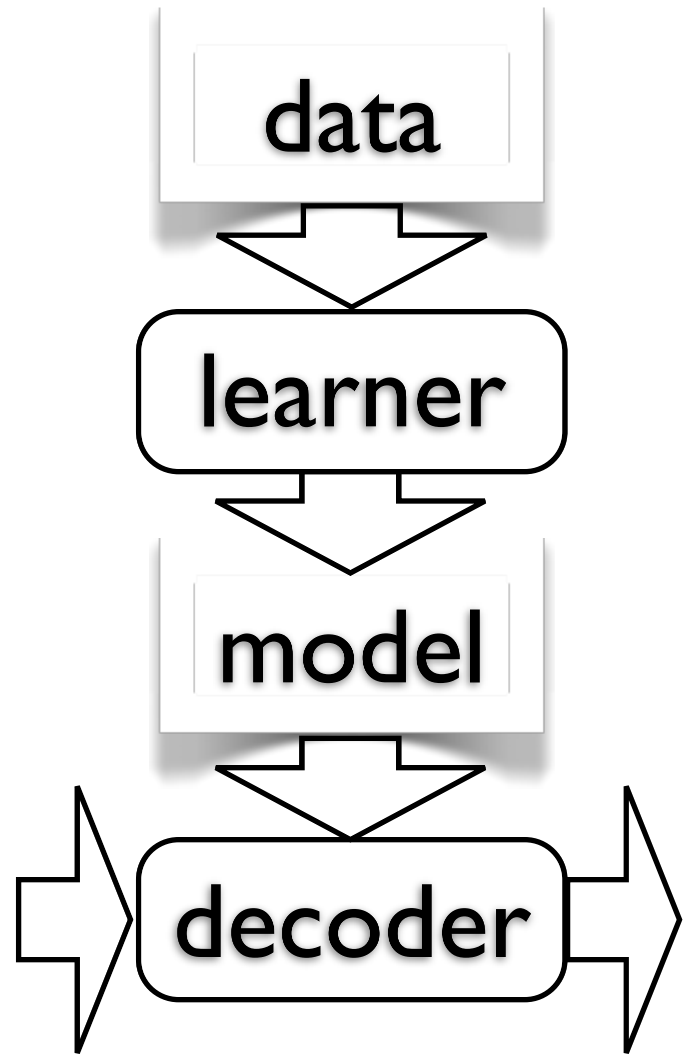
Structures in Statistical Machine Translation

Taro Watanabe @ NICT
taro . watanabe @ nict . go . jp

Machine Translation

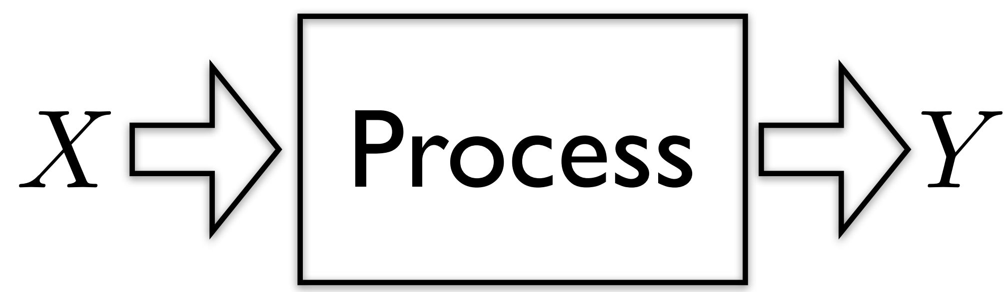
黑山头口岸联检部门将原来要二至三天办完的出入境手续改为一天办完。



The United Inspection Department of Heishantou Port has shortened the procedures for leaving and entering the territory from originally 2 - 3 days to 1 day.

- We learn parameters from data assuming a “model”
- Decode by the learned parameters

Channel Model



Channel Model

+ noise

$$
\begin{aligned}
\hat{y} &= \operatorname*{argmax}_y Pr(y|x) \\
&= \operatorname*{argmax}_y \frac{Pr(x|y)Pr(y)}{Pr(x)} \\
&= \operatorname*{argmax}_y Pr(x|y)Pr(y)
\end{aligned}
$$

f = source

e = target

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} Pr(\mathbf{f}|\mathbf{e})Pr(\mathbf{e})$$

- Employed in: ASR, OCR, MT...

Translation Model

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \underbrace{\boxed{Pr(\mathbf{f}|\mathbf{e})}}_{\text{Translation Model}} \underbrace{\boxed{Pr(\mathbf{e})}}_{\text{Language Model}}$$

(Brown et al., 1990)

- Translation Model: adequacy of translation
- Language Model: grammatical correctness, consistent style, fluency

Language Model

$$
\begin{aligned}
Pr(\text{I do not know}) &= ? \\
Pr(\text{I not do know}) &= ?
\end{aligned}
$$

- Likelihood of a string of English words
- Usually modeled by ngrams

$$W = w_1, w_2, w_3, \cdots w_N$$

$$
\begin{aligned}
p(W) &= p(w_1, w_2, w_3, \cdots, w_N) \\
&= p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\cdots \\
&\qquad p(w_N|w_1, w_2, w_3, \cdots, w_{N-1})
\end{aligned}
$$

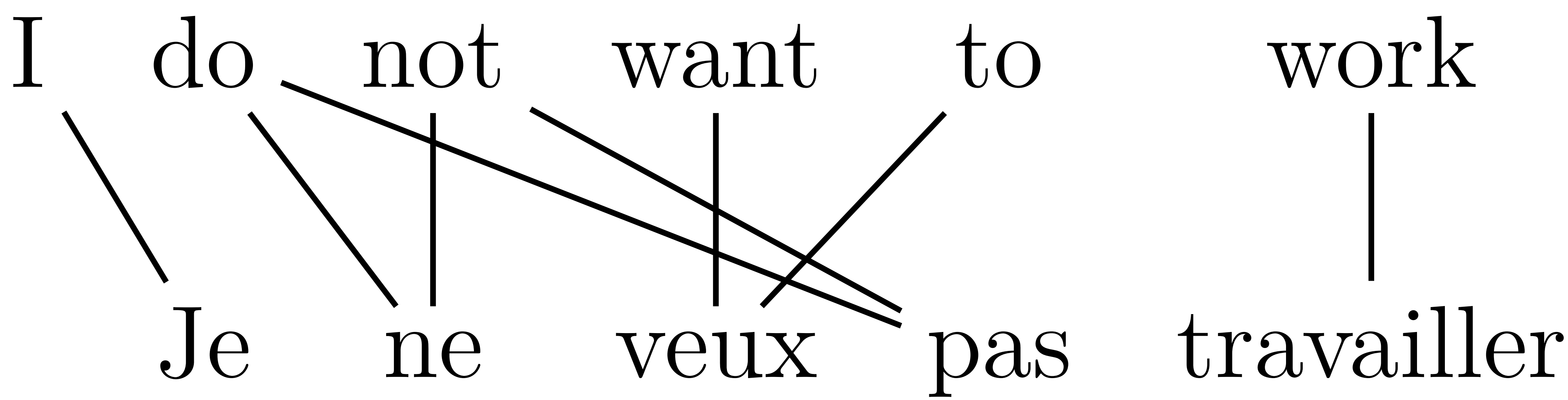
ngram Language Model

- Markov assumption: only n-word history is memorized
- Bigram:

$$p(\text{I do not know}) \quad = \quad p(\text{I})p(\text{do}|\text{I})p(\text{not}|\text{do})p(\text{know}|\text{not})$$

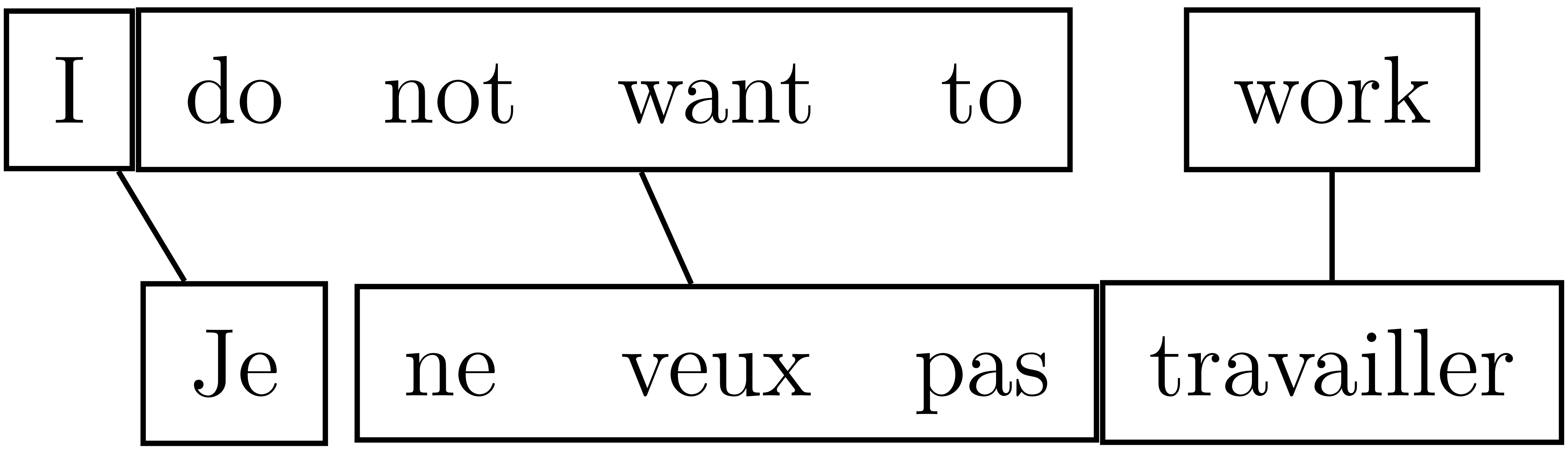
- Training: Maximum likelihood estimate + smoothing (Good-Turing, Witten-Bell, Kneser-Ney etc.)

Word-based MT



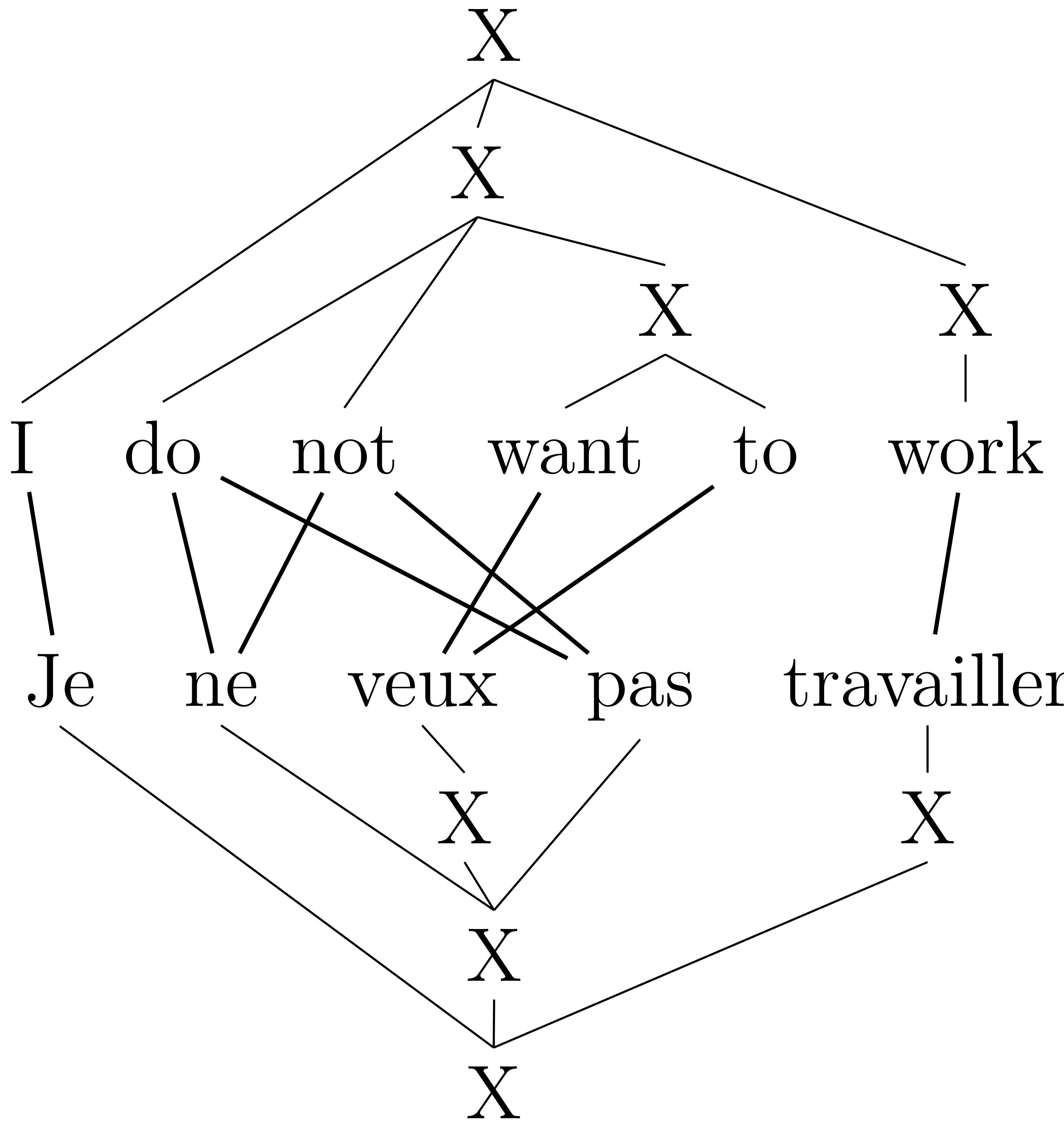
(Brown et al., 1993)

