Trees and Forests in Machine Translation

Liang Huang
City University of New York

Joint work with Kevin Knight (ISI), Aravind Joshi (Penn), Haitao Mi and Qun Liu (ICT), 2006--2010

University of Pennsylvania, March 31st, 2015

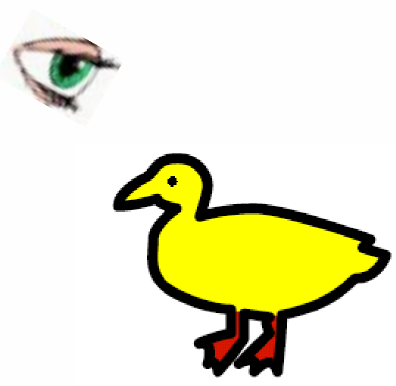
NLP is Hard

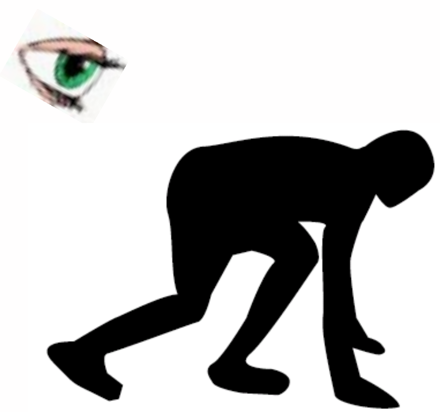
- *how many interpretations?*

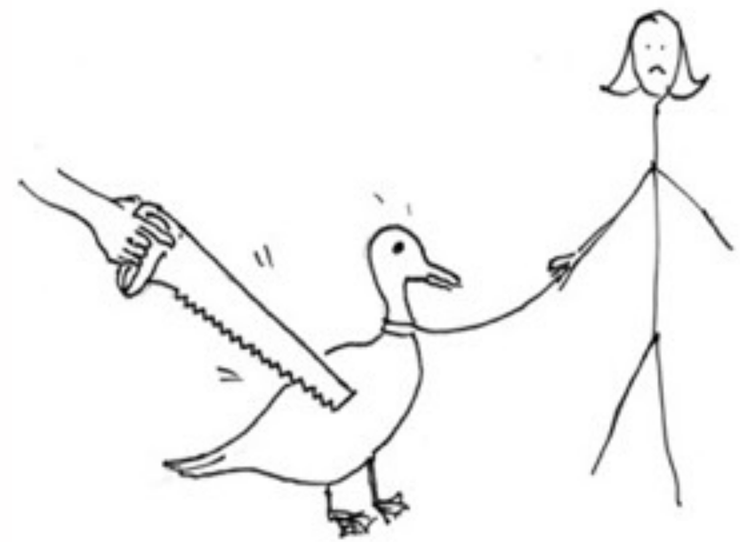
Aravind Joshi

I saw her duck

lexical ambiguity







NLP is Hard

- *how many interpretations?*

Aravind Joshi

I eat sushi with tuna

structural ambiguity

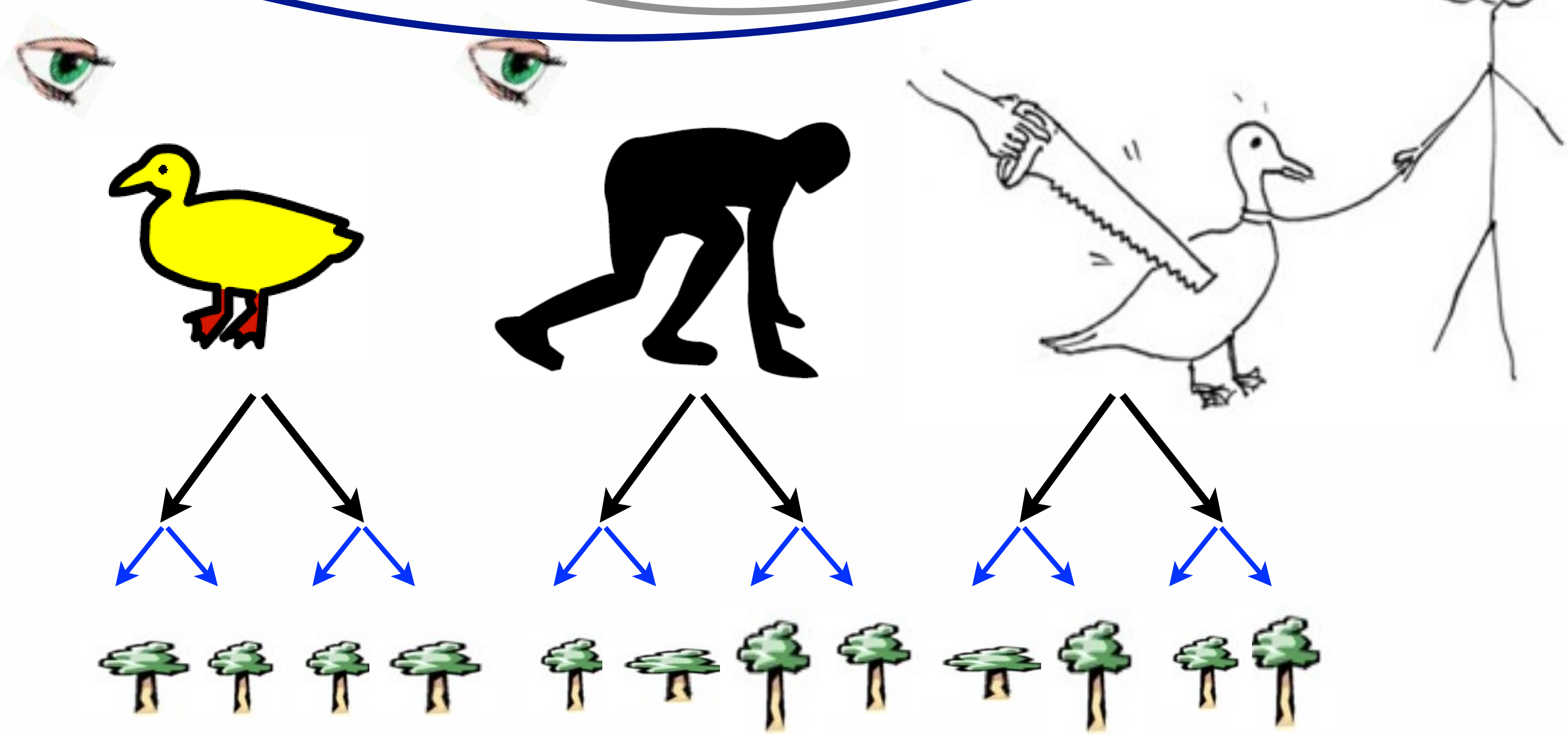
Ambiguity Explosion

- *how many interpretations?*

NLP: ambiguity explosion

TCS: combinatorial explosion

I saw her duck with a telescope in the garden ...



...

Unexpected Structural Ambiguity

Ambiguity in Translation

zi zhu zhong duan
自 助 终 端

self help terminal device

help oneself terminating machine

translation requires understanding!

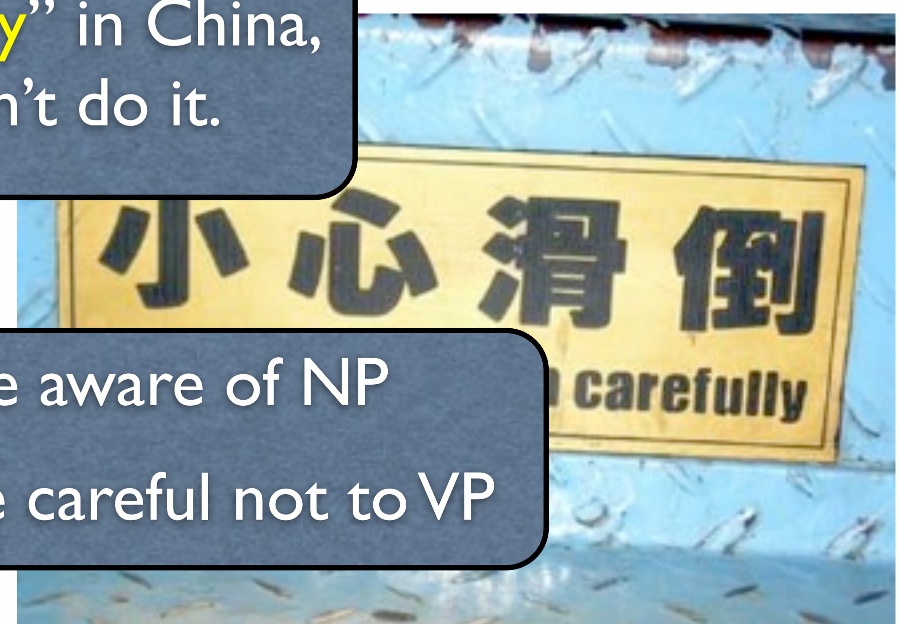
(ATM, “self-service terminal”)

Ambiguity in Translation

"domain adaptation" problem in machine learning

Ambiguity in Translation

liang's rule: if you see "X carefully" in China, just don't do it.



小心 NP <=> be aware of NP

小心 VP <=> be careful not to VP

Translate Server Error

clear evidence that MT is used in real life.

How do people translate?

1. understand the source language sentence
2. generate the target language translation

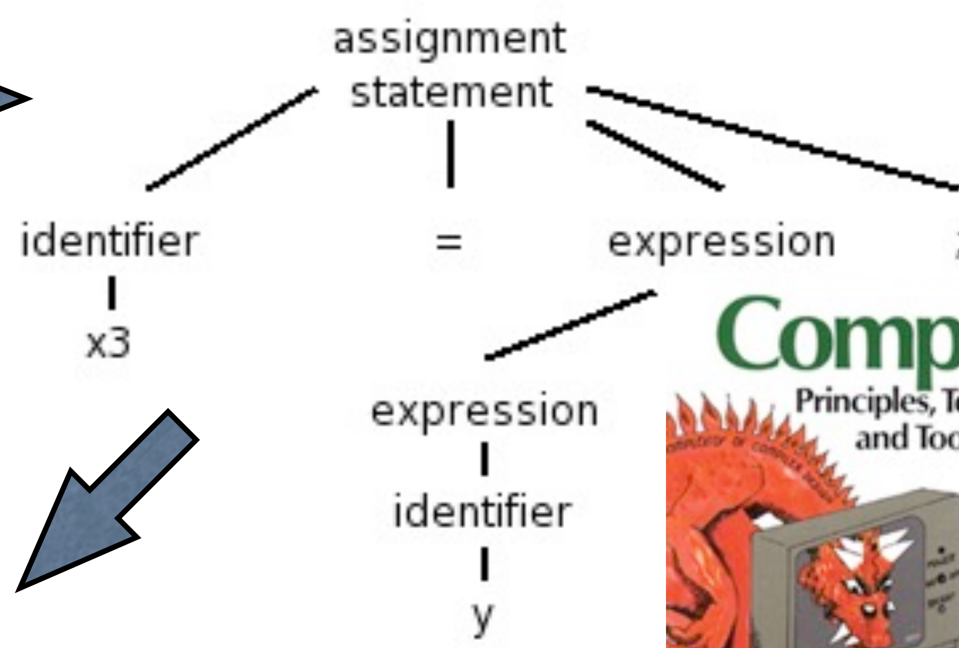
布什	与	沙龙	举行	了	会谈
Bùshí	*yu*	*Shalóng*	*juxíng*	*le*	*huìtán*
Bush	and/ with	Sharon	hold	[*past.*]	meeting

"Bush held a meeting with Sharon"

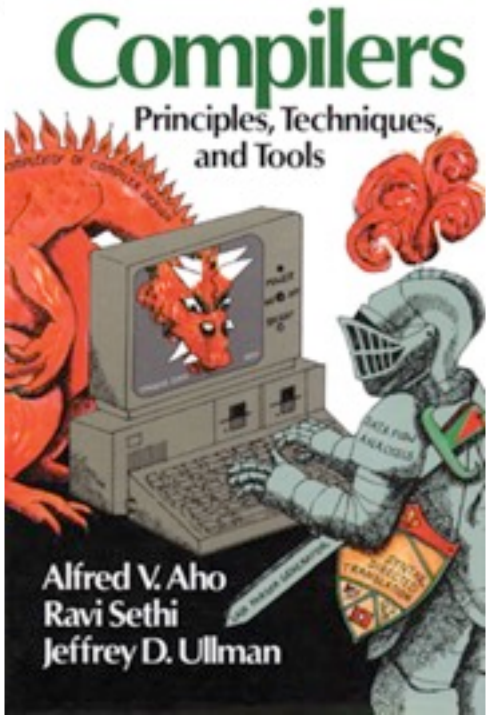
How do compilers translate?

