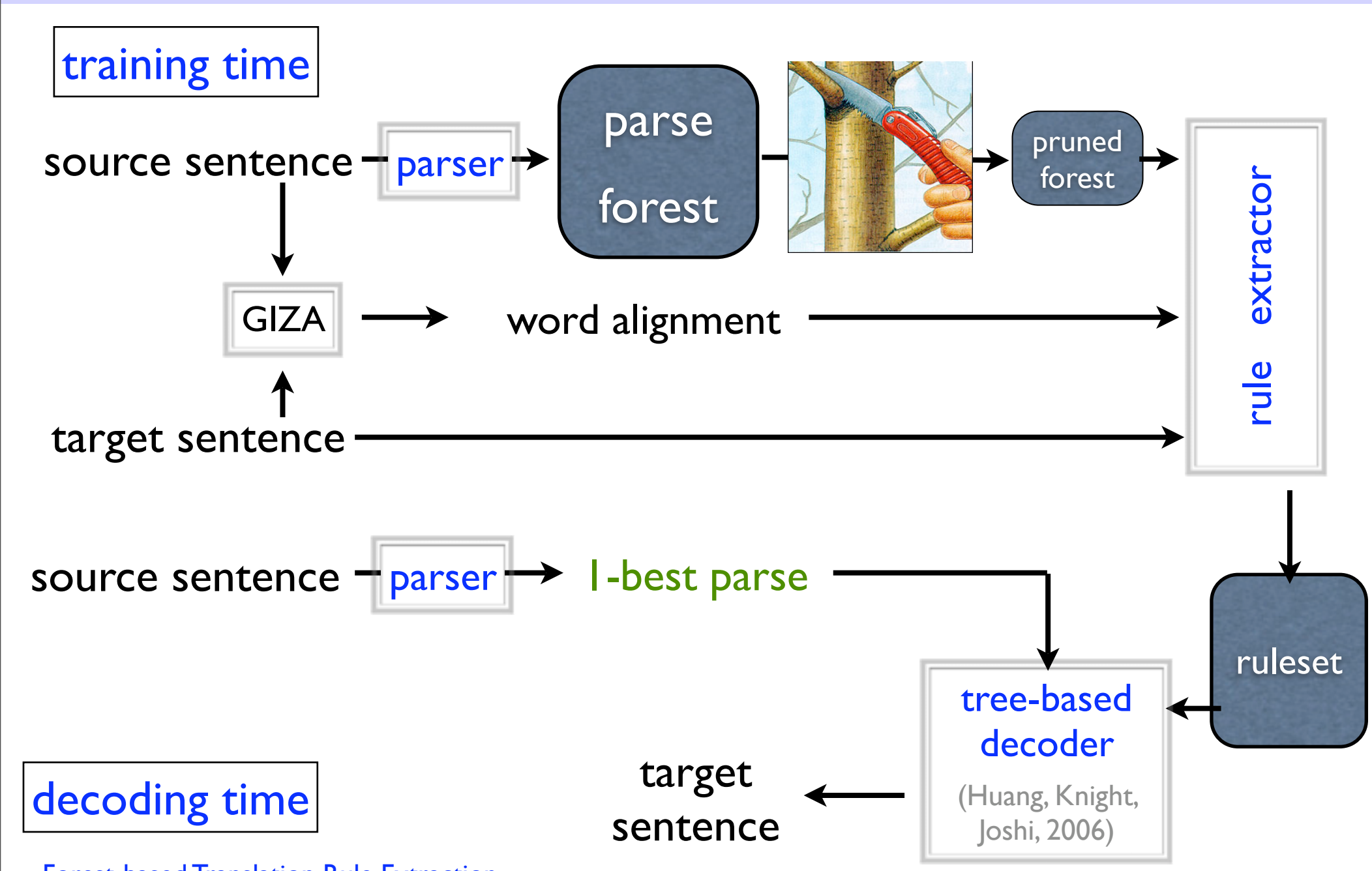
$$\text{count}(r) = \alpha\beta(r) \,/\, \beta(\text{TOP})$$