Phrase-based MT



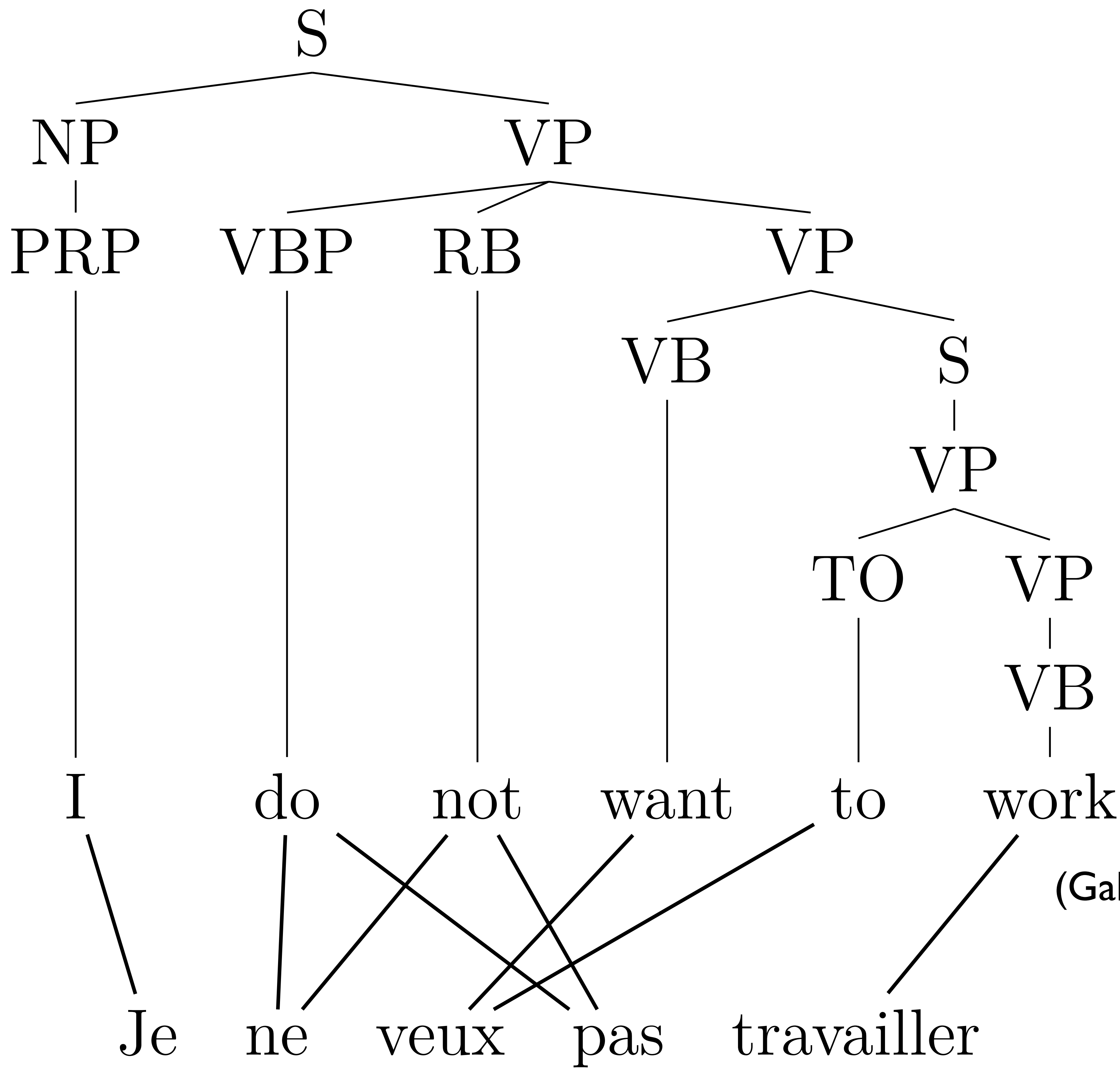
(Koehn et al., 2003)

Hierarchical PBMT



(Chiang, 2007)

Syntax-based MT



(Galley et al., 2004)

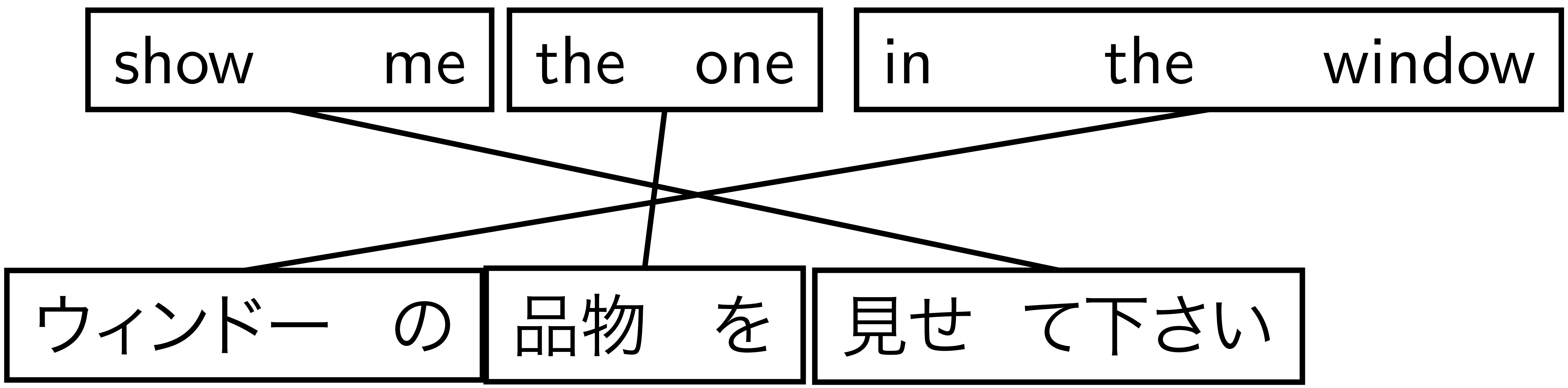
Structures in SMT

- Tutorial
  - Phrase-based MT
  - Tree-based MT
- Syntactic Structures in System Combination

Why Phrases?

- Use phrases as a unit of translations
  - Directly handle many-to-many word correspondence + local reordering
  - Allow local context + non-compositional phrases
- Employed in many systems, including Google, NICT(VoiceTra, TexTra) and open-source, Moses (http://www.statmt.org/moses/)

Phrase-based Model



- Generative story:
  - f is segmented into phrases
  - Each phrase is translated
  - Translated phrases are reordered

Phrase-based Model

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} \frac{\exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\right)}{\sum_{\mathbf{e}', \phi'} \exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}', \phi', \mathbf{f})\right)} \\ &= \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\end{aligned}$$

- Maximization of a log-linear combination of multiple feature functions h(e, Φ, f)
- Φ: phrasal partition of f and e
- w: weight of feature functions

Questions

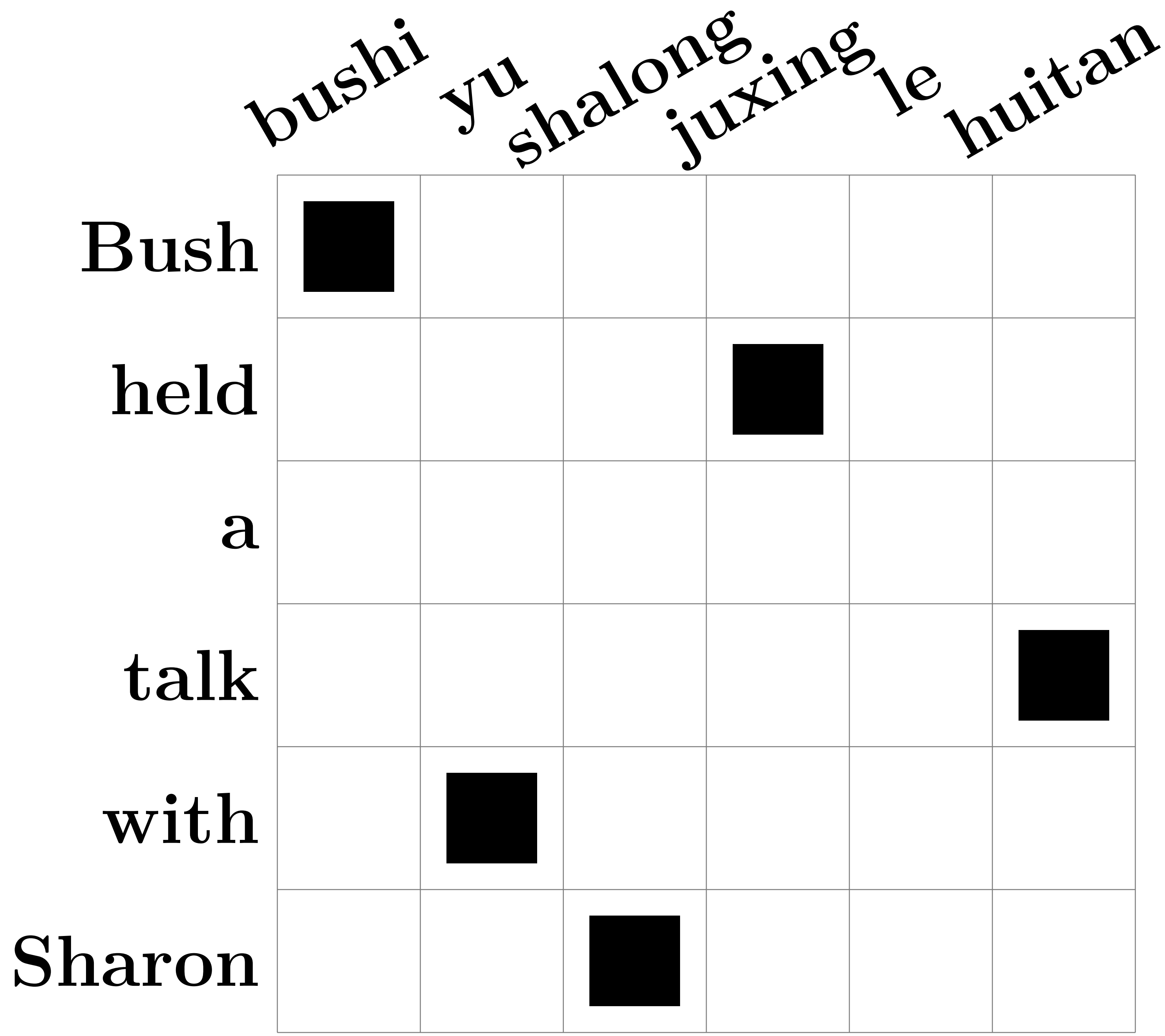
$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})$$

- Training: How to learn phrases and parameters (Φ and h)?
- Decoding (or search): How to find the best translation (argmax)?
- Tuning (or optimization): How to learn the scaling of features (w)?

Training

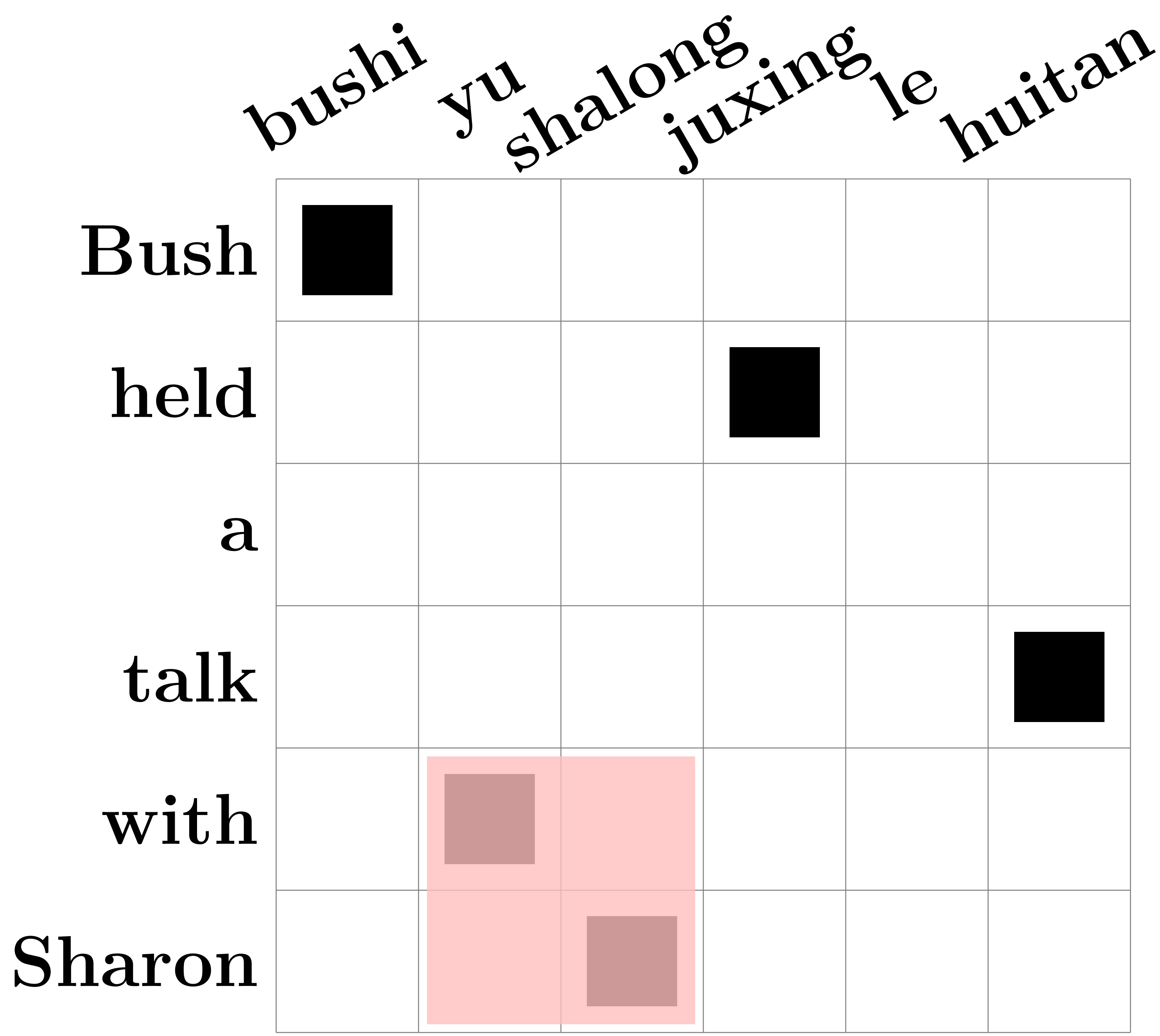
- Learn phrase pairs from $\mathcal{D} = \langle \mathcal{F}, \mathcal{E} \rangle$
- A standard heuristic approach (Koehn et al., 2003)
  - Compute word alignment
  - Extract phrase pairs
  - Score phrases