1. parse high-level language program into a syntax tree
2. generate intermediate or machine code accordingly

```
x3 = y + 3;
```




```
LD      R1,  id2
ADDF    R1,  R1, #3.0    // add float
RTOI    R2,  R1          // real to int
ST      id1, R2
```



syntax-directed translation (~1960)

Syntax-Directed Machine Translation

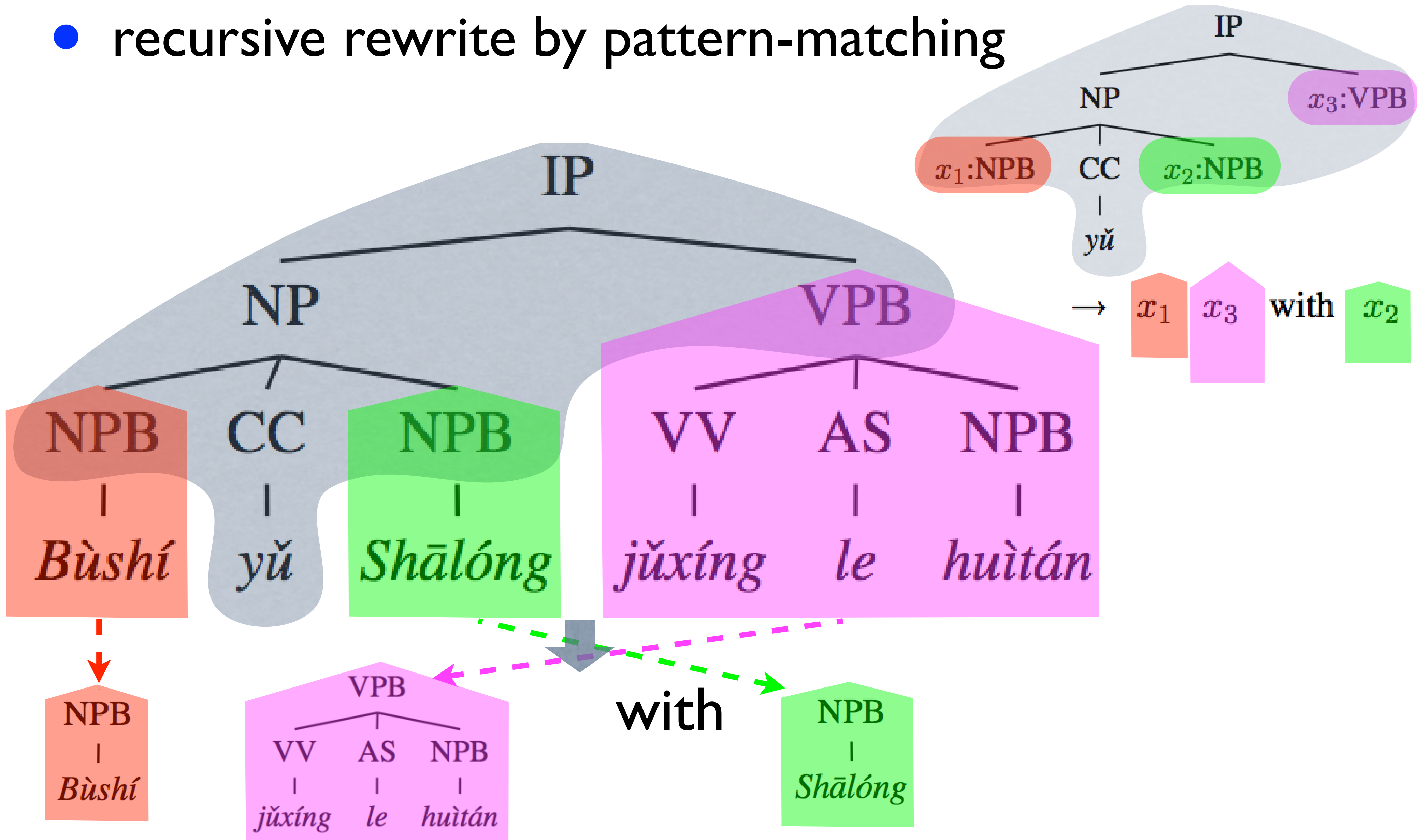
1. parse the source-language sentence into a tree
2. recursively convert it into a target-language sentence

Bùshí	*yǔ*	*Shālóng*	*jǔxíng*	*le*	*huìtán*
Bush	and/ with	Sharon	hold	[*past.*]	meeting

 (Irons 1961; Lewis, Stearns 1968; Aho, Ullman 1972) ==> (Huang, Knight, Joshi 2006)

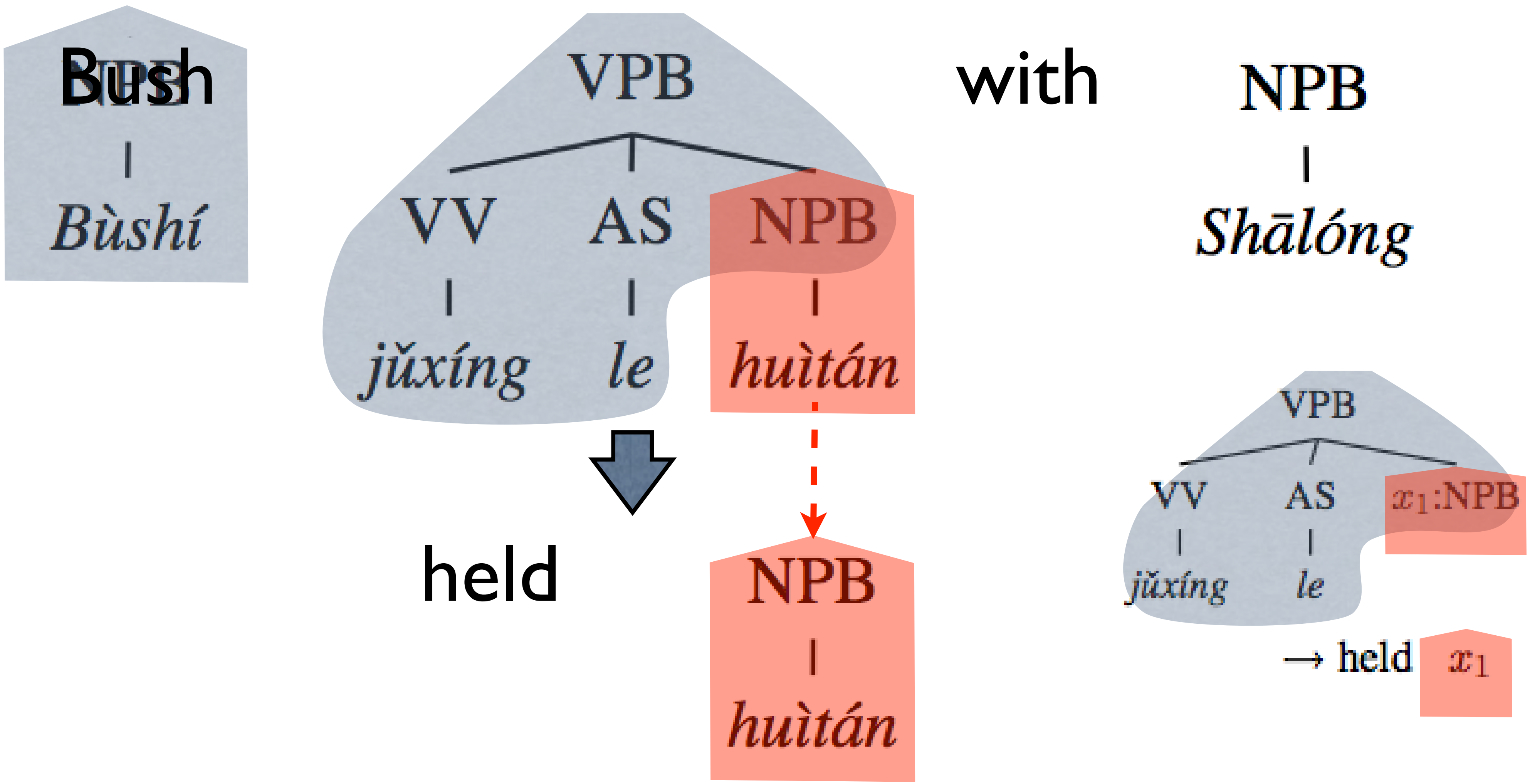
Syntax-Directed Machine Translation

- recursive rewrite by pattern-matching



Syntax-Directed Machine Translation?

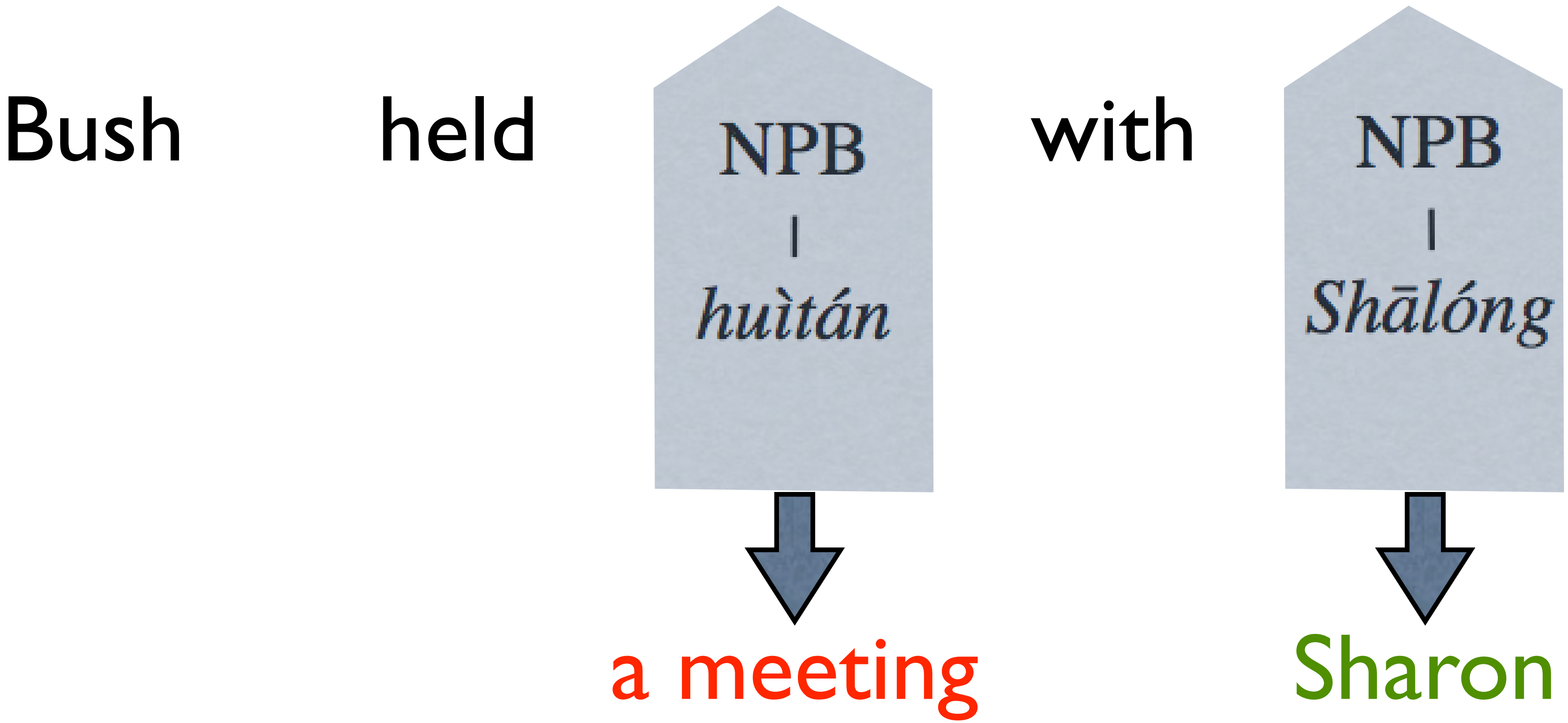
- recursively solve unfinished subproblems



(Huang, Knight, Joshi 2006); rules from (Galley et al., 04)

Syntax-Directed Machine Translation?

- continue pattern-matching



(Huang, Knight, Joshi 2006); rules from (Galley et al., 04)

Syntax-Directed Machine Translation?

- continue pattern-matching

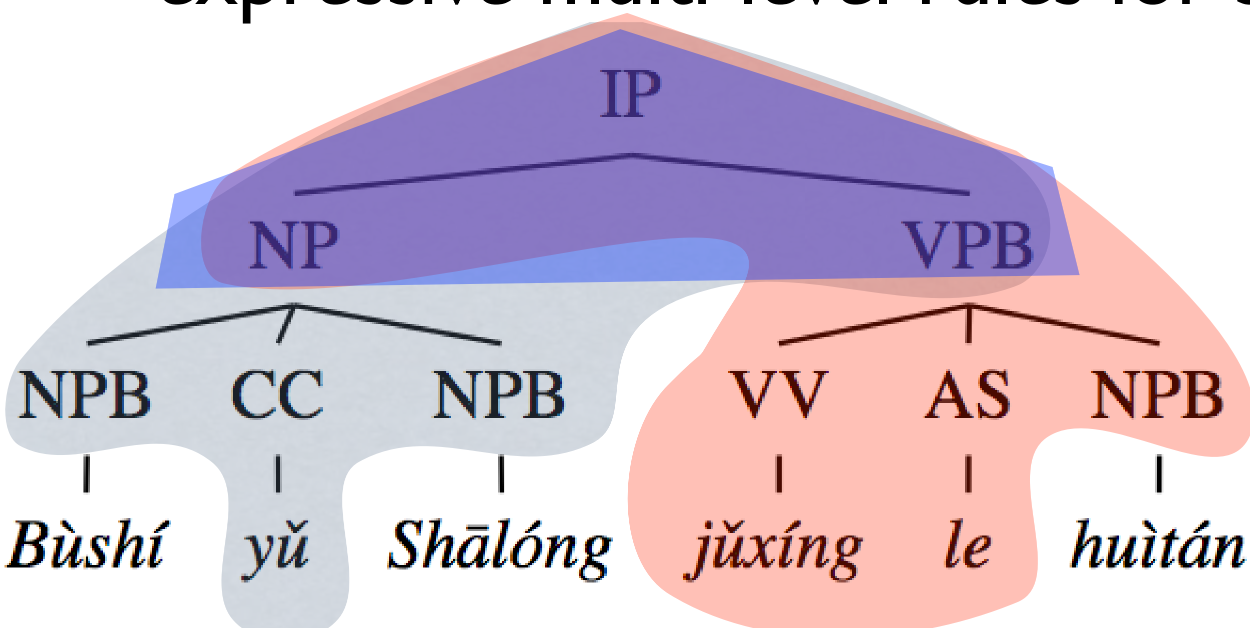
Bush held a meeting with Sharon

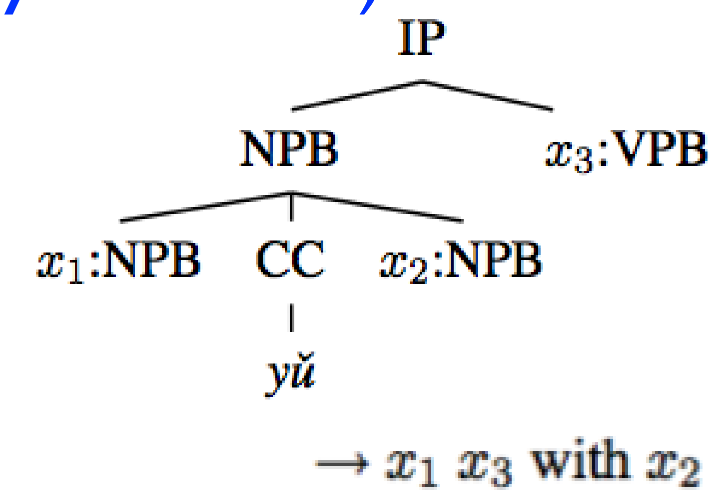
(Huang, Knight, Joshi 2006); rules from (Galley et al., 04)