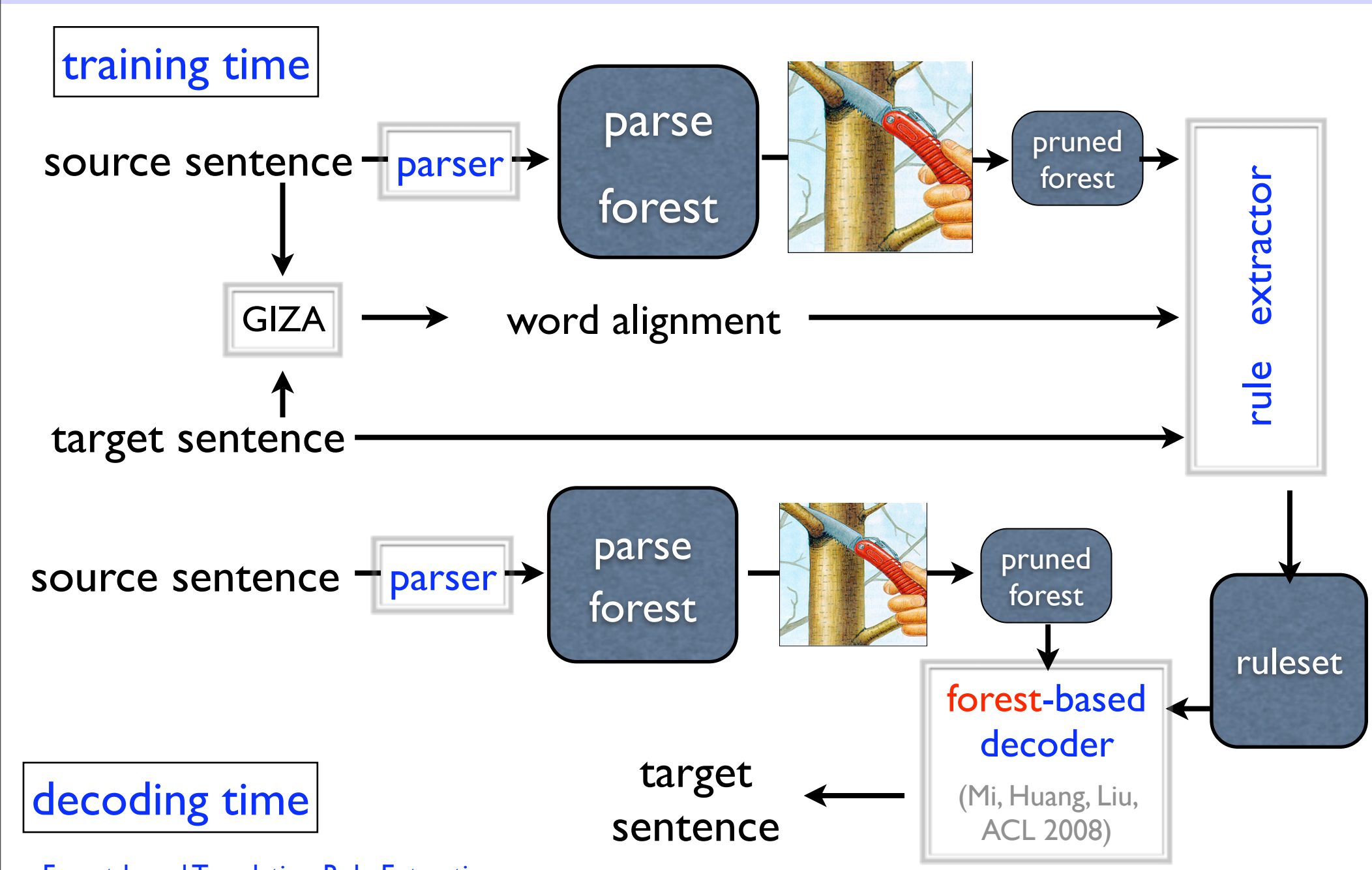
The Whole Forest Pipeline



The Whole Forest Pipeline



The Whole Forest^2 Pipeline



Forest-based Translation Rule Extraction

Related Work

- forest in rule extraction
  - Wang, Knight, Marcu (2007) pack different binarizations of a single parse tree into a binarization forest
  - we use a real parse forest of many different parses
  - then use EM to guess best binarization for each parse; real rules only extracted from one single binarized tree
- multiple parses for rule extraction
  - Venugopal et al (2008) use *k*-best trees; negative results
  - we will show small improvement from *k*-best trees, but big improvement from forests