Word alignment



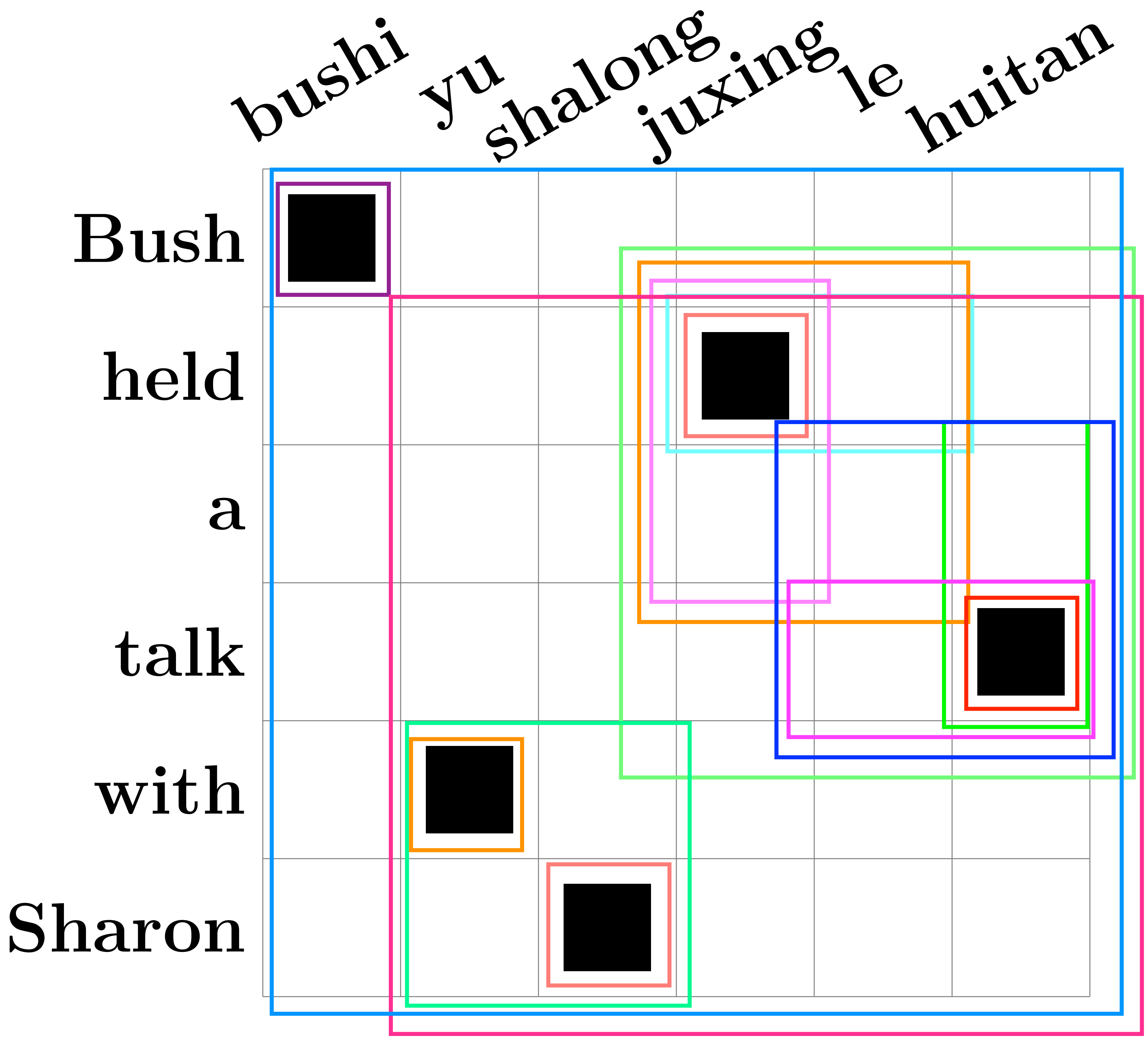
(Example from Huang and Chiang, 2007)

Extract Phrase Pairs



- From word alignment, extract a phrase pair consistent with word alignment

Exhaustive Extraction



- Exhaustively extract phrases from f, e

Features from Phrases

$$
\begin{aligned}
\log p_\phi(\bar{\mathbf{f}}|\bar{\mathbf{e}}) &= \log \frac{\mathrm{count}(\bar{\mathbf{e}}, \bar{\mathbf{f}})}{\sum_{\bar{\mathbf{f}}'} \mathrm{count}(\bar{\mathbf{e}}, \bar{\mathbf{f}}')} \\
\log p_\phi(\bar{\mathbf{e}}|\bar{\mathbf{f}}) &= \log \frac{\mathrm{count}(\bar{\mathbf{e}}, \bar{\mathbf{f}})}{\sum_{\bar{\mathbf{e}}'} \mathrm{count}(\bar{\mathbf{e}}', \bar{\mathbf{f}})}
\end{aligned}
$$

- Collect all the phrase pairs from the data
- Maximum likelihood estimates by relative frequencies
- Employ scores in two directions

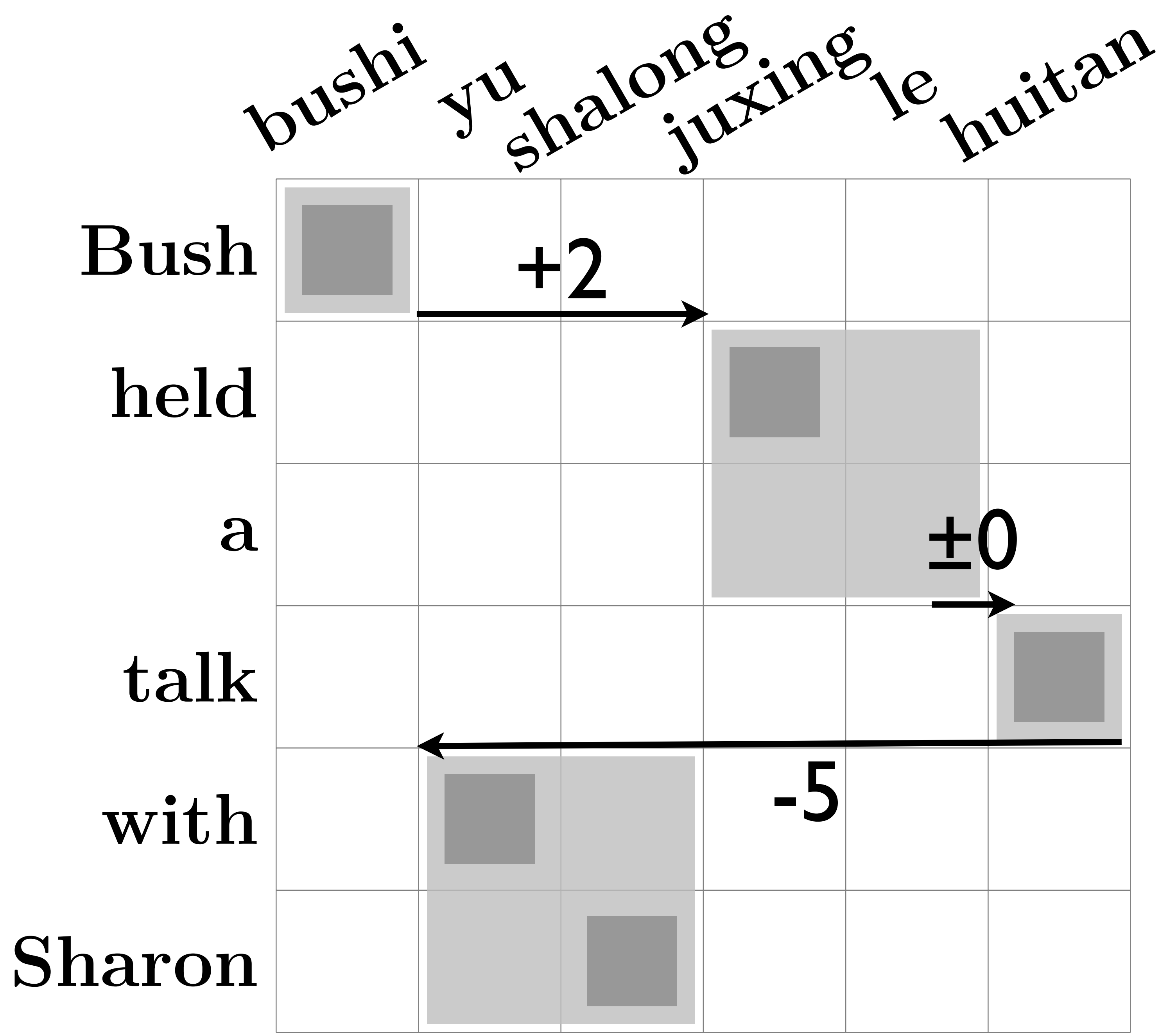
Features from Alignment

$$\log p_{lex}(\bar{\mathbf{f}}|\bar{\mathbf{e}},\bar{\mathbf{a}}) = \log \prod_{i}^{|\bar{\mathbf{e}}|} \frac{1}{|\{j|(i,j) \in \bar{\mathbf{a}}\}|} \sum_{\forall (i,j) \in \bar{\mathbf{a}}} t(e_i|f_j)$$

$$\log p_{lex}(\bar{\mathbf{e}}|\bar{\mathbf{f}},\bar{\mathbf{a}}) = \log \prod_{j}^{|\bar{\mathbf{f}}|} \frac{1}{|\{i|(j,i) \in \bar{\mathbf{a}}\}|} \sum_{\forall (j,i) \in \bar{\mathbf{a}}} t(f_j|e_i)$$

- Lexical weighing which scores by word translation probabilities
- Idea: counts for rare phrase pairs are unreliable
    - Smoothing effect by decomposing into word pairs

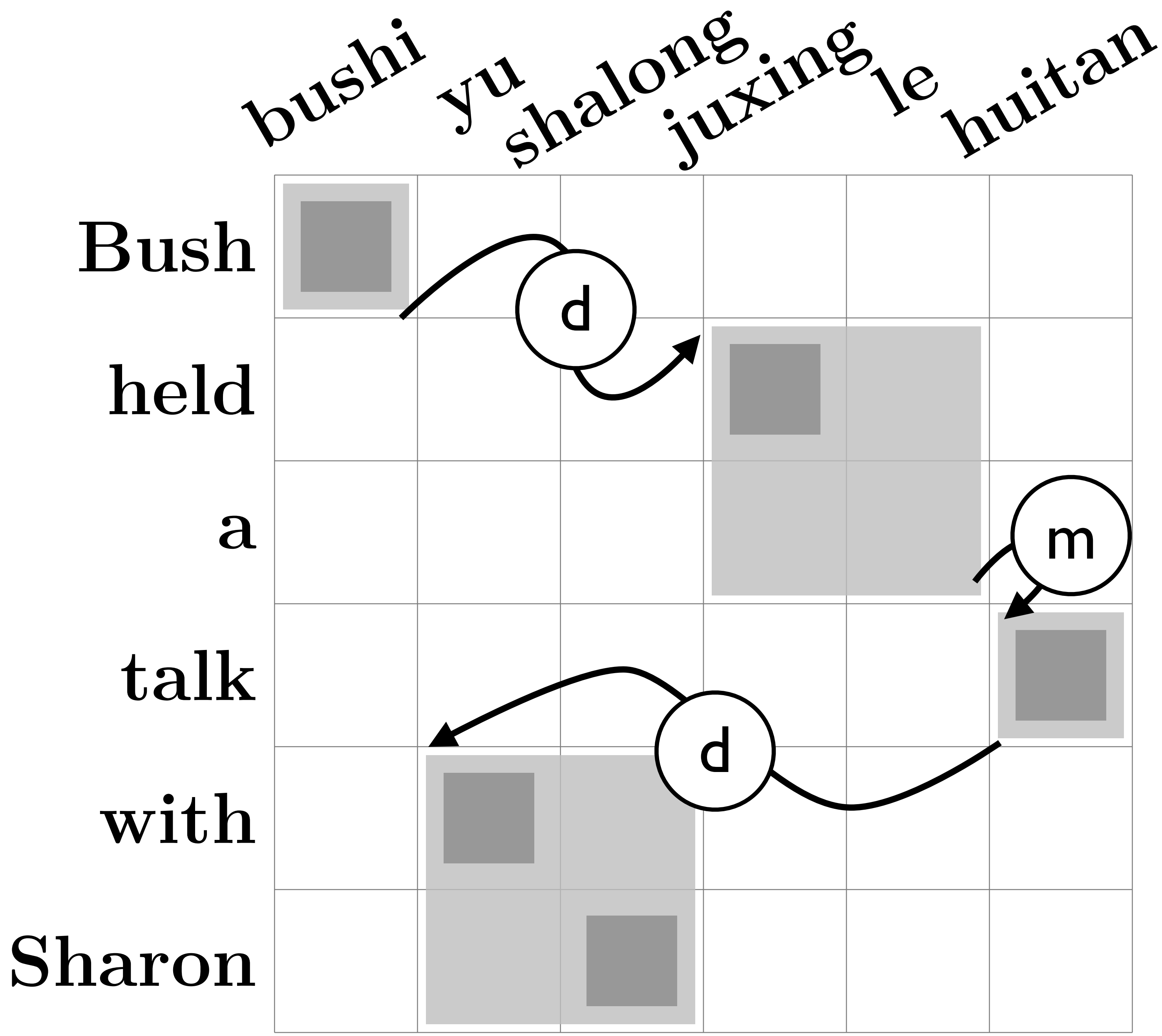
Features for Distortion



- Distance-based distortion modeling

$$d(\mathbf{f}, \phi, \mathbf{e}) = |+2| + |0| + |-5| = 7$$

Features for Reordering



- Fine grained reordering features: $\log p_o(o \in \{m, s, d\} \,|\bar{\mathbf{f}}, \bar{\mathbf{e}})$
- Either monotone, swap, discontinuous

Other Features

- log of ngram language model(s)
- word count: bias for ngram language model(s)
- phrase count: shorter or longer phrases

Direct Training

- Instead of word alignment + extraction pipeline, directly learn phrase-pairs (Marcu and Wong, 2002)
- Bayesian approach + blocked Gibbs sampling to learn parameters (Blunsom et al., 2009)
- Exhaustively memorize longer phrases (Neubig et al., 2011)

Questions

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}}\, \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})$$