Pros: simple, fast, and expressive

- simple architecture: separate parsing and translation
- efficient linear-time dynamic programming
  - "soft decision" at each node on which rule to use
  - (trivial) depth-first traversal with memoization
- expressive multi-level rules for syntactic divergence

(beyond CFG)





(Huang, Knight, Joshi 2006); rules from (Galley et al., 04)

Cons: Parsing Errors

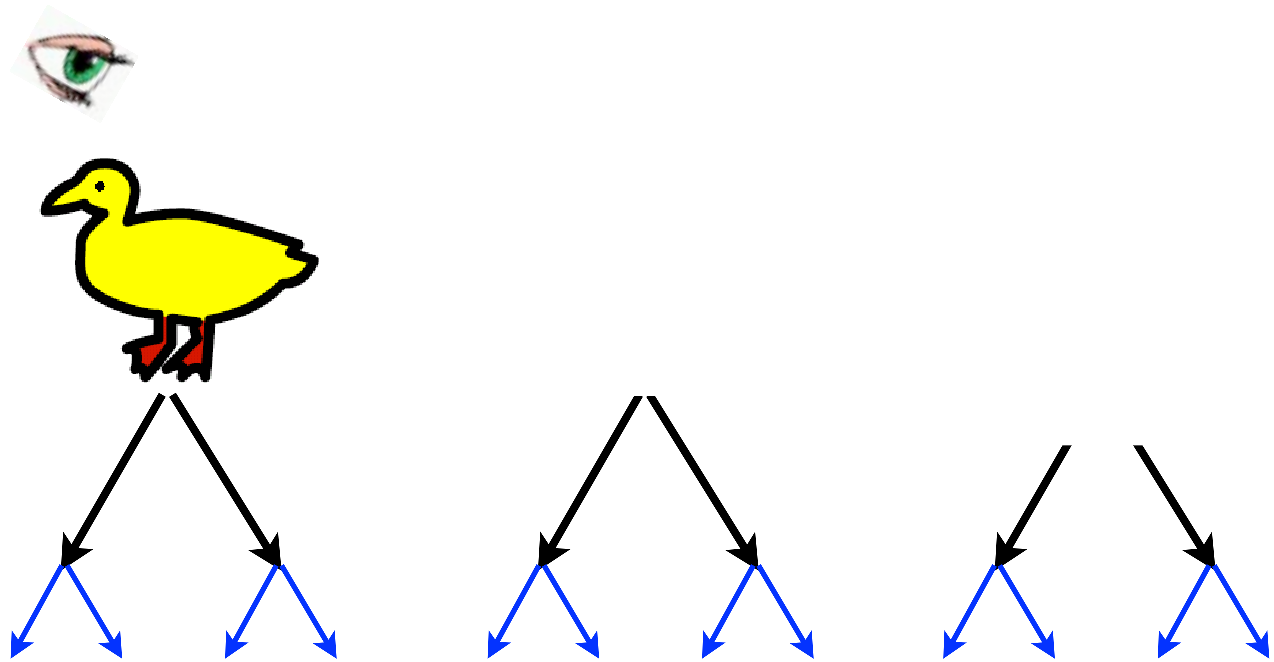
- ambiguity is a fundamental problem in natural languages
  - probably will never have perfect parsers (unlike compiling)
- parsing errors affect translation quality!

emergency exit
or “safe exports”?

mind your head
or “meet cautiously”?

Exponential Explosion of Ambiguity

I saw her duck.



...

- how about...
  - I saw her duck with a telescope.
  - I saw her duck with a telescope in the garden...

NLP == dealing with ambiguities.

Tackling Ambiguities in Translation

- simplest idea: take top-*k* trees rather than 1-best parse
  - but only covers tiny fraction of the exponential space
  - and these *k*-best trees are very similar

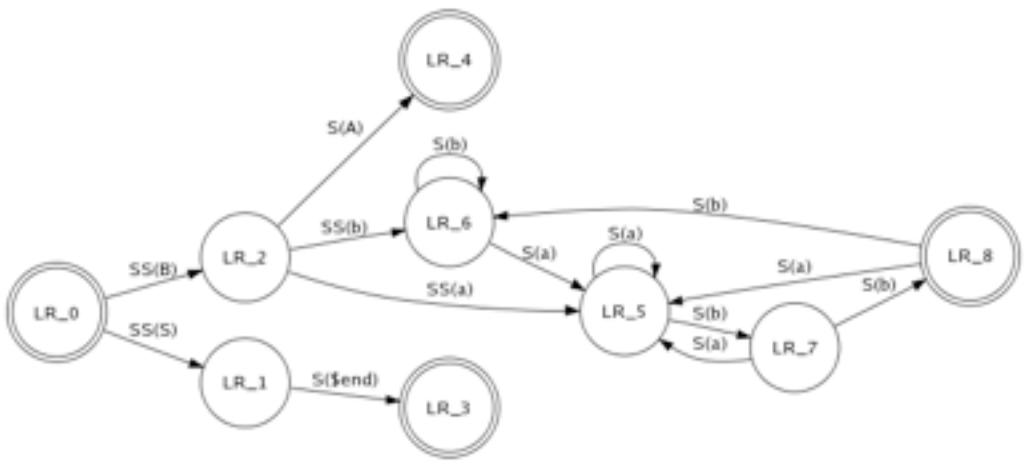
    - e.g., 50-best trees ~ 5-6 binary ambiguities ($2^5 < 50 < 2^6$)
    - very inefficient to translate on these very similar trees
- most ambitious idea: combining parsing and translation
  - start from the input string, rather than 1-best tree
  - essentially considering all trees (search space too big)
- our approach: *packed forest* *(poly. encoding of exp. space)*
  - almost as fast as 1-best, almost as good as combined

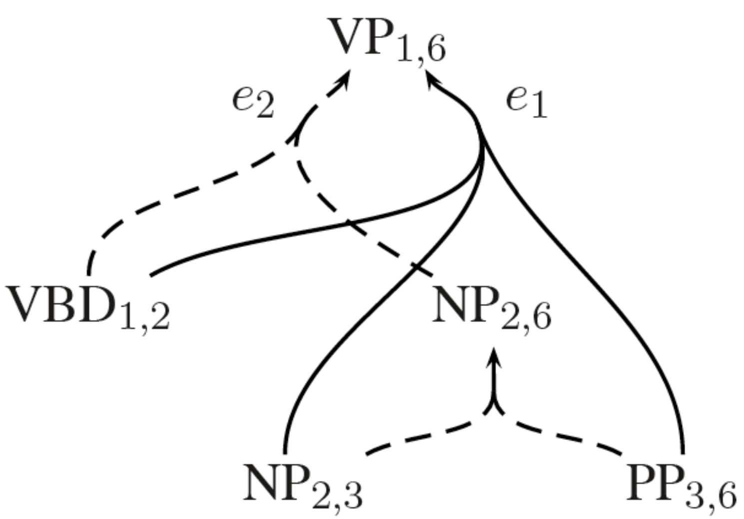
Outline

- Overview: Tree-based Translation
- Forest-based Translation
  - Packed Forest
  - Translation on a Forest
  - Experiments
- Forest-based Rule Extraction
  - Large-scale Experiments

From Lattices to Forests

- common theme: polynomial encoding of exponential space
  - forest generalizes "lattice/graph" from finite-state world
    - paths => trees (in DP: knapsack vs. matrix-chain multiplication)
    - graph => hypergraph; regular grammar => CFG

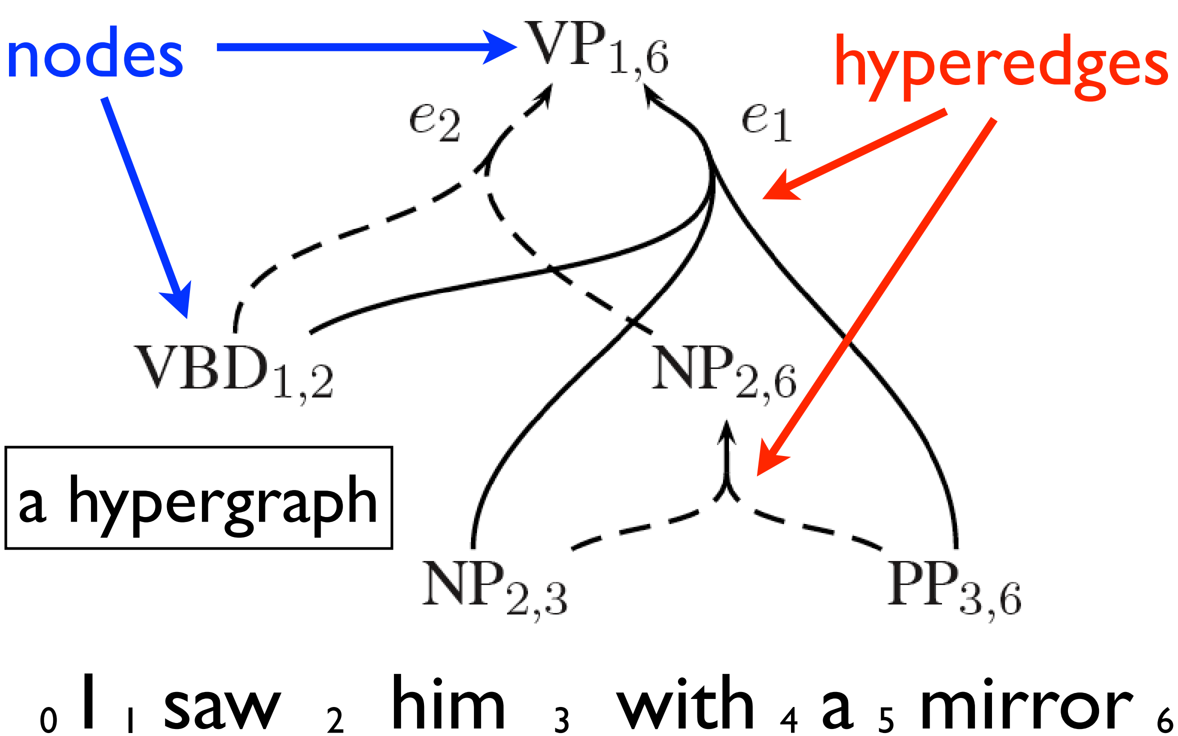




(Earley 1970; Billot and Lang 1989)

Packed Forest

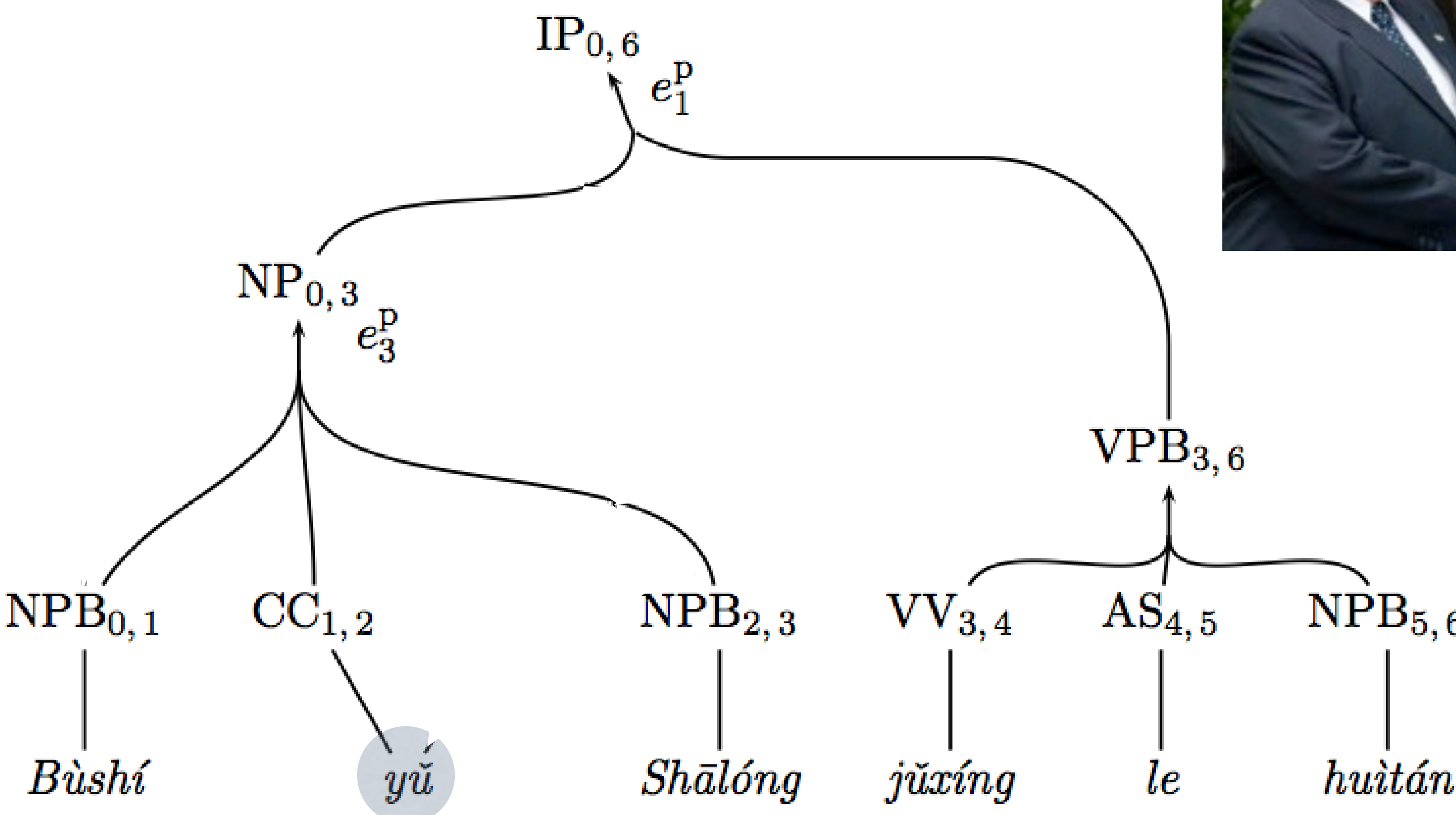
- a compact representation of many many parses
  - by sharing common sub-derivations
  - polynomial-space encoding of exponentially large set



$$e_1 \quad \frac{VBD_{1,2} \quad NP_{2,3} \quad PP_{3,6}}{VP_{1,6}}$$

(Klein and Manning, 2001; Huang and Chiang, 2005)