Experiments

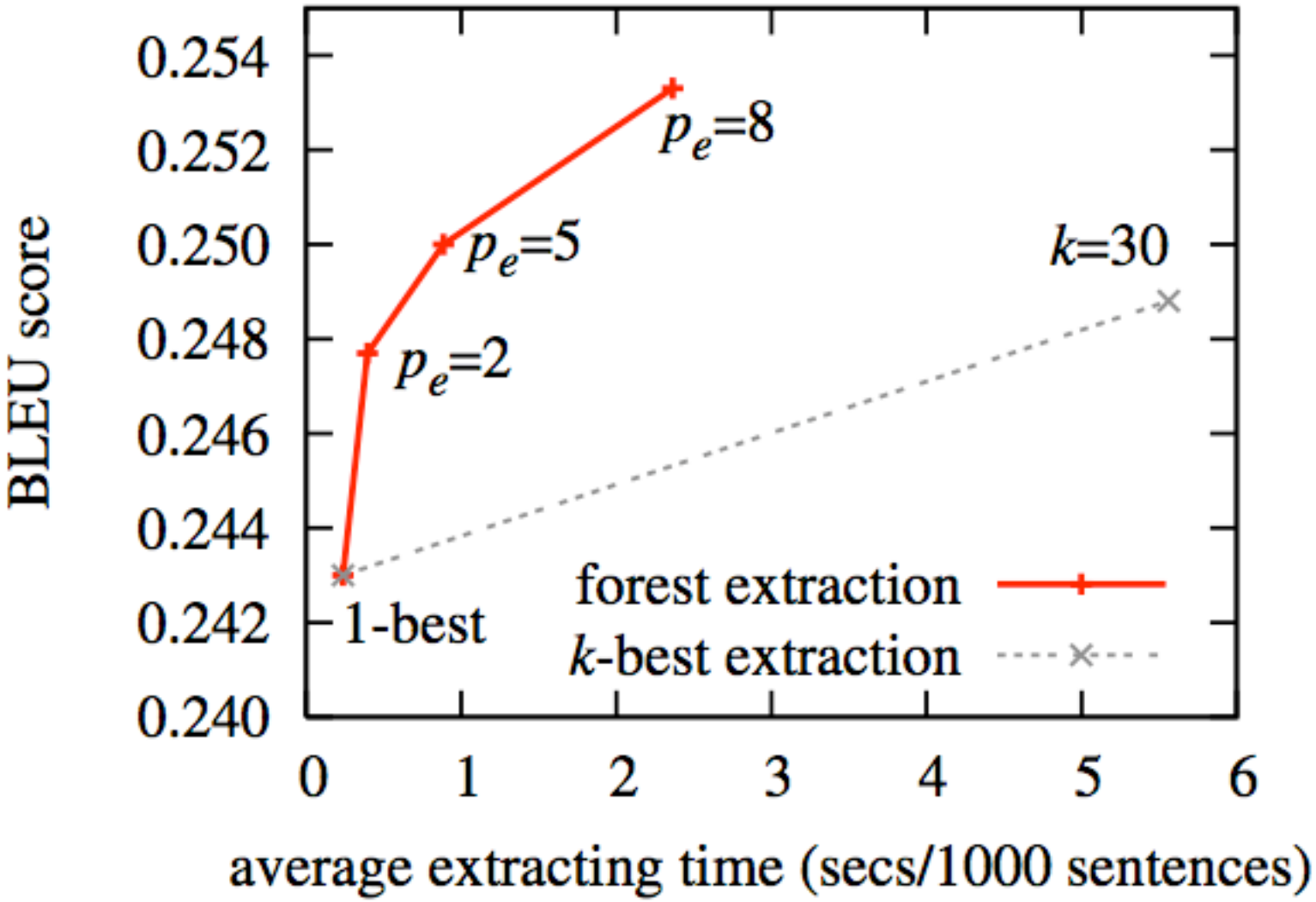
both small-scale and large scale

Small-Scale Experiments

- Chinese-to-English translation
  - on a tree-to-string system similar to (Liu et al, 2006)
- 31k sentences pairs (0.8M Chinese & 0.9M English words)
- GIZA++ aligned
- trigram language model trained on the English side
- dev: NIST 2002 (878 sent.); test: NIST 2005 (1082 sent.)
- Chinese-side parsed by the parser of Xiong et al. (2005)
  - modified to output a forest for each sentence (Huang 2008)
- 1-best^2 baseline: 0.2430; Pharaoh: 0.2297

Forest vs. *k*-best Extraction

1.0 Bleu improvement over 1-best,
twice as fast as 30-best extraction



Large-Scale Experiments

- FBIS: 239k sentence pairs (7M/9M Chinese/English words)
- forest in both extraction and decoding
- forest^2 results is 2.5 points better than 1-best^2
  - and outperforms Hiero (by quite a bit)

decoding on ...

rules from ...

	1-best tree	forest
1-best tree	0.2560	0.2674
30-best trees	0.2634	0.2767
forest	0.2679	0.2816
Hiero	0.2738	

Translation Examples

- src 鲍威尔 说 与 阿拉法特 会谈 很 重要
  Baoweir shuo yu Alafate huitan hen zhongyao
  Powell say with Arafat talk very important
- ref Powell Said Talks with Arafat Very Important (headline)

- 1-best^2 Powell said the very important talks with Arafat
- forest^2 Powell said his meeting with Arafat is very important
- hiero Powell said very important talks with Arafat

Conclusion

- forest provides flexibility to extract better rules
  - contains exponentially more trees than *k*-best parses
  - efficient extraction thanks to structure sharing
- applicable to all *linguistically syntax-based* systems
  - tree-to-string, string-to-tree, tree-to-tree, tree-seq-to-str, ...
- very simple idea, but works very well in practice
  - ~1 Bleu points better than 1-best extraction
  - ~2.5 Bleu better when combined with forest decoding
    - outperforms the state-of-the art Hiero (Chiang, 2005)

Forest is your friend in machine translation.

Thank you!

you may need to prune,
but please save the forest.

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

Thanks to Qun Liu, Kevin Knight, Aravind Joshi,
Wei Wang, and the anonymous reviewers.

Penn
University of Pennsylvania