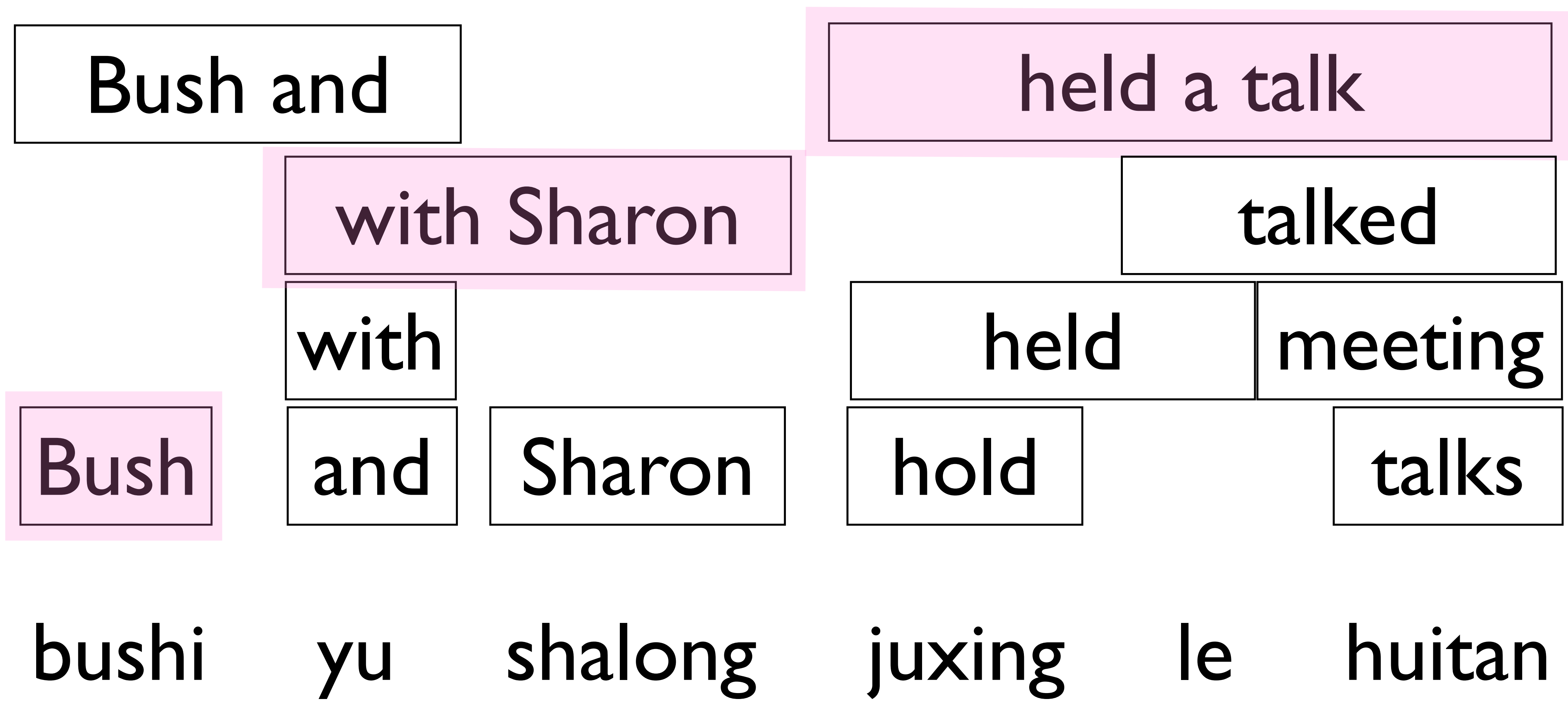
- Training: How to learn phrases and parameters (Φ and h)?
- Decoding (or search): How to find the best translation (argmax)?
- Tuning (or optimization): How to learn the scaling of features (w)?

Decoding

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} \frac{\exp\left(\mathbf{w}^{\top} \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\right)}{\sum_{\mathbf{e}', \phi'} \exp\left(\mathbf{w}^{\top} \cdot \mathbf{h}(\mathbf{e}', \phi', \mathbf{f})\right)} \\ &= \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^{\top} \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\end{aligned}$$

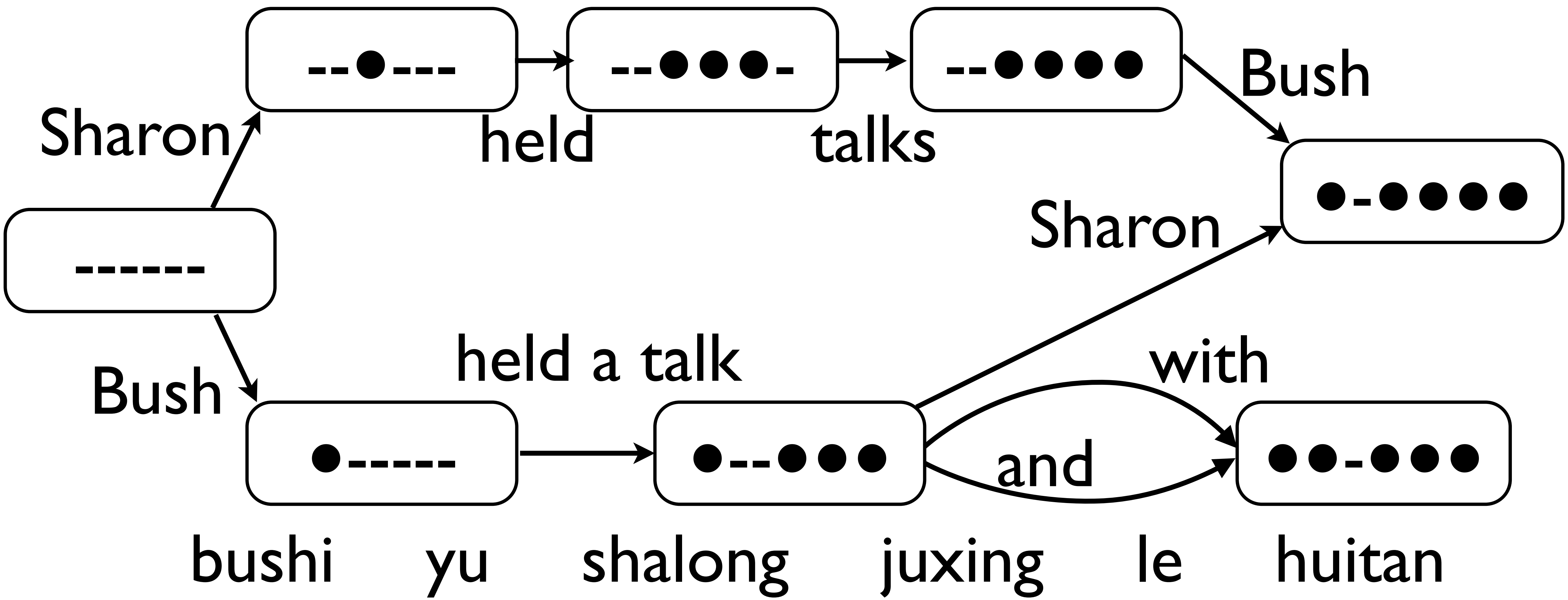
- Given an input sentence f and phrasal model h and w, seek e with the highest score
- Potential errors:
  - Search error: we cannot find the best scored hypothesis
  - Translation error: highest scored hypothesis is bad

Enumerate Phrase Pairs



- Given a input sentence f, we can enumerate all possible phrases that match with the source side
- Choose the best phrase pair + ordering

Phrase-based Search Space



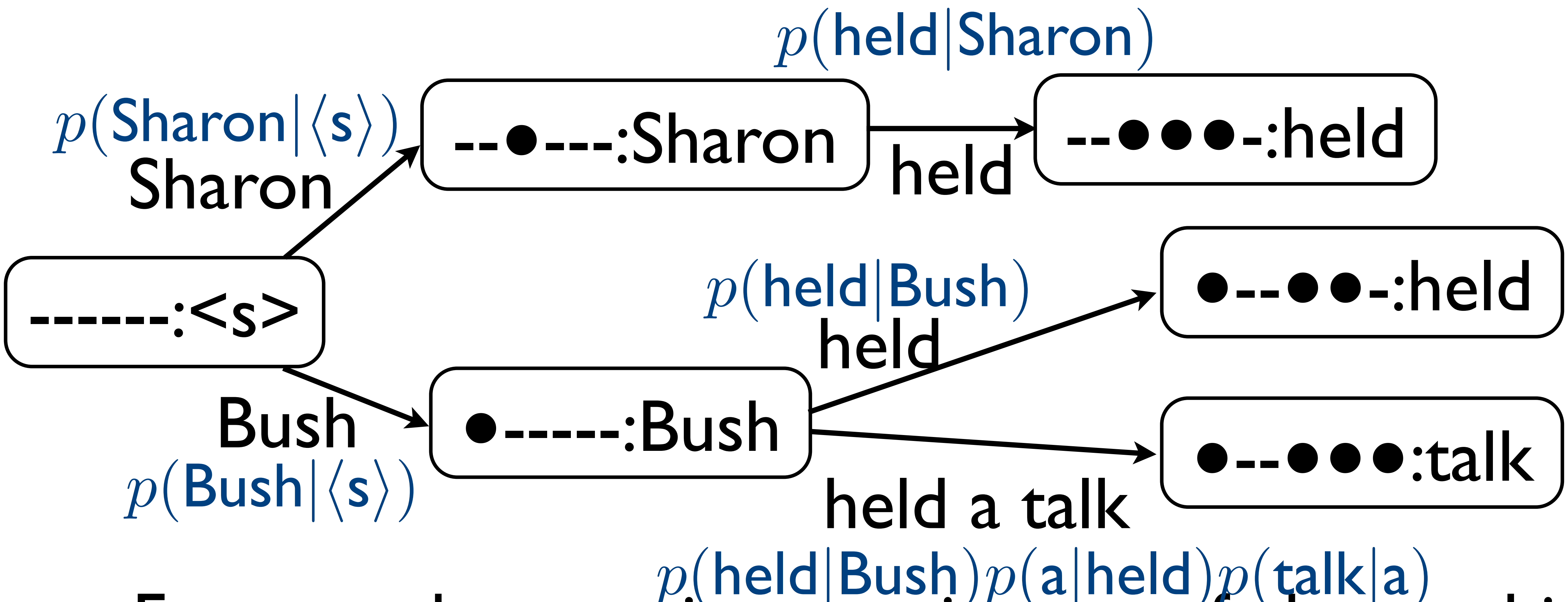
- Node: bit-vector representing covered source words
- Edge: phrasal translations, strictly left-to-right
- Search space: $O(2^n)$, Time: $O(2^n n^2)$ (Why?)

Traveling Salesman Problem

- NP-hard problem: visit each city only once
- MT as a Traveling Salesman Problem (Knight, 1999)
  - Each source word corresponds to a city
  - A Dynamic Programming solution:
    - State: visited cities (bit-vector)
    - Search space: $O(2^n)$
  - Distortion limit to reduce search space

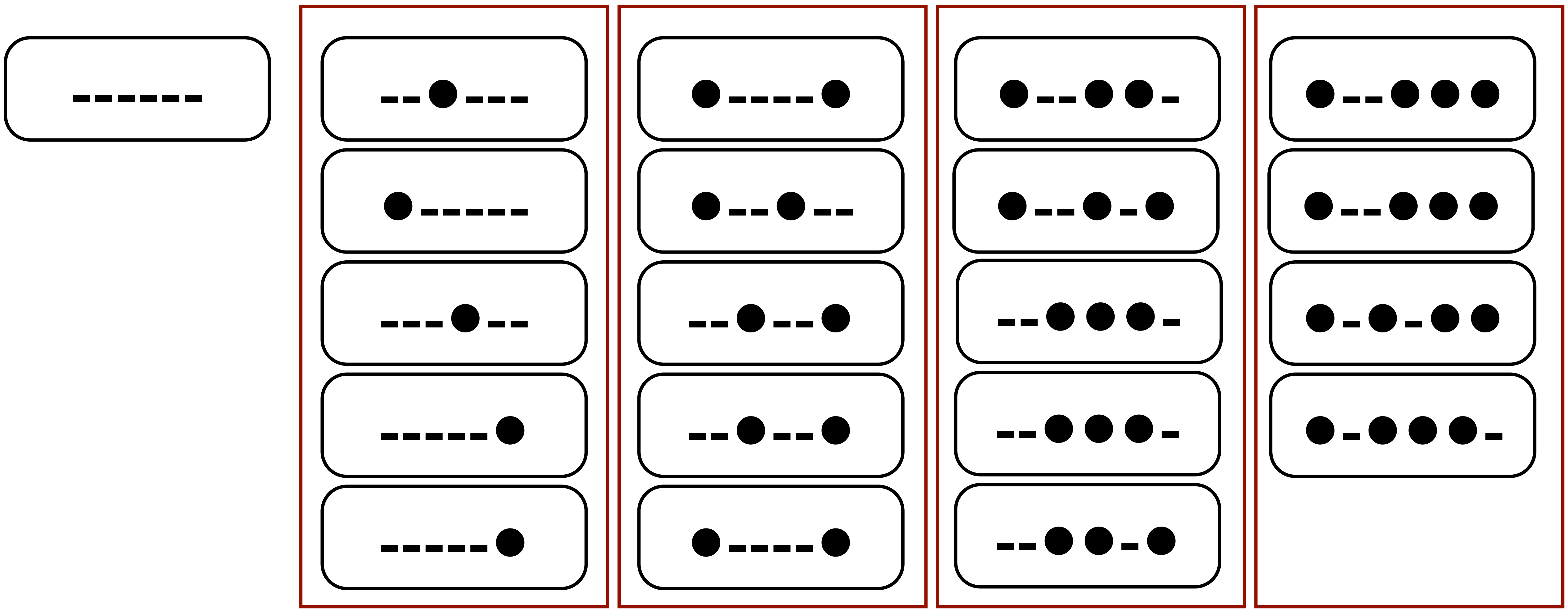
    i.e. long distortion: ●----- → ●----●

Non-local features



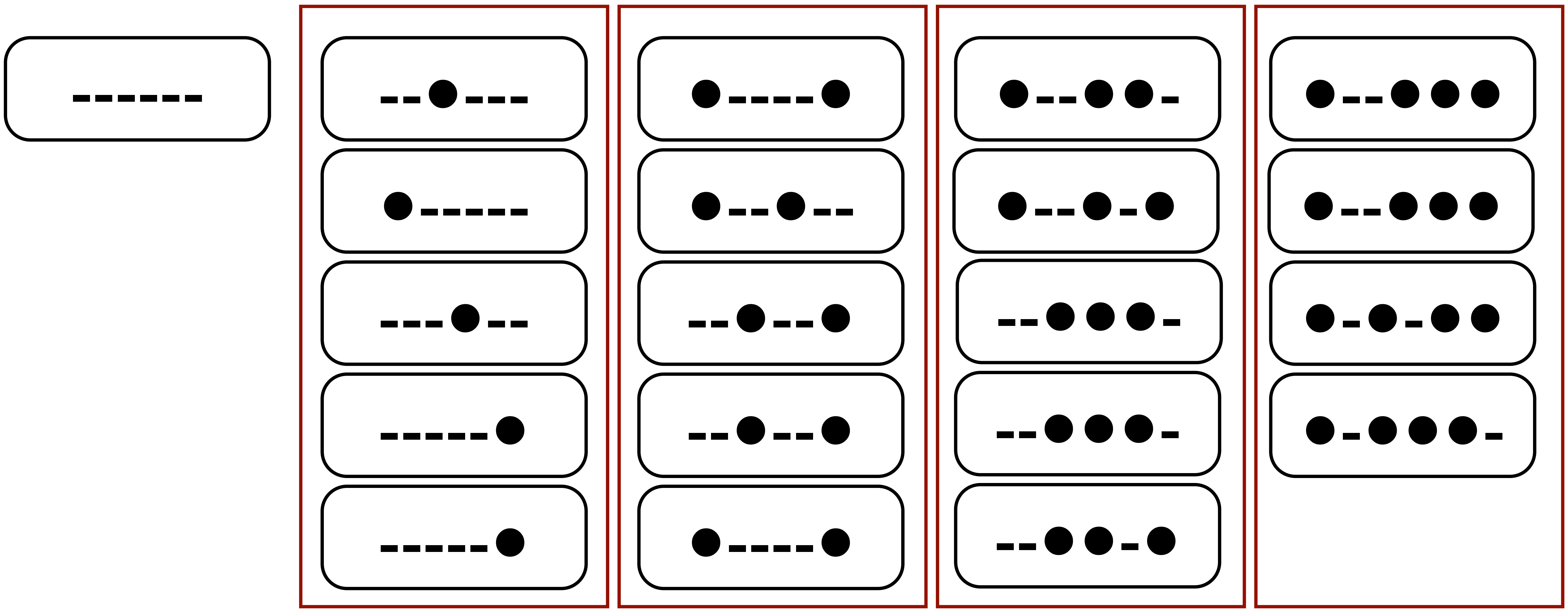
- Features that requires scoring out of phrases: bigram language model
- Additional state representation required for "future scoring": 1-word for bigram LM
- Space: $O(2^n V^{m-1})$, Time: $O(2^n V^{m-1} n^2)$ for m-gram LM

Phrase-based Decoding



- Re-organize the search space by the cardinality (= # of covered source words)
- Expand hypotheses from the smallest cardinality first

Pruning



- Prune hypotheses in a bin sharing the same cardinality
- Expand survived hypotheses only

Questions

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})$$

- Training: How to learn phrases and parameters (Φ and h)?
- Decoding (or search): How to find the best translation (argmax)?
- Tuning (or optimization): How to learn the scaling of features (w)?