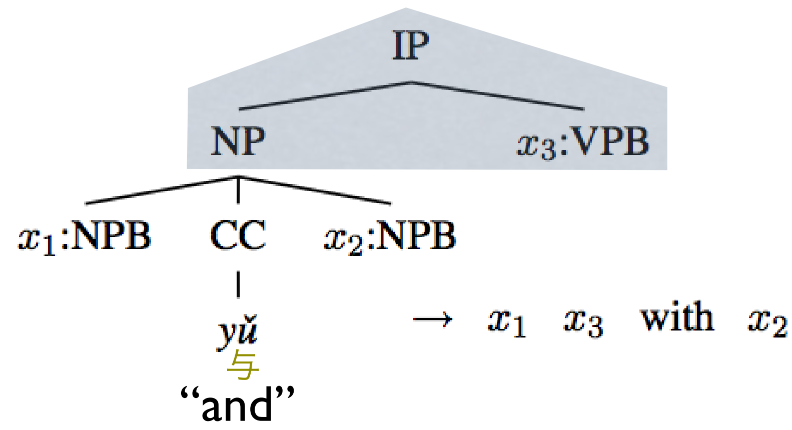
Forest-based Translation

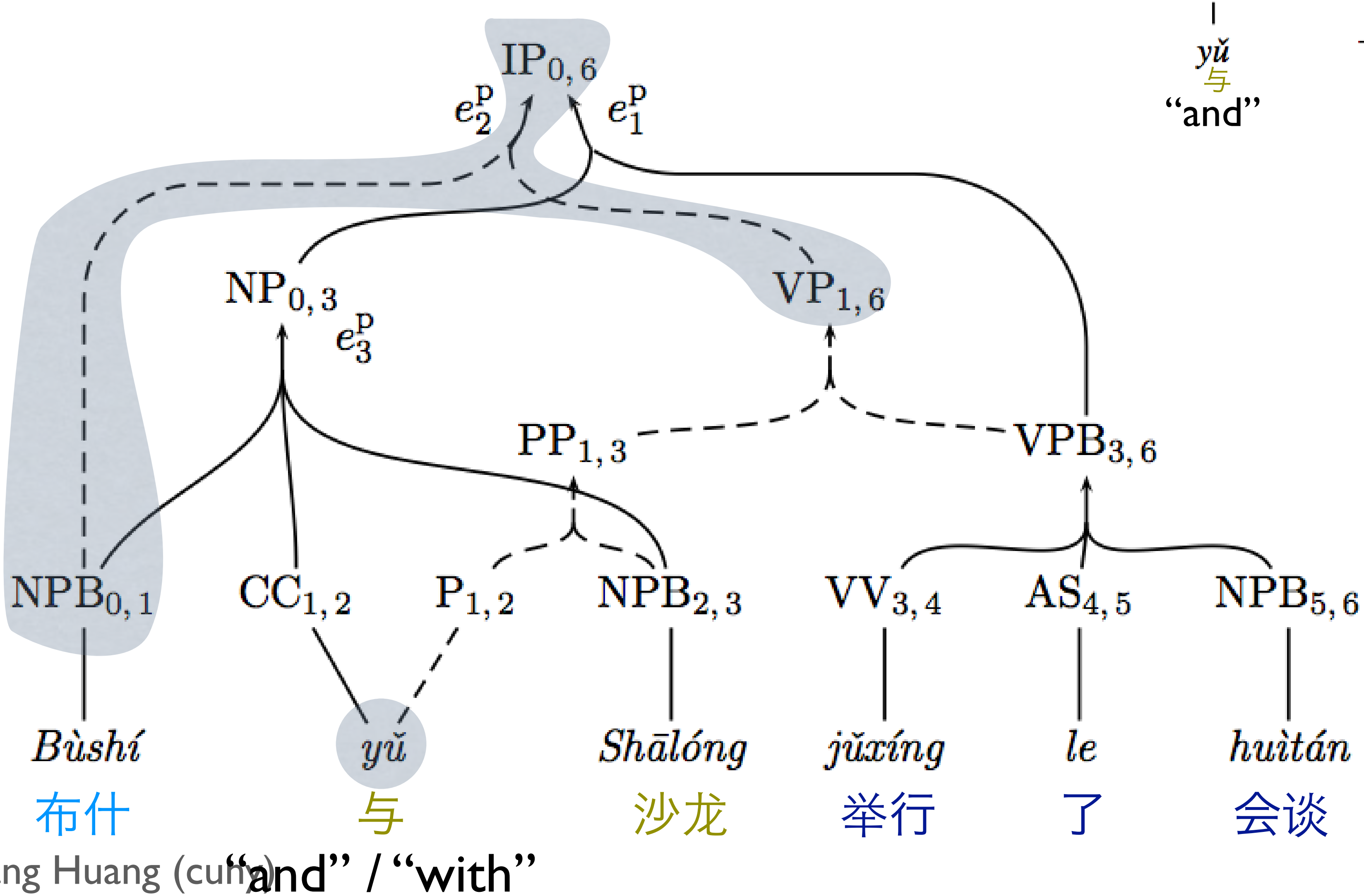


“and” / “with”

Forest-based Translation

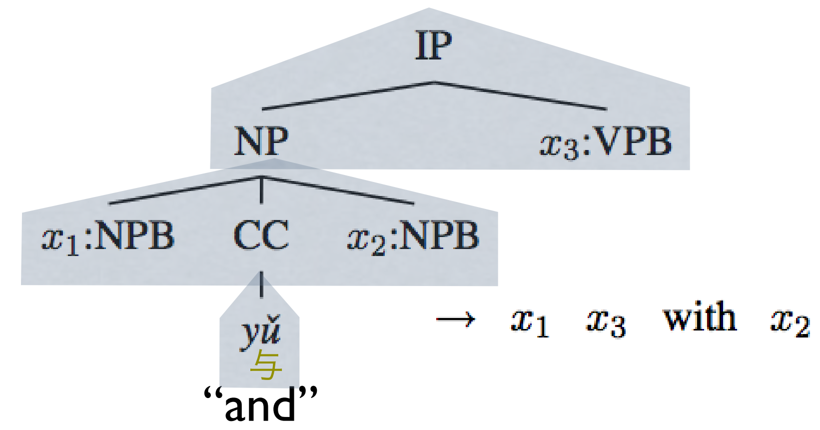
pattern-matching
on forest
(linear-time in forest size)

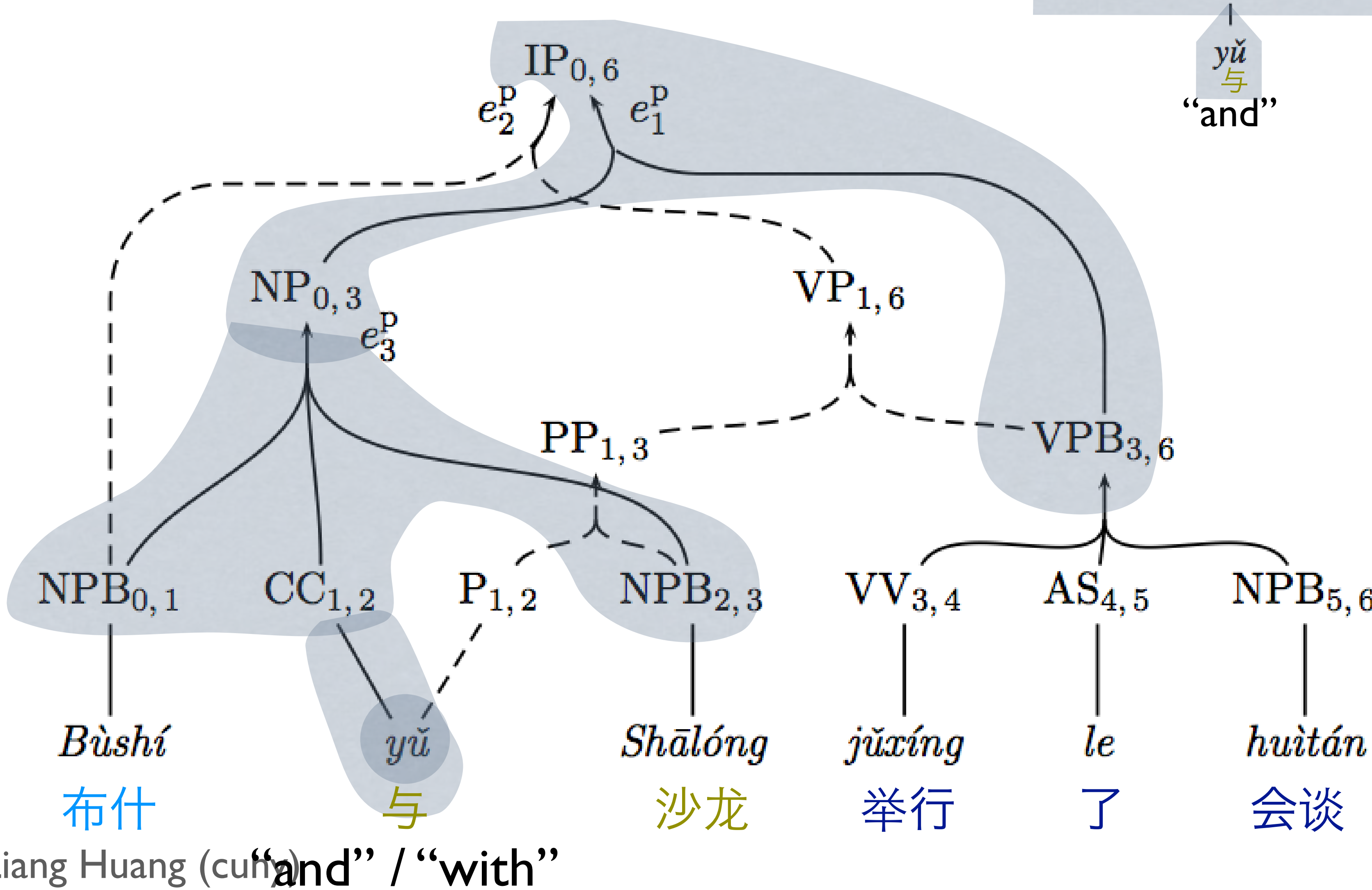




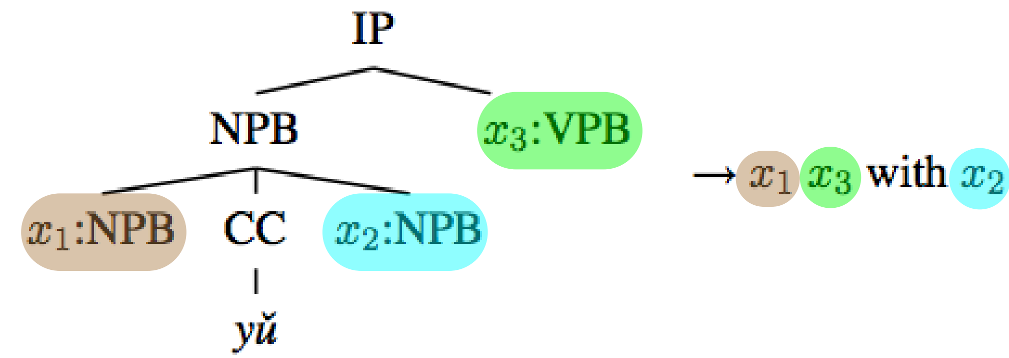
Forest-based Translation

pattern-matching
on forest
(linear-time in forest size)