Tuning

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} \frac{\exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\right)}{\sum_{\mathbf{e}', \phi'} \exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}', \phi', \mathbf{f})\right)} \\ &= \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \phi, \mathbf{f})\end{aligned}$$

- Three popular objectives (in SMT) for tuning w
  - (Direct) Error Minimization (Och, 2003)
  - Maximum Entropy (Och and Ney, 2002)
  - Large Margin (Watanabe et al., 2007; Chiang et al., 2008; Hopkins and May, 2011)

(Direct) Minimum Error

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \sum_{s=1}^{S} l(\operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \mathbf{f}_s), \mathbf{e}_s)$$

- MERT (Minimum ERror Training)
- Standard in SMT (but not in other NLP areas, such as tagging etc.)
  - We can incorporate arbitrary error functions, l
  - "Summation" can be replaced by document-wise BLEU specific summation
  - 10+ real valued features

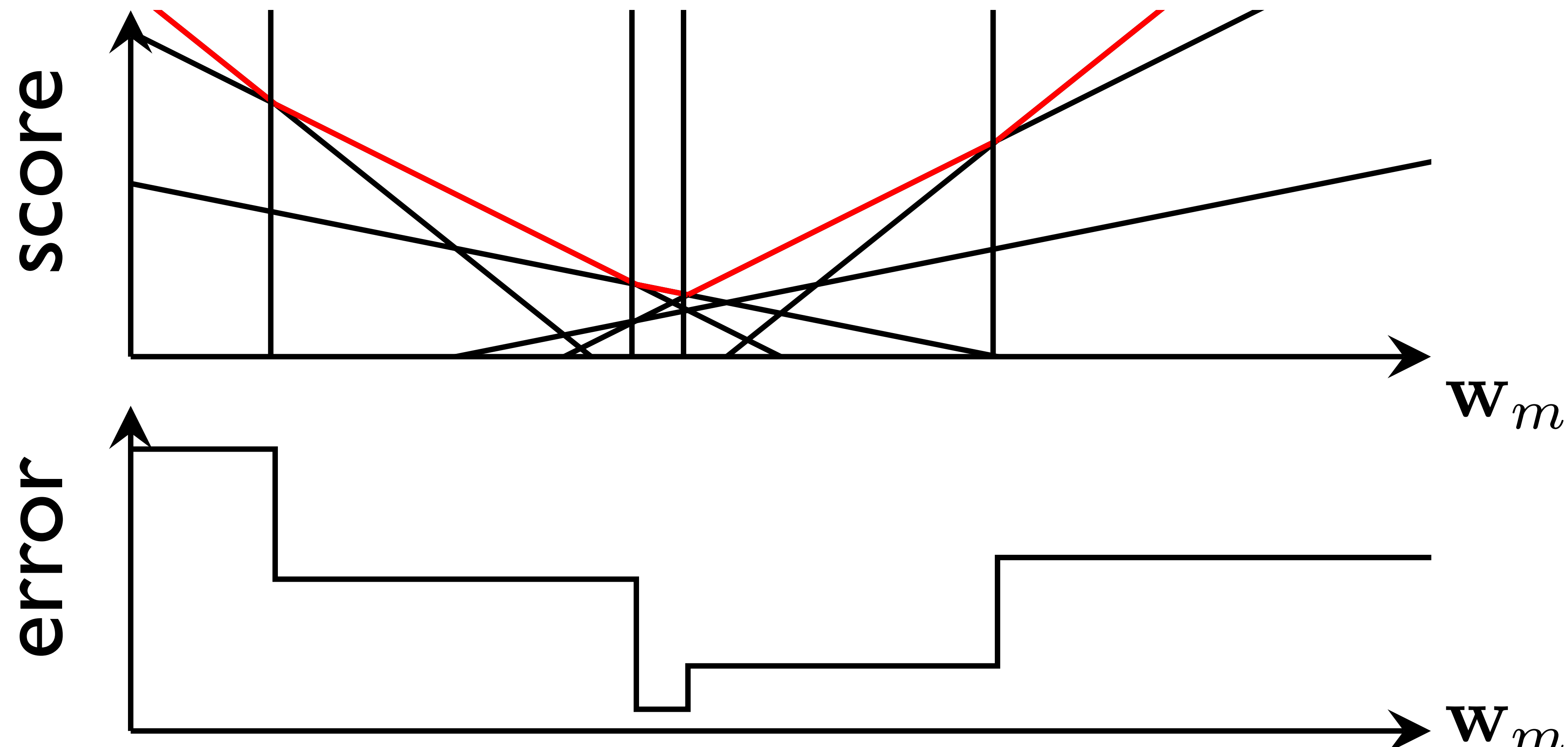
n-best Approximation

procedure MERT($\{(\mathbf{e}_s, \mathbf{f}_s)\}_{s=1}^{S}$)
    for $n = 1...N$ **do**
        Decode and generate nbest list using **w**
        Merge nbest list
        for $k = 1...K$ **do**
            for each parameter $m = 1...M$ **do**
                Solve one dimensional optimization
            end for
            update **w**
        end for
    end for
end procedure

- N iterations, with each iteration, n-bests are generated and merged
  - K iterations, with each iteration, M dimensions are tried (M = # of features), and w is updated

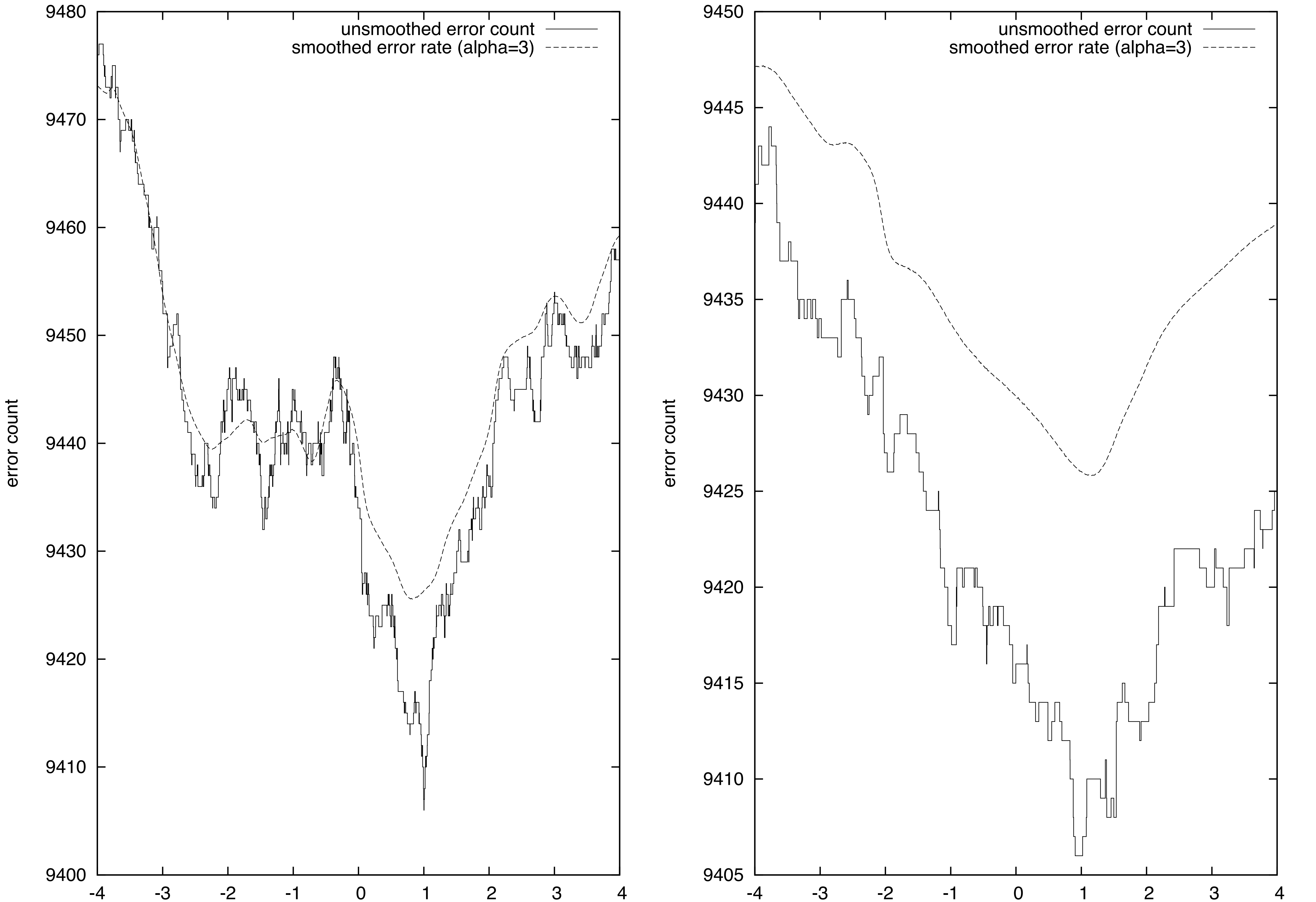
Efficient Line Search

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}_m^\top \cdot \underbrace{\mathbf{h}_m(\mathbf{e}, \mathbf{f}_s)}_{\text{slope}} + \underbrace{\mathbf{w}_{m\_}^\top \cdot \mathbf{h}_{m\_}(\mathbf{e}, \mathbf{f}_s)}_{\text{constant}}$$



- If we choose one dimension m, and others fixed, we can treat each hypothesis e as a "line"
- Compute convex hull of a set of "lines"

Error Surface



(Och, 2003)

MERT in Practice

- Many random starting points (Macherey et al., 2008; Moore and Quirk, 2008)
- Many random directions (Macherey et al., 2008)
- Error count smoothing (Cer et al., 2008)
- Regularization (Hayashi et al., 2009)
- Multi-dimensional search by efficiently computing convex hull (Galley and Quirk, 2011)
- MERT at least 3 times, and report average BLEU (Clark et al., 2011)

Maximum Entropy

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 - \sum_{s=1}^{S} \log \frac{\sum_{\mathbf{e}^* \in \mathsf{ORACLE}(\mathbf{f}_s)} \exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}^*, \mathbf{f}_s)\right)}{\sum_{\mathbf{e}' \in \mathsf{GEN}(\mathbf{f}_s)} \exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}', \mathbf{f}_s)\right)}$$

- Minimize the negative log-likelihood of generating good translations (Och and Ney, 2002)
- ORACLE is a subset of GEN, a set of hypotheses with minimum loss
- Optimized by L-BFGS or SGD
- Potentially large # of features as in NLP tasks

Why Not MaxEnt?

error criterion used in training	mWER [%]	mPER [%]	BLEU [%]	NIST	# words
confidence intervals	+/- 2.7	+/- 1.9	+/- 0.8	+/- 0.12	-
MMI	**68.0**	*51.0*	11.3	5.76	21933
mWER	*68.3*	*50.2*	13.5	6.28	22914
smoothed-mWER	*68.2*	*50.2*	13.2	6.27	22902
mPER	*70.2*	*49.8*	15.2	*6.71*	24399
smoothed-mPER	*70.0*	**49.7**	15.2	*6.69*	24198
BLEU	76.1	53.2	**17.2**	6.66	28002
NIST	73.3	*51.5*	*16.4*	**6.80**	26602

- In Och and Ney (2002), they used
  - WER to select oracle translations
  - n-best merging approach to approximate summation as in MERT

Large Margin

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{s=1}^{S} \sum_{\mathbf{e}_s^*} \sum_{\mathbf{e}_s'} \xi_{s,\mathbf{e}_s^*,\mathbf{e}_s'}$$

$$\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}_s^*, \mathbf{f}_s) - \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}_s', \mathbf{f}_s) \geq l(\mathbf{e}_s', \mathbf{e}_s^*) - \xi_{s,\mathbf{e}_s^*,\mathbf{e}_s'}$$

$$\mathbf{e}_s^* \in \mathsf{ORACLE}(\mathbf{f}_s)$$

$$\mathbf{e}_s' \in \mathsf{GEN}(\mathbf{f}_s)$$

- Structured output learning approach
- Very hard to enumerate all possible e’ and oracle translations e*
- Solution: online learning or n-best approximation

Online Learning

Require: $\{(\mathbf{f}_s, \mathbf{e}_s)\}_{s=1}^{S}$

$\mathbf{w}^1 = \{0\}$

$t = 1$

for $1 \ldots N$ **do**

  $s \sim \mathrm{random}(1, S)$

  $\hat{\mathbf{e}} \in \mathsf{GEN}(\mathbf{f}_s, \mathbf{w}^{t-1})$

  if $l(\hat{\mathbf{e}}, \mathbf{e}_s) \geq 0$ **then**

    $\mathbf{w}^{t+1} = \mathbf{w}^t + \mathbf{h}(\mathbf{e}_s, \mathbf{f}_s) - \mathbf{h}(\hat{\mathbf{e}}, \mathbf{f}_s)$

    $t = t + 1$

  end if

end for

return $\mathbf{w}^t$ or $\frac{1}{N} \sum_{i=1}^{N} \mathbf{w}^j$

- Averaged perceptron (Liang et al., 2006)
- Scale to large data, but each iteration requires decoding + weight update

Online Large Margin

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}'} \frac{\lambda}{2} ||\mathbf{w}' - \mathbf{w}||^2 + \max\left(l_s - \mathbf{w}'^\top \cdot \Delta\mathbf{h}_s\right)$$

$$\begin{aligned}
\hat{\mathbf{e}}_s &= \operatorname*{argmax}_{e} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, \mathbf{f}_s) \\
l_s &= l(\hat{\mathbf{e}}_s) - l(\mathbf{e}_s^*) \\
\Delta\mathbf{h}_s &= \mathbf{h}(\hat{\mathbf{e}}_s, \mathbf{f}_s) - \mathbf{h}(\mathbf{e}^*, \mathbf{f}_s)
\end{aligned}$$

- line 7 is replaced by the solution of the above equation
- Still, requires decoding + update in each iteration
- Hard to determine when to stop (watch another dev data)

Ranking Approach

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{s=1}^{S} \sum_{\mathbf{e}''_s} \sum_{\mathbf{e}'_s} \xi_{s,\mathbf{e}''_s,\mathbf{e}'_s}$$

$$-\log\left(1 + \exp(-\mathbf{w}^\top \cdot \Delta\mathbf{h}_{\mathbf{e}''_s,\mathbf{e}'_s})\right) \geq -\xi_{s,\mathbf{e}''_s,\mathbf{e}'_s}$$

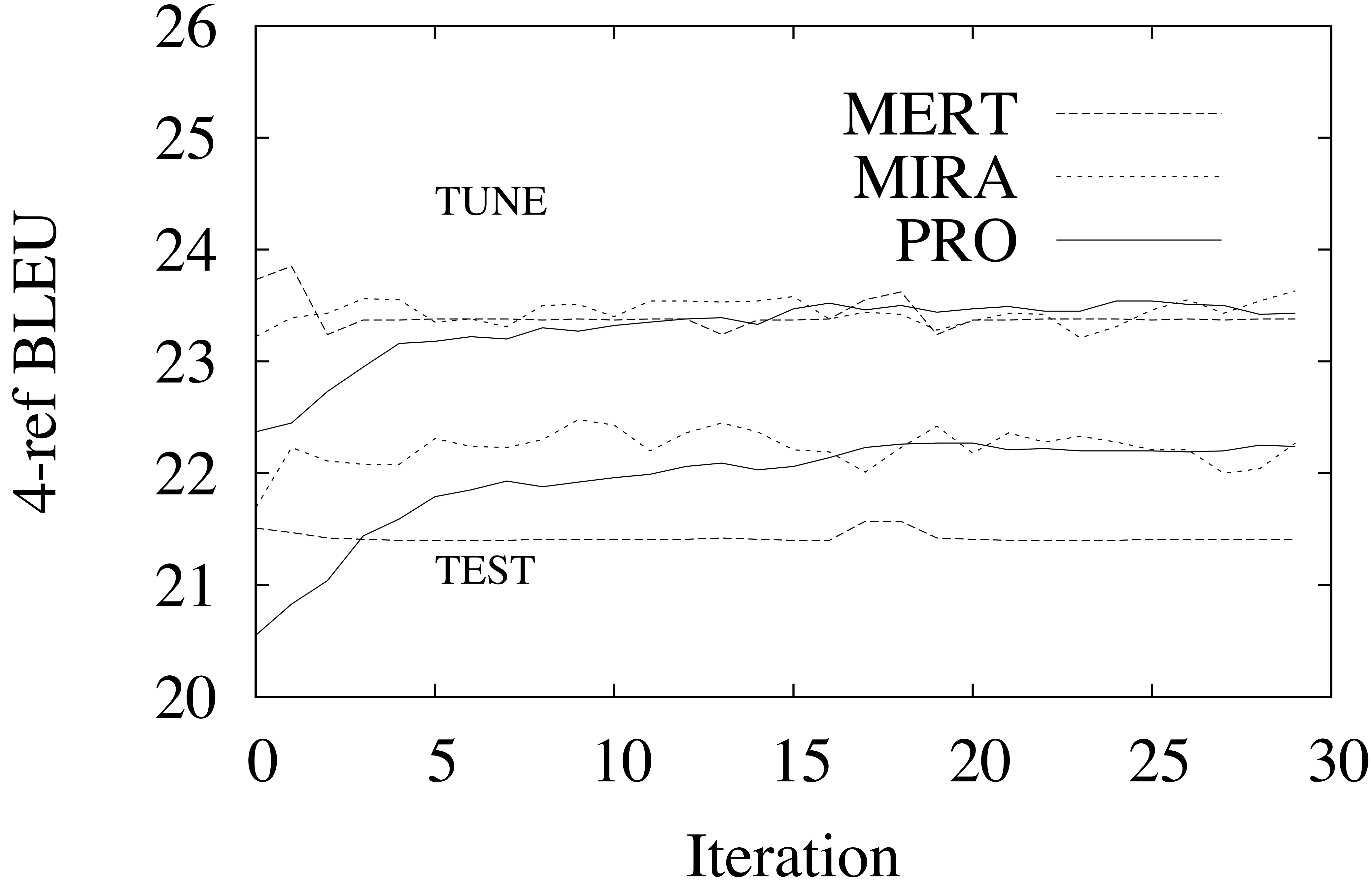
$$\mathbf{e}''_s, \mathbf{e}'_s \in \mathsf{GEN}(\mathbf{f}_s)$$

$$l(\mathbf{e}'_s, \mathbf{e}''_s) > 0$$

$$\Delta\mathbf{h}_{\mathbf{e}''_s,\mathbf{e}'_s} = \mathbf{h}(\mathbf{e}''_s, \mathbf{f}_s) - \mathbf{h}(\mathbf{e}'_s, \mathbf{f}_s)$$

- An n-best approximation approach (Hopkins and May, 2011)
- Pair-wise comparison of all the hypotheses
- logistic-loss (or 0-1 loss): use an off-the-shelf binary classifier

Results



- Reranking is competitive to MERT and MIRA, and scales to large # of features

Answered?

- Grammar-less model (but very strong)
- Fast decoding
- Why MERT? (Good for non-binary, numerical features)

Structures in SMT

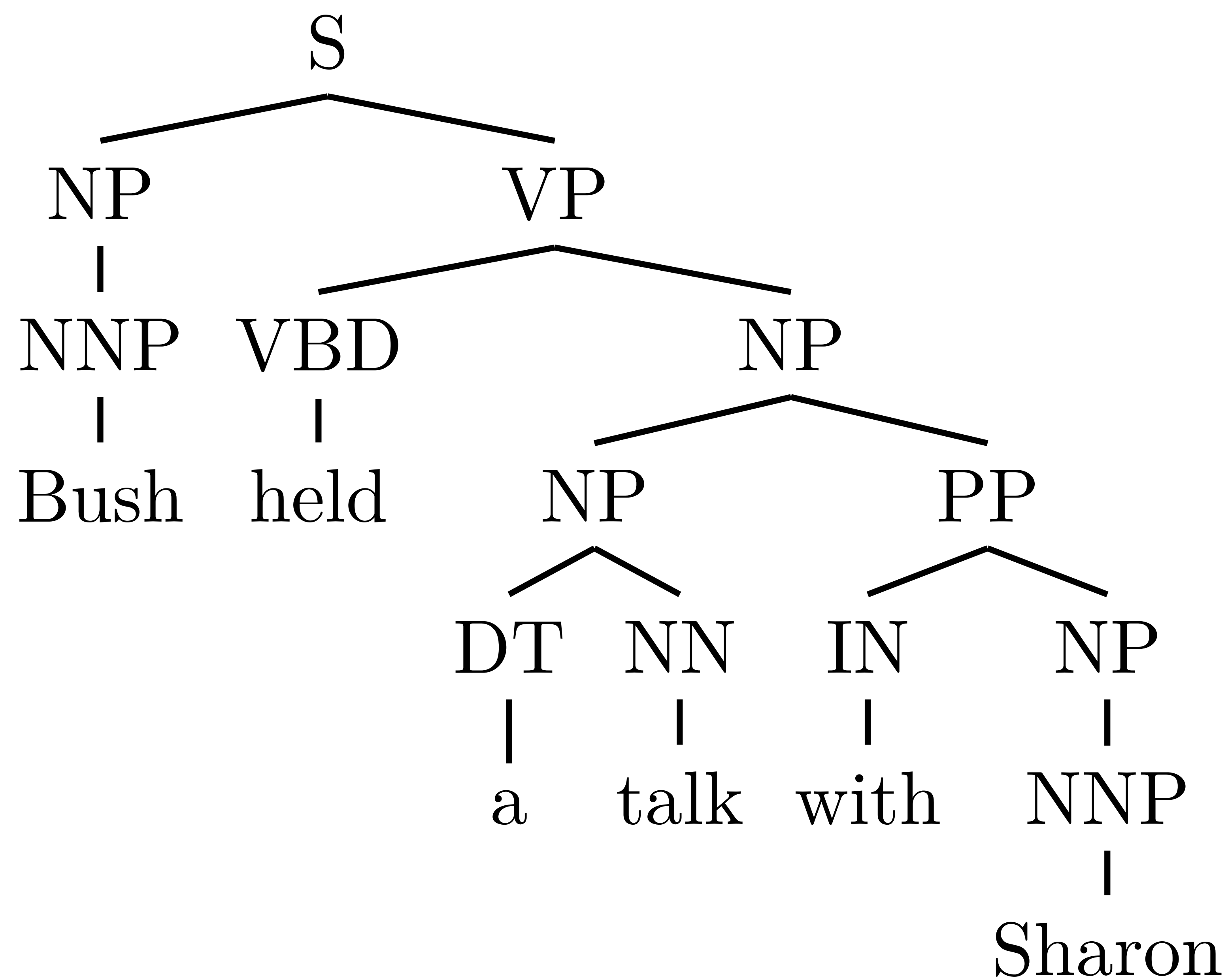
- Tutorial
  - Phrase-based MT
  - Tree-based MT
- Syntactic Structures in System Combination

Tree-based MT

- Backgrounds
  - CFG, parsing, hypergraph, deductive system semirings
- Tree-based SMT
  - Synchronous-CFG
  - String-to-Tree, Tree-to-String

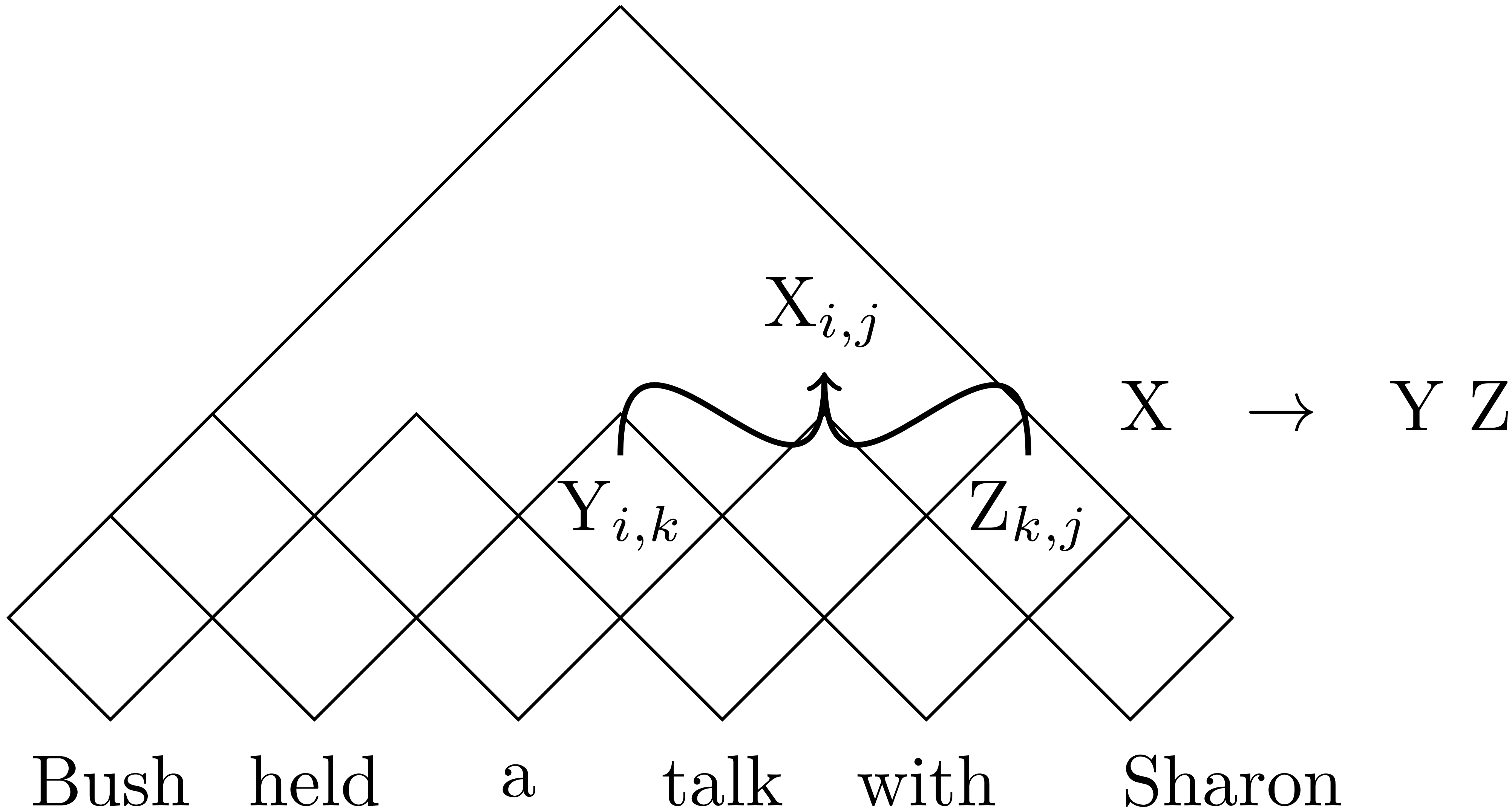
Backgrounds: CFG

$$
\begin{array}{rcl}
\mathrm{S} & \rightarrow & \mathrm{NP\ VP} \\
\mathrm{NP} & \rightarrow & \mathrm{NNP} \\
\mathrm{NP} & \rightarrow & \mathrm{NP\ PP} \\
\mathrm{NP} & \rightarrow & \mathrm{DP\ NN} \\
\mathrm{NP} & \rightarrow & \mathrm{DT\ NN} \\
\mathrm{VP} & \rightarrow & \mathrm{VBD\ NP} \\
\mathrm{NNP} & \rightarrow & \mathrm{Bush} \\
\mathrm{VBD} & \rightarrow & \mathrm{held} \\
 & \vdots &
\end{array}
$$