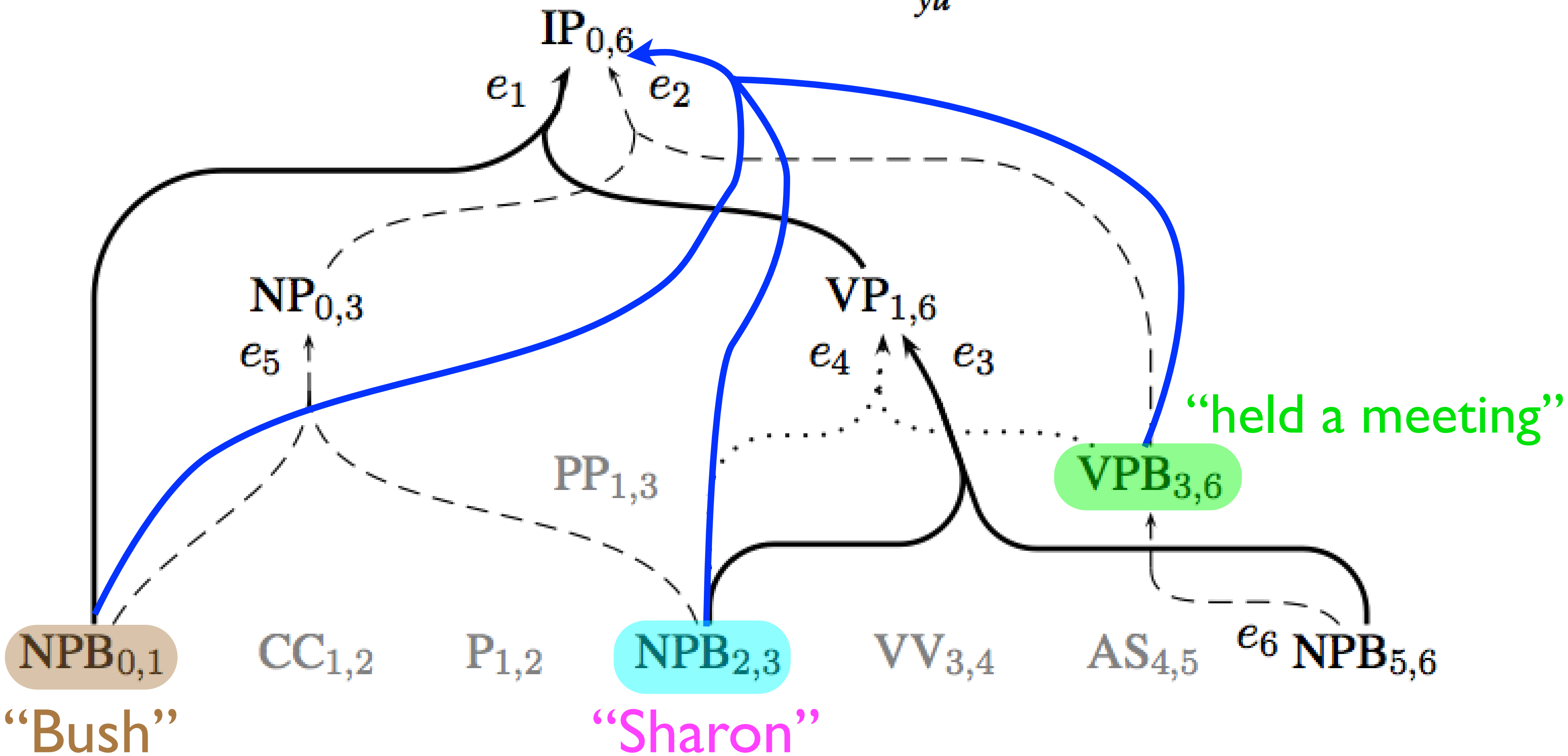




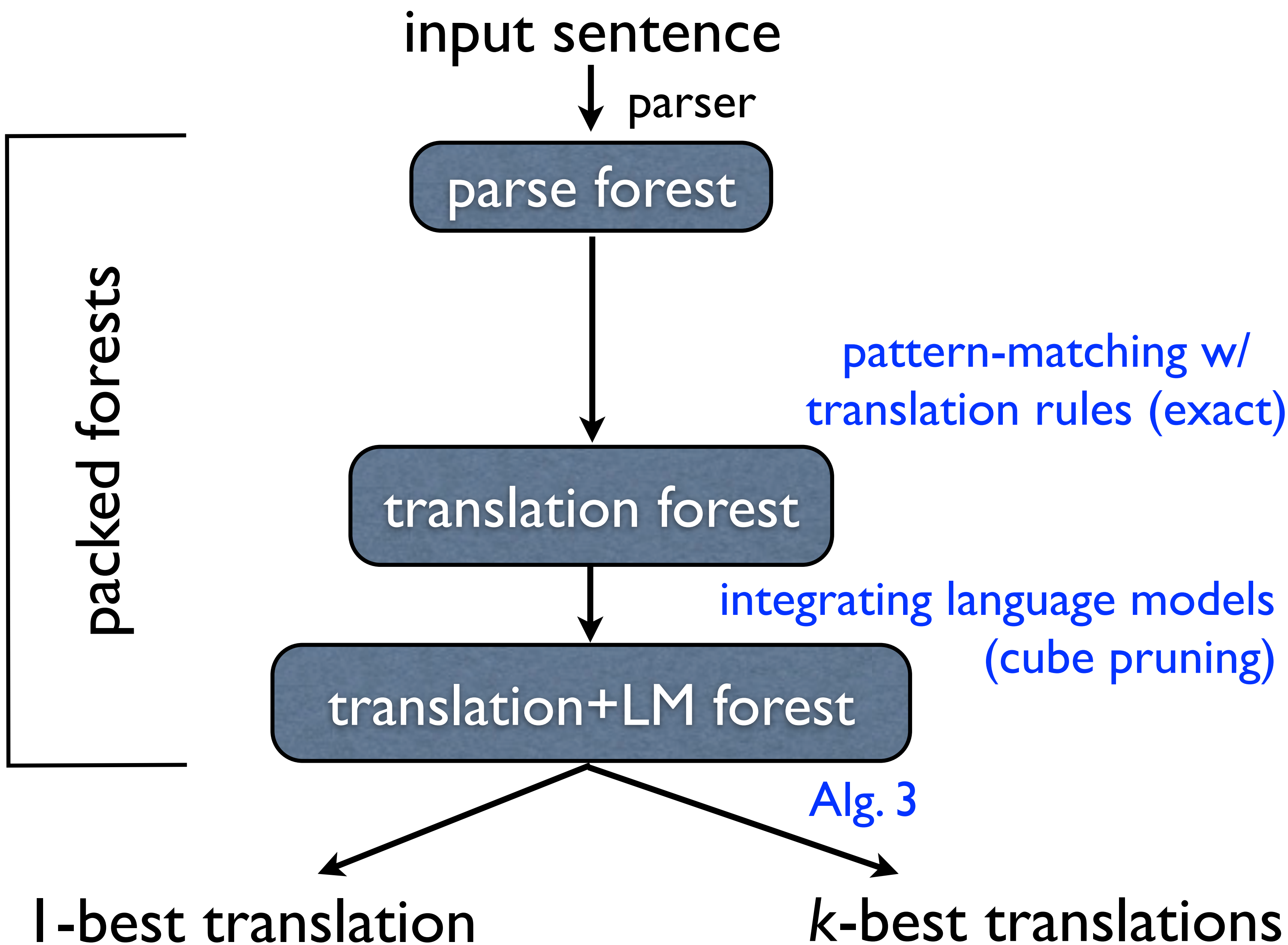
Translation Forest



"Bush held a meeting with Sharon"



The Whole Pipeline



(Huang and Chiang, 2005; 2007; Chiang, 2007)

The Whole Pipeline

input sentence

parser

parse forest

forest pruning

pruned forest

pattern-matching w/ translation rules (exact)

translation forest

integrating language models (cube pruning)

translation+LM forest

packed forests

Alg. 3

1-best translation

k-best translations

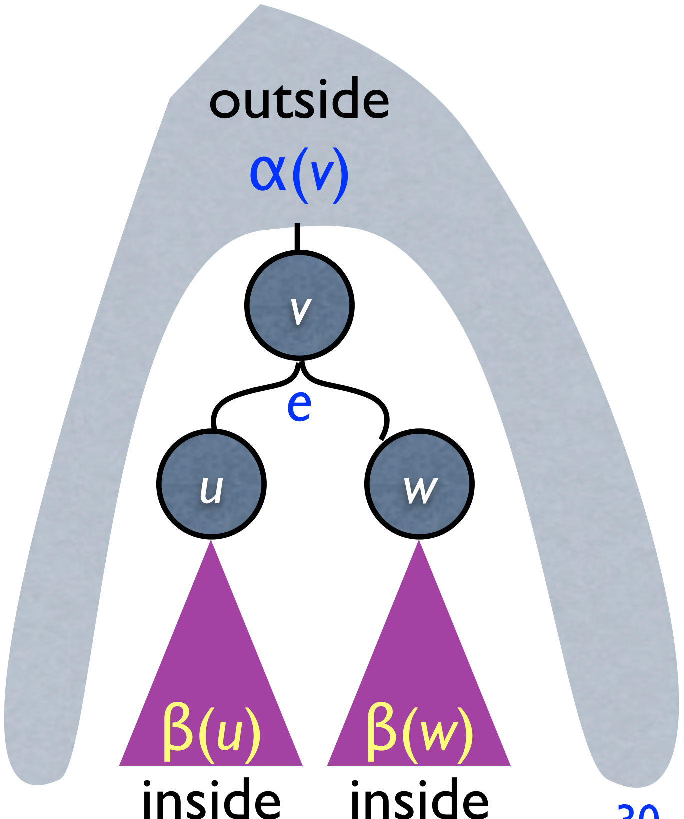
(Huang and Chiang, 2005; 2007; Chiang, 2007)

Parse Forest Pruning

- prune *unpromising* hyperedges
- principled way: inside-outside
  - first compute Viterbi inside β, outside α
- then $\alpha\beta(e) = \alpha(v) + c(e) + \beta(u) + \beta(w)$
  - cost of best deriv that traverses e
  - similar to "expected count" in EM
- prune away hyperedges that have

  $\alpha\beta(e)$ - $\alpha\beta(\textsc{top}) > p$

  for some threshold p



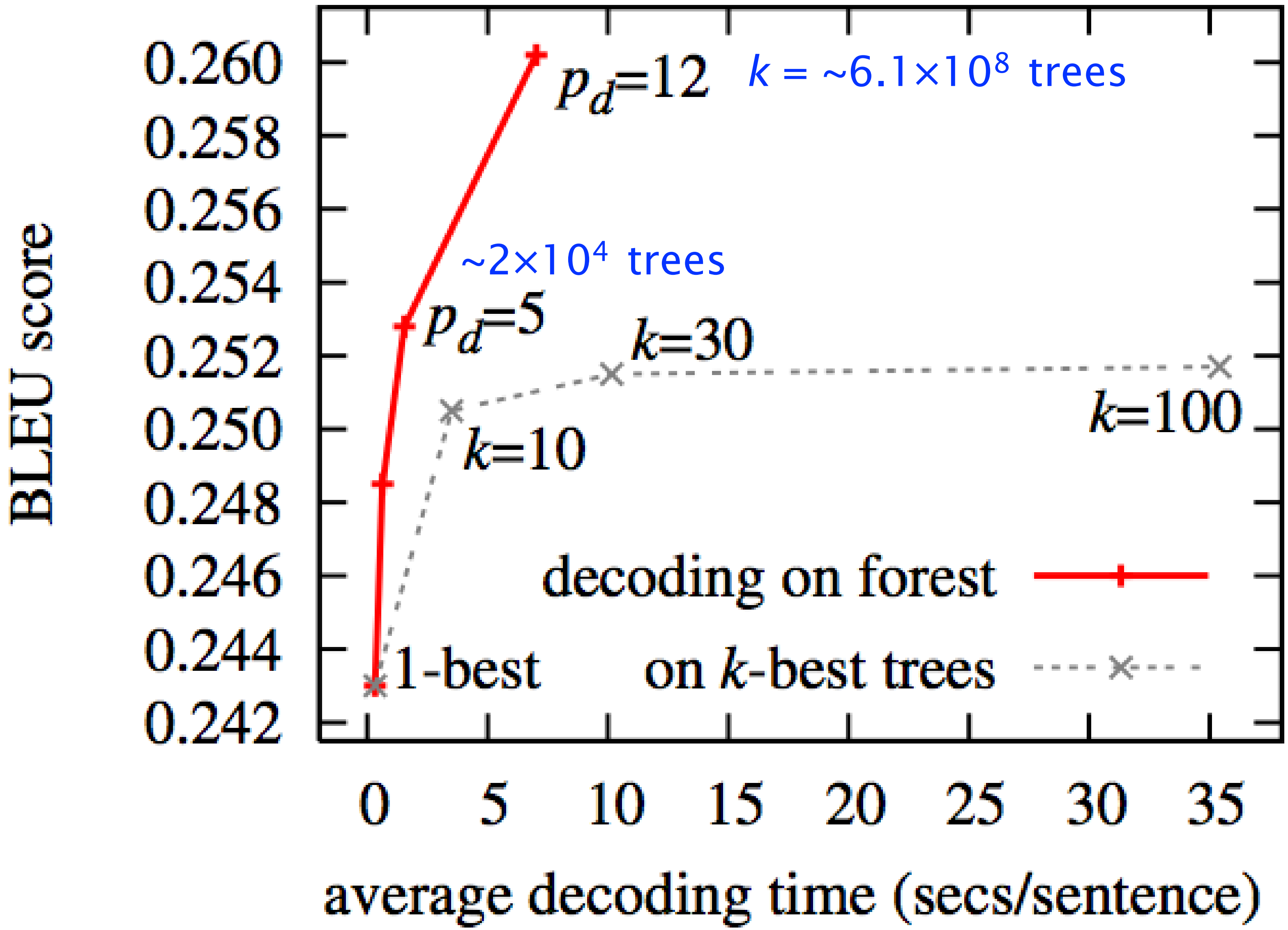
Jonathan Graehl: relatively useless pruning

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)
- BLEU score: 1-best baseline: 0.2430 vs. Pharaoh: 0.2297

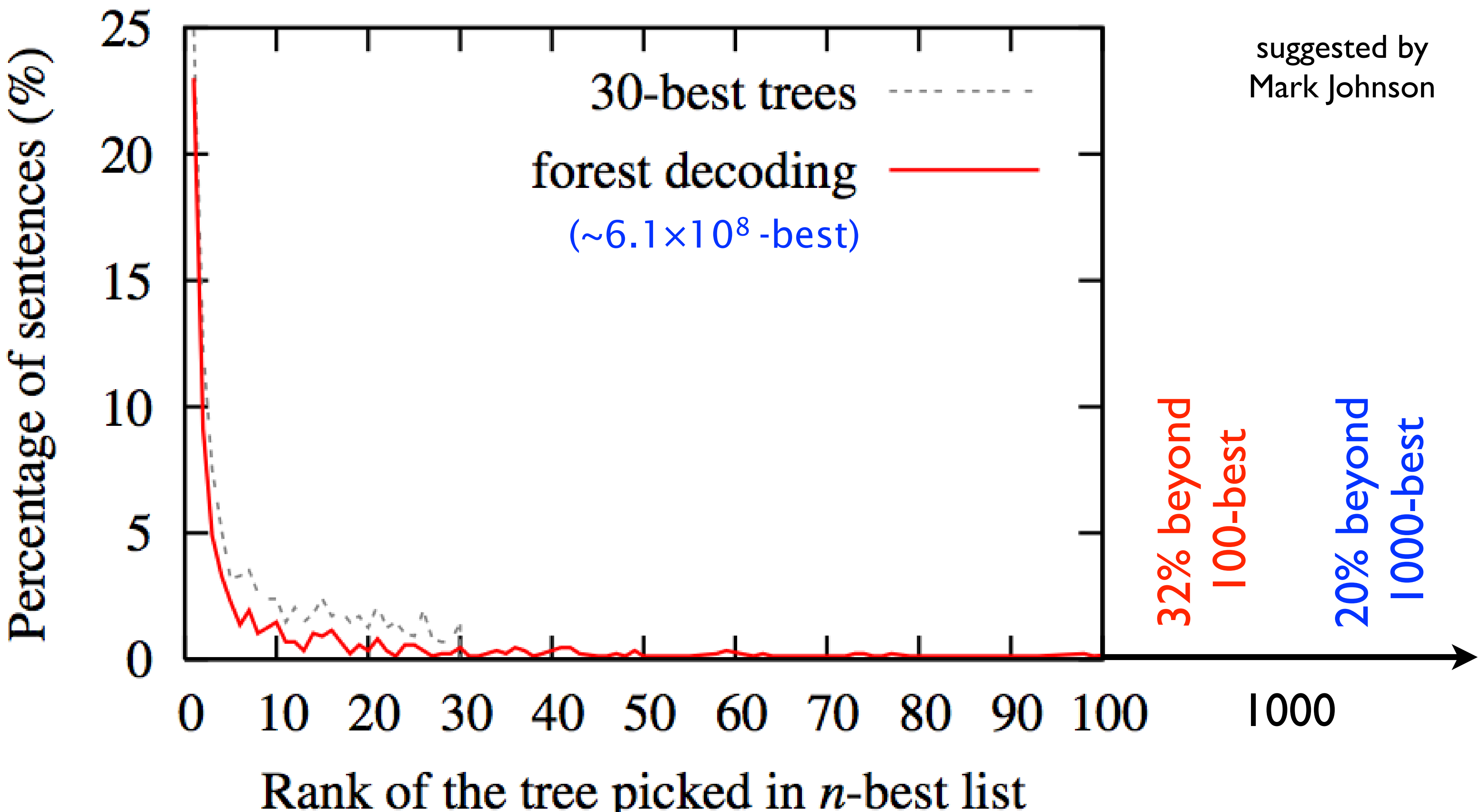
k-best trees vs. forest-based

1.7 Bleu improvement over 1-best,
0.8 over 30-best, and even faster!



forest as virtual ∞-best list

- how often is the *i*th-best tree picked by the decoder?



suggested by Mark Johnson

wait a sec... where are the rules from?