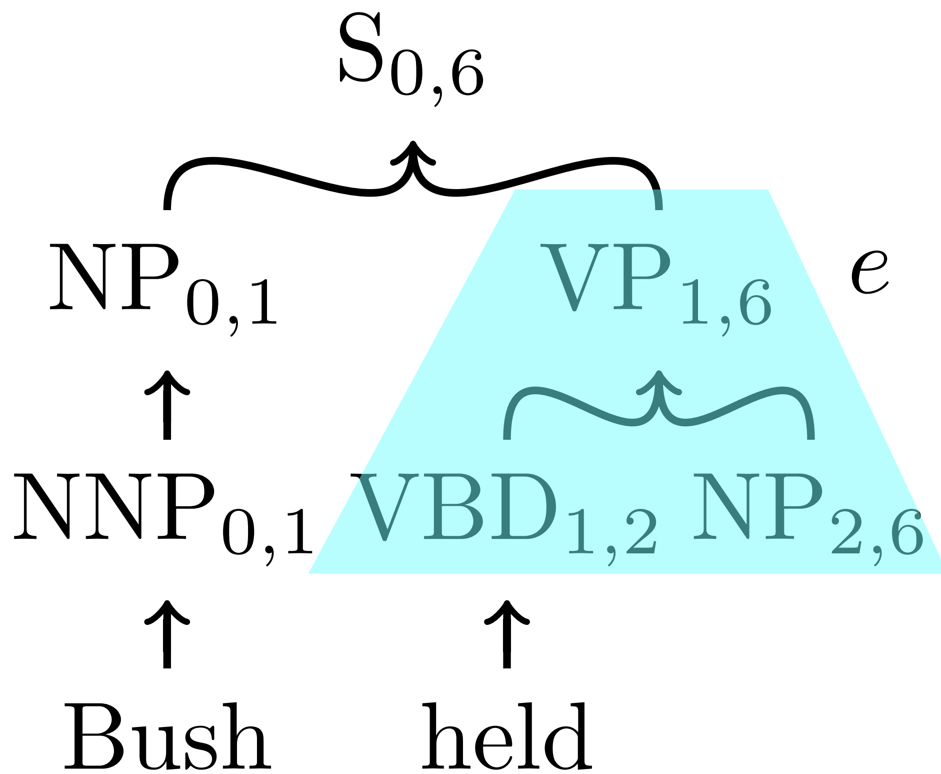
- parsing = intersection of CFG with a string (regular grammar)

Parsing: CKY

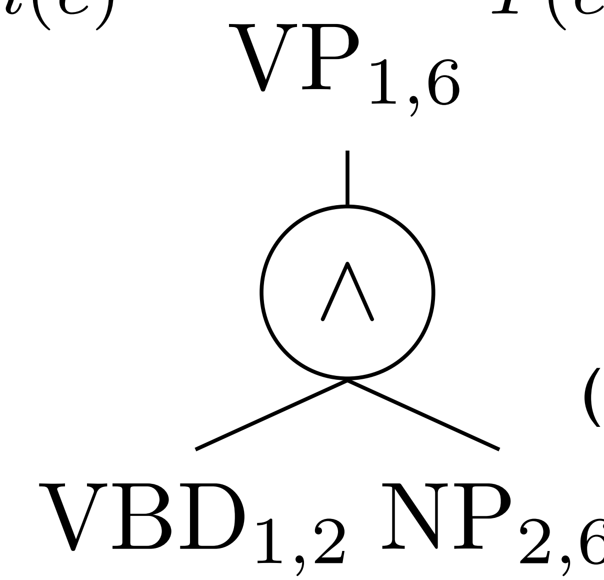


- $O(n^3)$: For each length n, for each position i, for each rule X → Y Z, for each split point k
- (Bottom-up) topological order

Hypergraph



$$e = \langle \underbrace{VP_{1,6}}_{h(e)}, \underbrace{\{VBD_{1,2}, NP_{2,6}\}}_{T(e)} \rangle$$



(Klein and Manning, 2001)

- Generalization of graphs:
  - h(e): head node of hyperedge e
  - T(e): tail node(s) of hyperedge e, arity = |T(e)|
  - hyperedge = instantiated rule
- Represented as and-or graphs

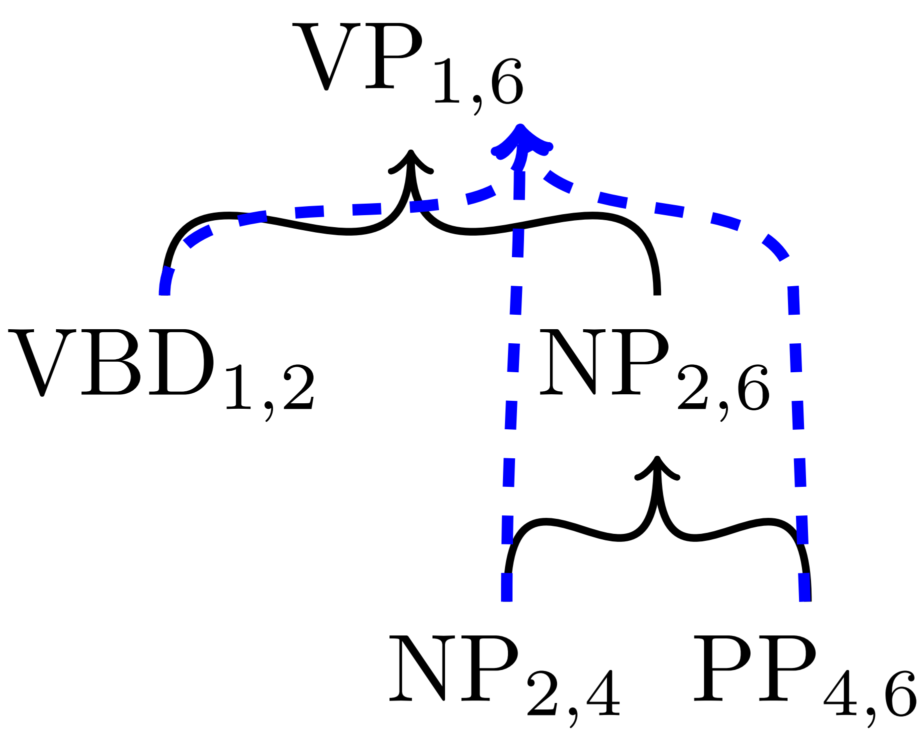
Deductive System

$$\underbrace{\mathrm{VBD}_{1,2} \quad \mathrm{NP}_{2,6}}_{\mathrm{VP}_{1,6}} \qquad \frac{\overbrace{\mathrm{VBD}_{1,2}\ \mathrm{NP}_{2,6}}^{\text{antecedents}}}{\underbrace{\mathrm{VP}_{1,6}}_{\text{consequent}}} \mathrm{VP}_{[i,j]} \rightarrow \mathrm{VBZ}_{[j,k]}\ \mathrm{NP}_{[i,k]}$$

(Shieber et al., 1995)

- Parsing algorithm as a deductive system
- We start from initial items (axioms) until we reach a goal item
- If antecedents are proved, its consequent is proved
- deduction = hyperedge

Packed Forest



$$\frac{\mathrm{VBD}_{1,2} \quad \dfrac{\mathrm{NP}_{2,4}\ \mathrm{PP}_{4,6}}{\mathrm{NP}_{2,6}}}{\mathrm{VP}_{1,6}}$$

$$\frac{\mathrm{VBD}_{1,2}\ \mathrm{NP}_{2,4}\ \mathrm{PP}_{4,6}}{\mathrm{VP}_{1,6}}$$

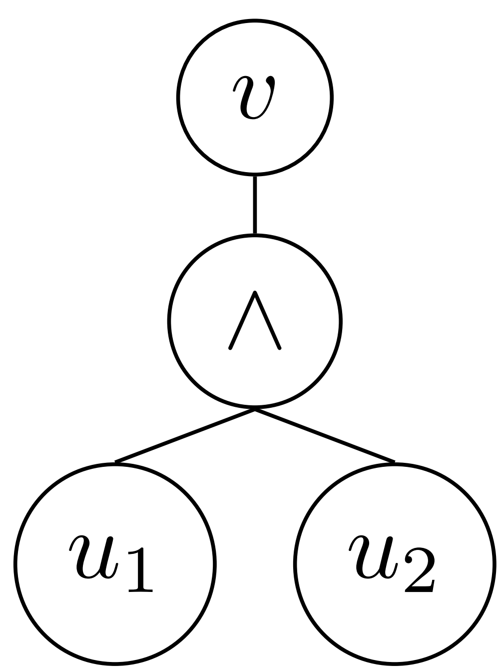
(Klein and Manning, 2001; Huang and Chiang, 2005)

- A polynomial space encoding of exponentially many parses by sharing common sub-derivations
- Single derivation = tree

Summary of Formalisms

hypergraph	AND/OR graph	CFG	deductive system
vertex source-vertex target-vertex	OR-node leaf OR-node root OR-node	symbol terminal start symbol	item axiom goal item
hyperedge	AND-node	production	instantiated deduction

$\langle v, \{u_1, u_2\} \rangle$



$v \rightarrow u_1 \ u_2$

$$\frac{u_1 \ u_2}{v}$$

Weights and Semirings

$$\mathrm{VP} \xrightarrow{w_1} \mathrm{VBD}\ \mathrm{NP}$$

$$\mathrm{NP} \xrightarrow{w_2} \mathrm{NP}\ \mathrm{PP}$$

$$\overbrace{\mathrm{VBD}_{1,2} : c \quad \mathrm{NP}_{2,6} : d}^{\mathrm{VP}_{1,6} : w_1 \otimes c \otimes d}$$

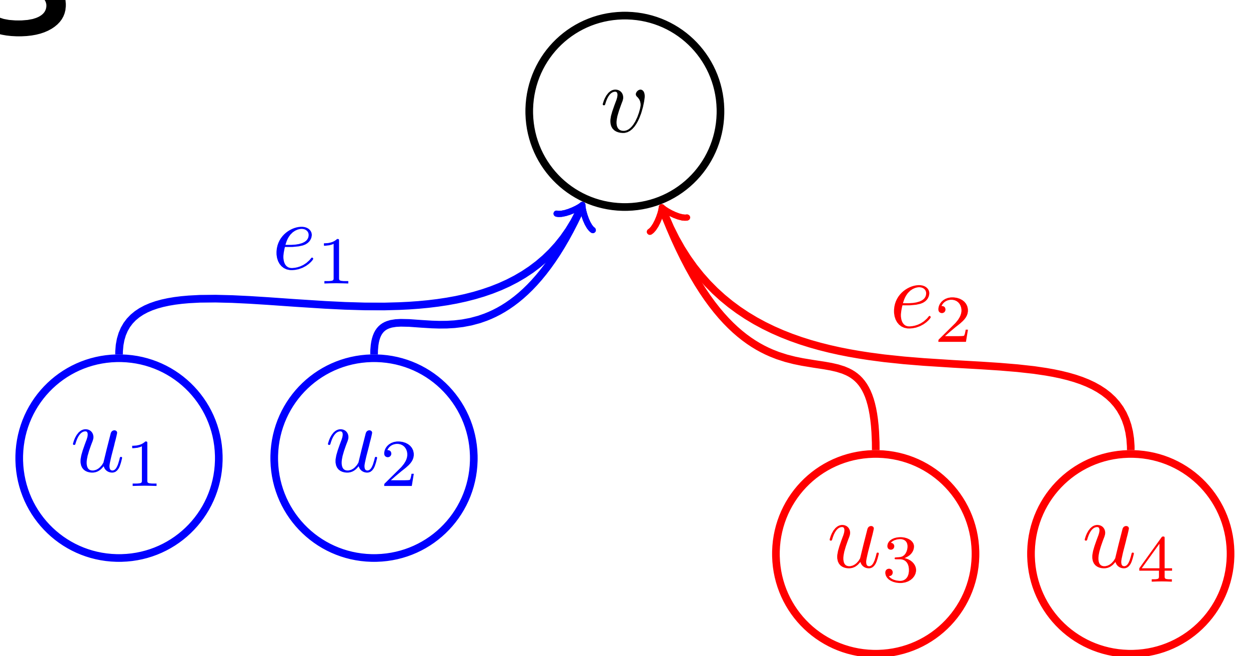
$$\overbrace{\mathrm{NP}_{2,4} : a \quad \mathrm{PP}_{4,6} : b}^{\mathrm{NP}_{2,6} : w_2 \otimes a \otimes b}$$

$$\frac{\mathrm{VBD}_{1,2} : c \ \ \mathrm{NP}_{2,6} : d}{\mathrm{VP}_{1,6} : w_1 \otimes c \otimes d} : w_1$$

$$\frac{\mathrm{NP}_{2,4} : a \ \ \mathrm{PP}_{4,6} : b}{\mathrm{NP}_{2,6} : w_2 \otimes a \otimes b} : w_2$$

- Associate weights as in WFST
- $\otimes$: extension (multiplicative), $\oplus$: summary (additive)

Weights and Semirings



$$
\begin{aligned}
d(v) \quad = \quad & (w(e_1, u_1, u_2) \otimes d(u_1) \otimes d(u_2)) \\
& \oplus (w(e_2, u_3, u_4) \otimes d(u_3) \otimes d(u_4))
\end{aligned}
$$

- The weight of a hyperedge is dependent on antecedents (non-monotonic)
- The weight of a derivation is the product of hyperedge weights
- The weight of a vertex is the summary of (sub-)derivation weights

Semirings

$$\mathbf{K} = \langle K, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$$

semiring	K	$\oplus$	$\otimes$	0	1
Viterbi	[0,1]	max	$\times$	0	1
Real	R	+	x	0	1
Log	R	logsumexp	+	$+\infty$	0
Tropical	R	min	+	$+\infty$	0
Expectation	$\langle P,R \rangle$	$\langle p_1 \oplus p_2, r_1 \oplus r_2 \rangle$	$\langle p_1 \otimes p_2, p_1 \otimes r_2 \oplus p_2 \otimes r_1 \rangle$	$\langle 0,0 \rangle$	$\langle 1,0 \rangle$