小心 VP <=> be careful not to VP

小心 NP <=> be careful of NP

. . .

xiǎoxīn gǒu
小心 狗 <=> be aware of dog

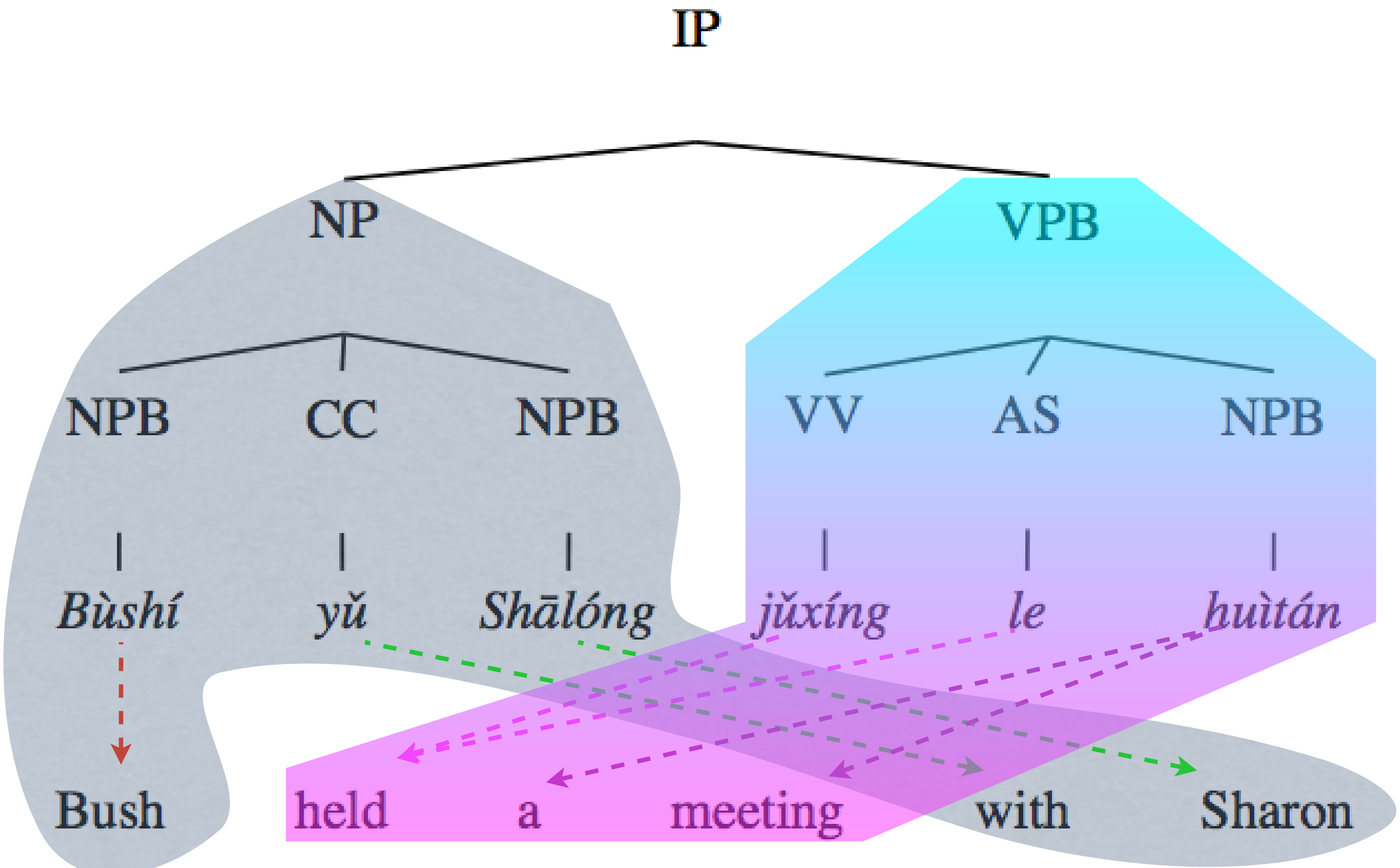
xiǎoxīn
小心 X <=> be careful not to X

Outline

- Overview: Tree-based Translation
- Forest-based Translation
- Forest-based Rule Extraction
  - background: tree-based rule extraction (Galley et al., 2004)
  - extension to forest-based
  - large-scale experiments

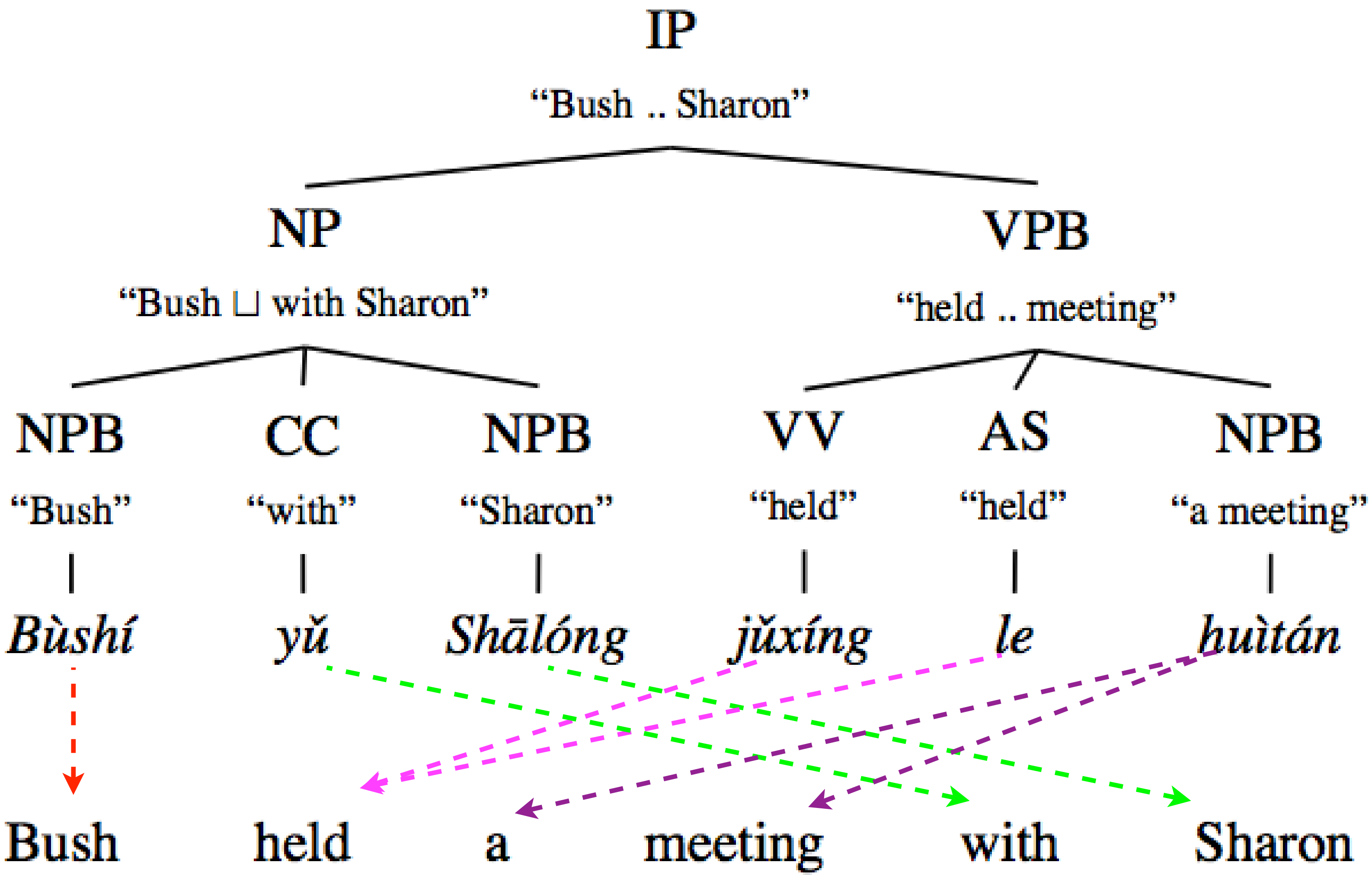
Where are the rules from?

- data: source parse tree, target sentence, and alignment
- intuition: fragment the tree; contiguous span



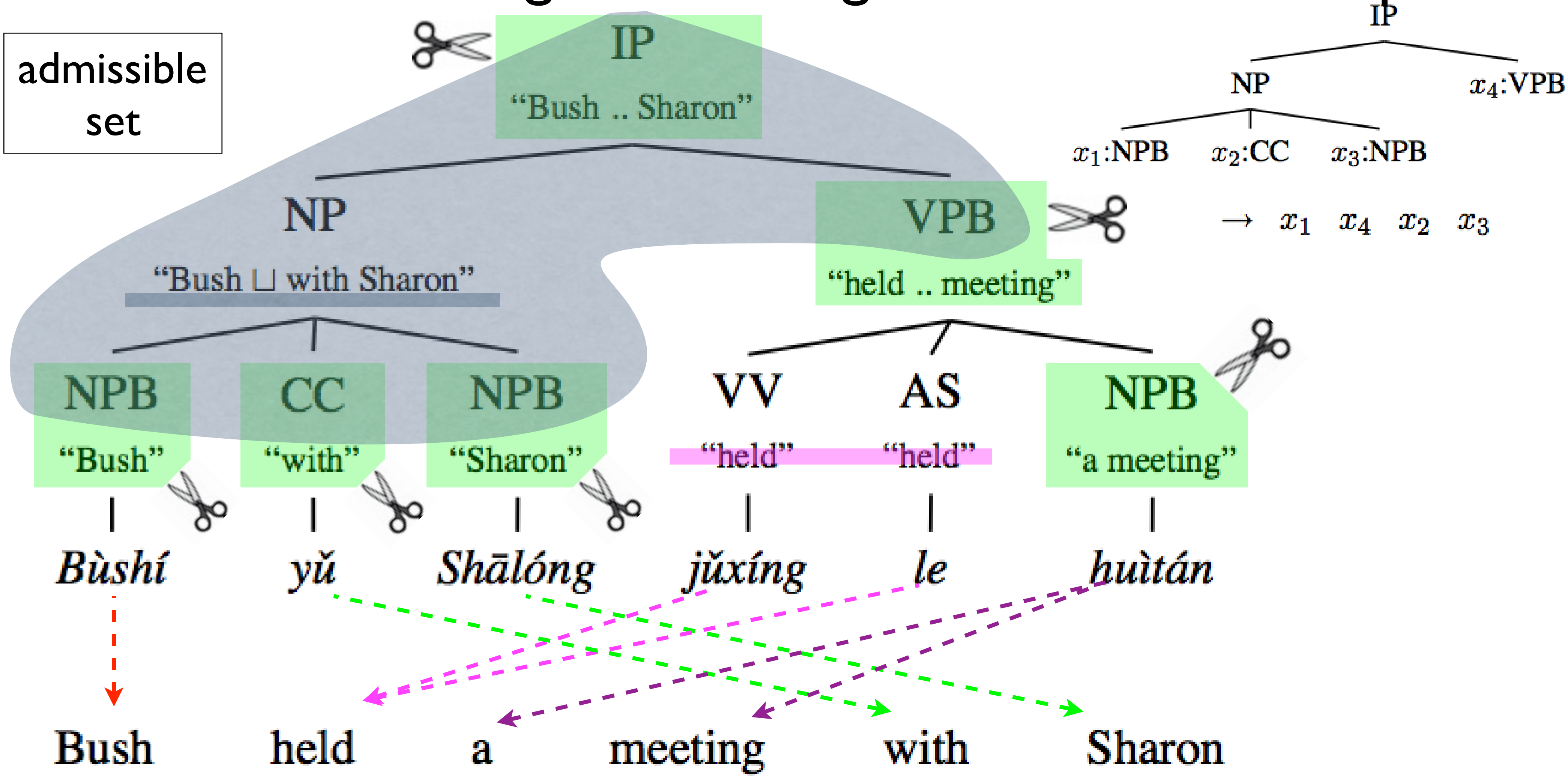
Where are the rules from?

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



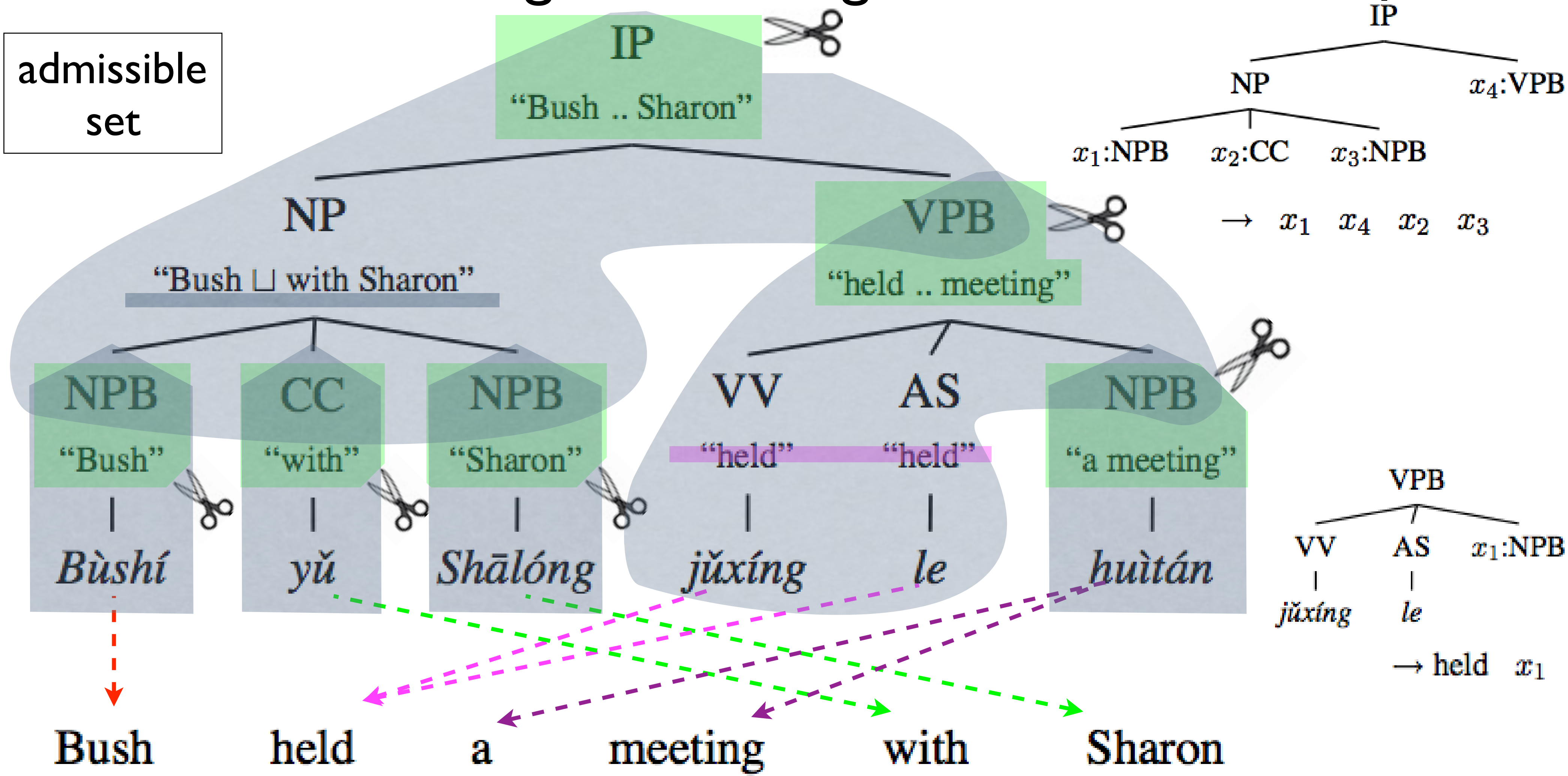
Where are the rules from?

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



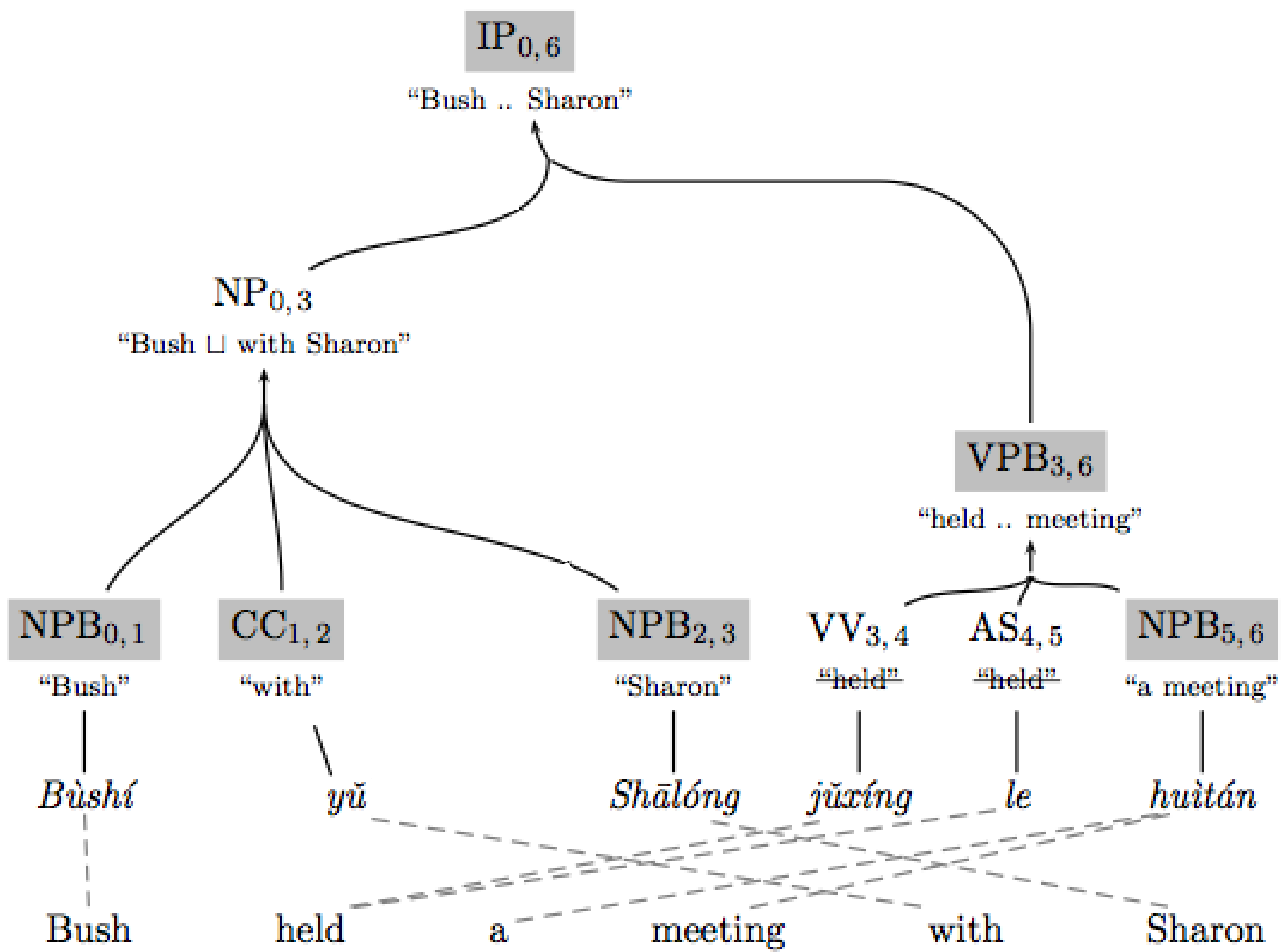
Where are the rules from?

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



Forest-based Rule Extraction

- same cut set computation; different fragmentation

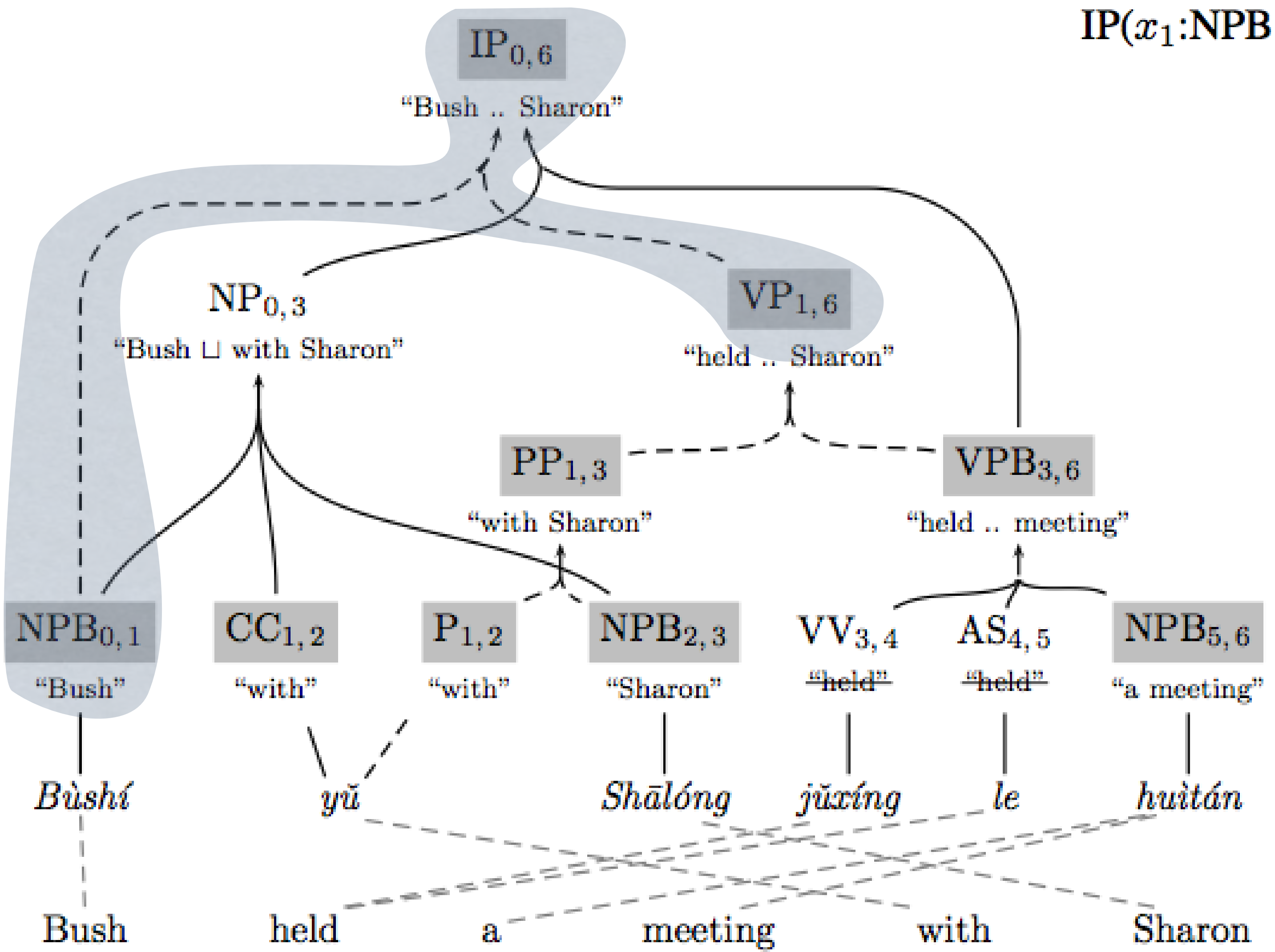


also in (Wang, Knight, Marcu, 2007)

Forest-based Rule Extraction

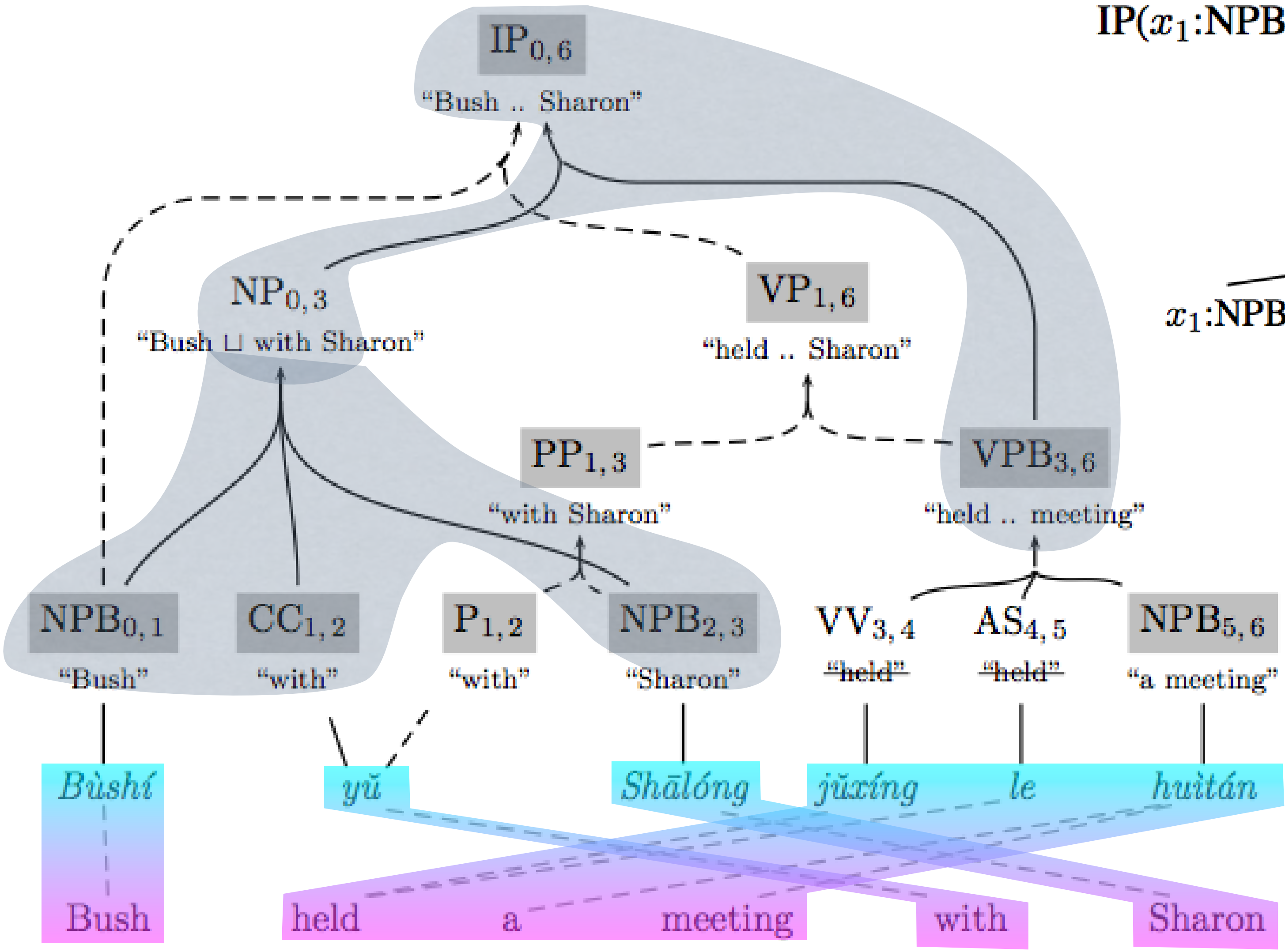
- same cut set computation; different fragmentation

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

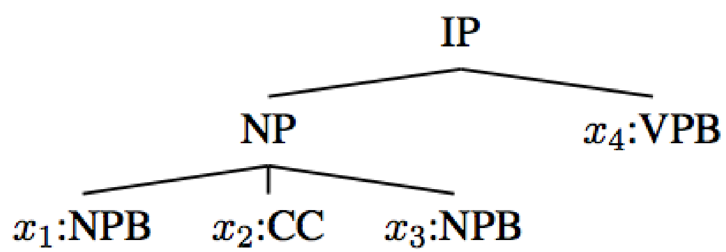


Forest-based Rule Extraction

- same admissible set definition; different fragmentation



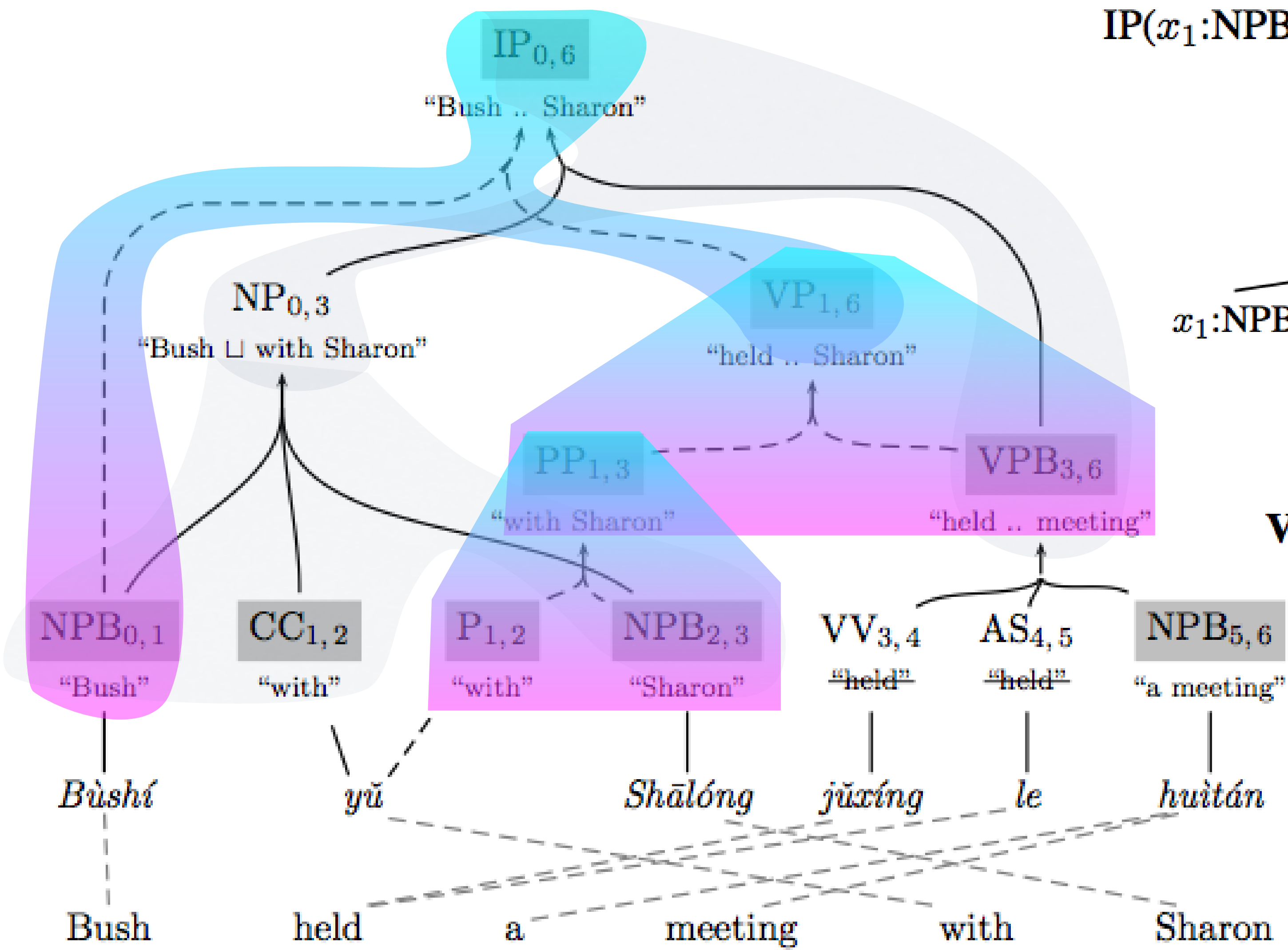
$\mathrm{IP}(x_1{:}\mathrm{NPB}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$



$\rightarrow\ x_1\ x_4\ x_2\ x_3$

Forest-based Rule Extraction

- forest can extract smaller chunks of rules



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

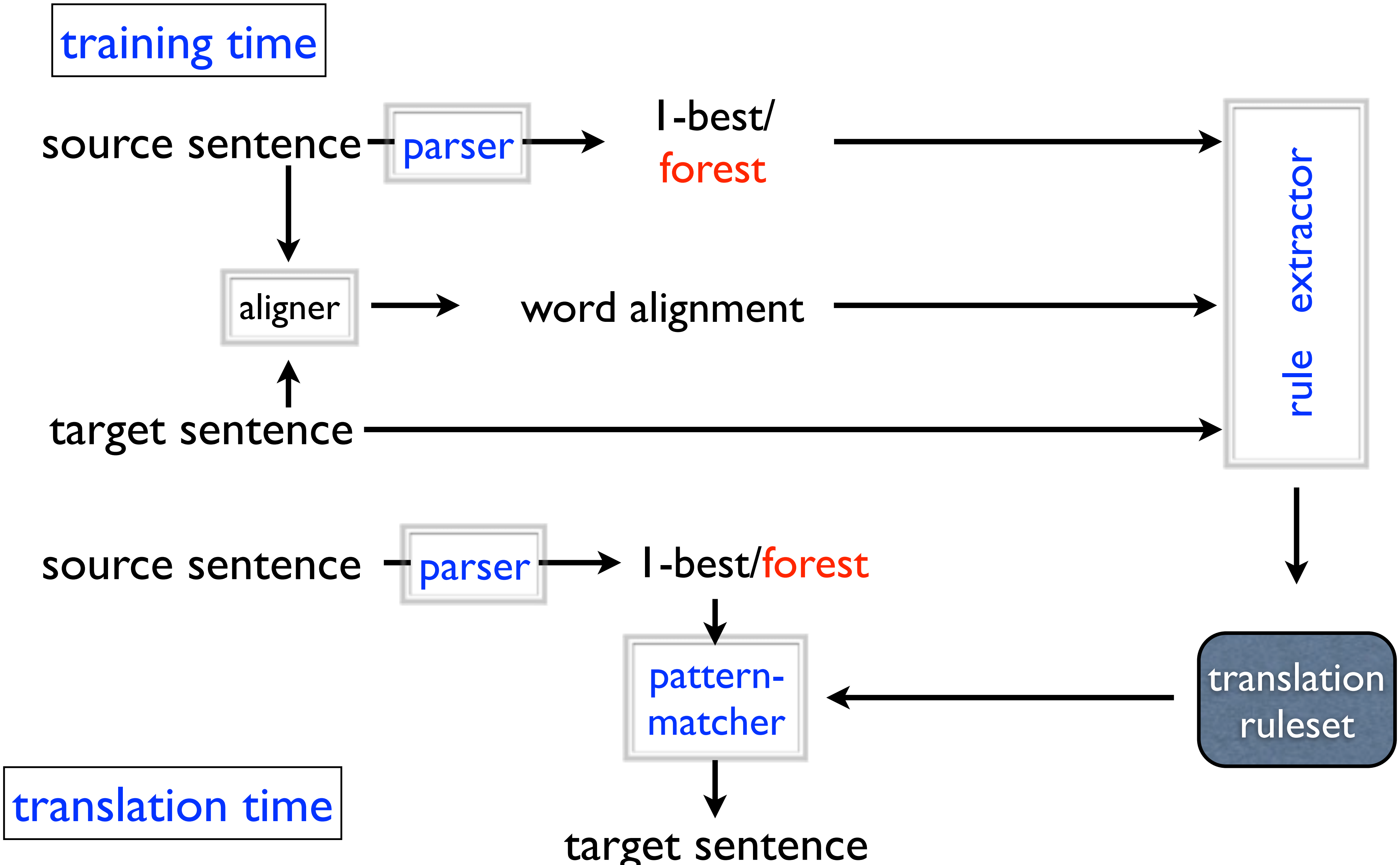
$\mathrm{IP}(\mathrm{NP}(x_1\mathrm{:NPB}\ x_2\mathrm{:CC}\ x_3\mathrm{:NPB})\ x_4\mathrm{:VPB})$

$\rightarrow\ x_1\ x_4\ x_2\ x_3$

$\mathbf{VP}\ (x_1\mathrm{:PP}\ \ x_2\mathrm{:VPB}) \rightarrow x_2\ x_1$

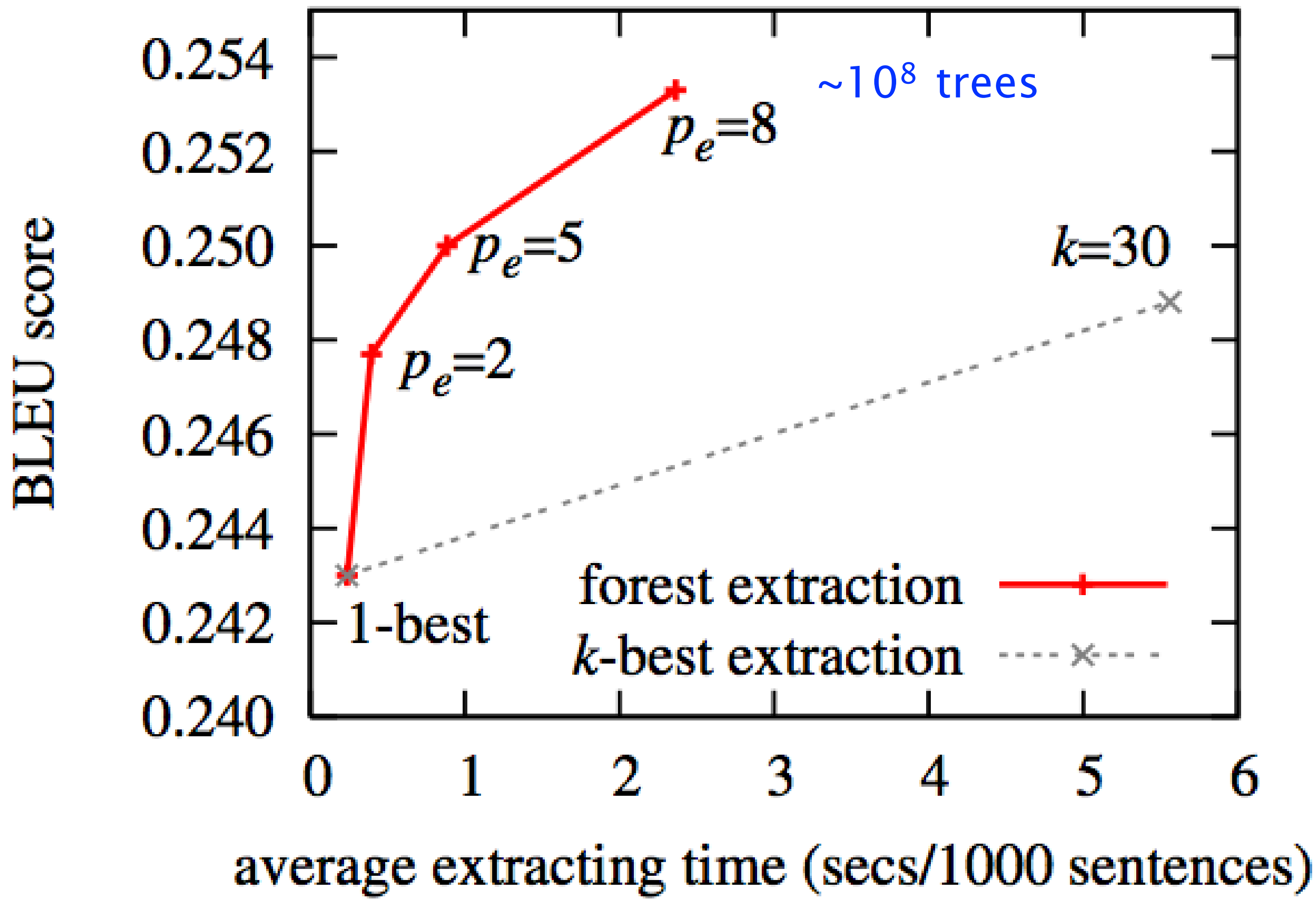
$\mathbf{PP}\ (x_1\mathrm{:P}\ \ x_2\mathrm{:NPB}) \rightarrow x_1\ x_2$

The Forest^2 Pipeline



Forest vs. *k*-best Extraction

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



Forest^2

- 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 (Chiang 2007) by quite a bit

translating on ...

rules from ...

<table>
<tr><th></th><th>1-best tree</th><th>forest</th></tr>
<tr><td>1-best tree</td><td>0.2560</td><td>0.2674</td></tr>
<tr><td>30-best trees</td><td>0.2634</td><td>0.2767</td></tr>
<tr><td>forest</td><td>0.2679</td><td>0.2816</td></tr>
<tr><td>Hiero</td><td colspan="2">0.2738</td></tr>
</table>

Translation Examples

• src	鲍威尔	说	与	阿拉法特	会谈	很	重要
	Bàowēir	shūo	yǔ	Alāfǎtè	huìtán	hěn	zhòngyào
	Powell	say	with	Arafat	talk	very	important

- 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

Conclusions

- main theme: efficient syntax-directed translation
- forest-based translation
  - forest = "underspecified syntax": polynomial vs. exponential
  - still fast (with pruning), yet does not commit to 1-best tree
  - translating millions of trees is faster than just on top-*k* trees
- forest-based rule extraction: improving rule set quality
- very simple idea, but works well in practice
  - significant improvement over 1-best syntax-directed
  - final result outperforms hiero by quite a bit

Forest is your friend in machine translation.

help save the forest.

Larger Decoding Experiments (ACL)

- 2.2M sentence pairs (57M Chinese and 62M English words)
- larger trigram models (1/3 of Xinhua Gigaword)
- also use bilingual phrases (BP) as flat translation rules
  - phrases that are consistent with syntactic constituents
- forest enables larger improvement with BP

	T2S	T2S+BP
1-best tree	0.2666	0.2939
30-best trees	0.2755	0.3084
forest	0.2839	0.3149
improvement	1.7	2.1