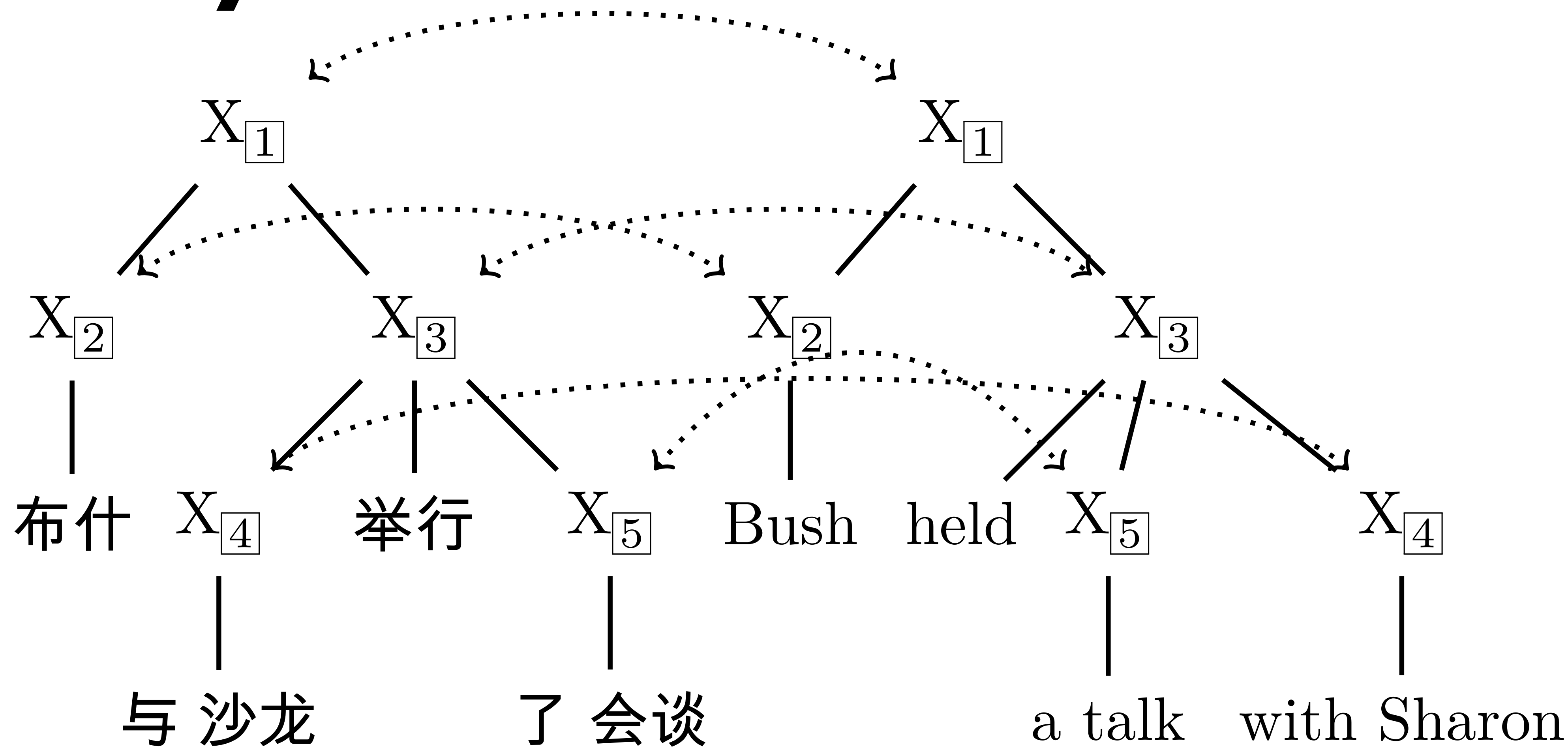
Conclusion

- Review important concepts from “parsing”
  - CFG, parsing, hypergraph, deductive system, weights, semirings

Tree-based MT

- Backgrounds
  - CFG, parsing, hypergraph, deductive system semirings
- Tree-based SMT
  - Synchronous-CFG
  - String-to-Tree, Tree-to-String

Synchronous-CFG



(Chiang, 2007)

$$\begin{aligned}\hat{\mathbf{e}} &= \operatorname*{argmax}_{\mathbf{e}} \frac{\exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, D, \mathbf{f})\right)}{\sum_{\mathbf{e}', D'} \exp\left(\mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}', D', \mathbf{f})\right)} \\ &= \operatorname*{argmax}_{\mathbf{e}} \mathbf{w}^\top \cdot \mathbf{h}(\mathbf{e}, D, \mathbf{f})\end{aligned}$$

- D: a single derivation constructed by intersecting SCFG with input string

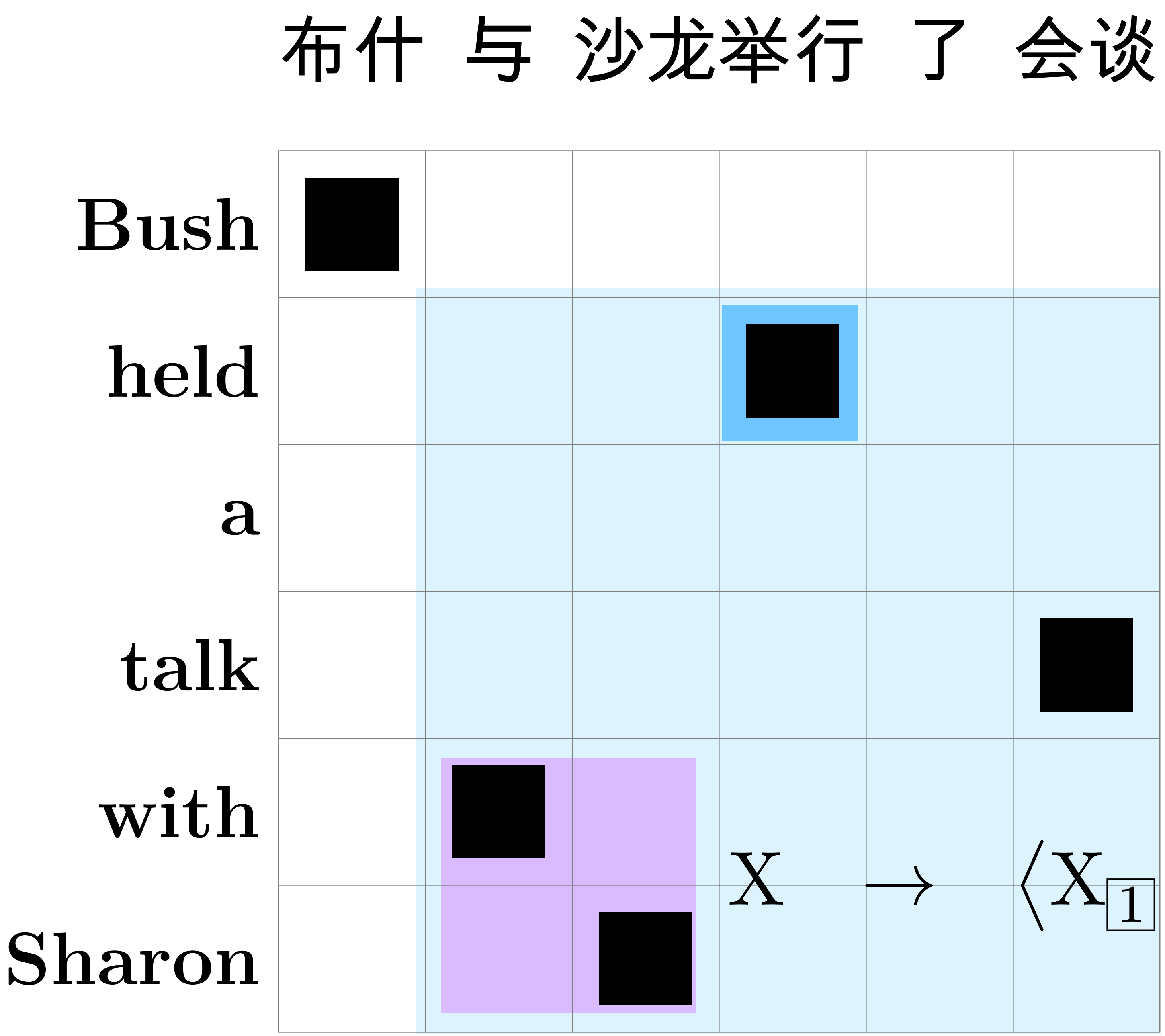
Synchronous-CFG: Model

$$\begin{aligned}
S &\rightarrow \langle S_{\boxed{1}}\ X_{\boxed{2}}, S_{\boxed{1}}\ X_{\boxed{2}} \rangle \\
S &\rightarrow \langle X_{\boxed{1}}, X_{\boxed{1}} \rangle \\
X &\rightarrow \langle X_{\boxed{1}}\ \text{举行}\ X_{\boxed{2}}, \text{hold}\ X_{\boxed{2}}\ X_{\boxed{1}} \rangle \\
X &\rightarrow \langle \text{与 沙龙}, \text{with Sharon} \rangle
\end{aligned}$$

$$\begin{aligned}
\text{VP} &\rightarrow \langle \text{VBD}_{\boxed{1}}\ \text{NP}_{\boxed{2}},\ \text{NP}_{\boxed{2}}\ \text{VBD}_{\boxed{1}} \rangle \\
\text{NP} &\rightarrow \langle \text{NP}_{\boxed{1}}\ \text{PP}_{\boxed{2}},\ \text{NP}_{\boxed{1}}\ \text{PP}_{\boxed{2}} \rangle \\
\text{VP} &\rightarrow \langle \text{VBD}_{\boxed{1}}\ \text{NP}_{\boxed{2}}\ \text{PP}_{\boxed{3}},\ \text{NP}_{\boxed{2}}\ \text{PP}_{\boxed{3}}\ \text{VBD}_{\boxed{1}} \rangle
\end{aligned}$$

- We use two categories, S and X (Chiang, 2007)
- Or, borrow linguistic categories from syntactic parse (Zollman and Venugopal, 2006)

Rule Extraction



$\langle$held a talk with Sharon, 与 沙龙 举行 了 会谈$\rangle$

$\langle$with Sharon, 与 沙龙$\rangle$

$\langle$held, 举行$\rangle$

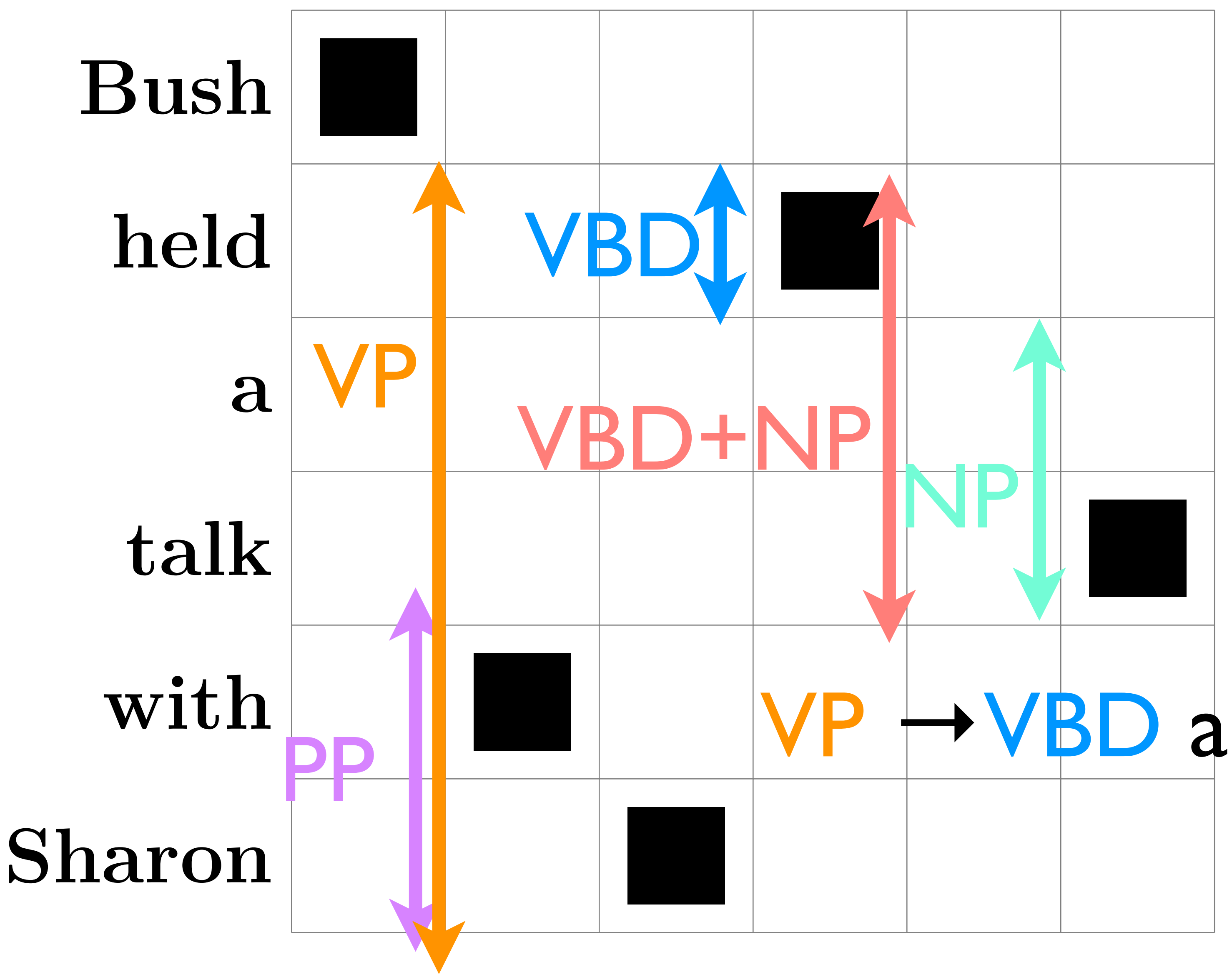
$X \rightarrow \langle X_{\boxed{1}}\ X_{\boxed{2}}$ 了 会谈, $X_{\boxed{2}}$ a talk $X_{\boxed{1}} \rangle$

(Example from Huang and Chiang, 2007)

- As in phrase-based models, extract phrases then, use sub-phrases as non-terminals, aka Hiero (Chiang, 2007)

Syntactic Categories

布什 与 沙龙举行 了 会谈



⟨held a talk with Sharon, 与 沙龙 举行 了 会谈⟩

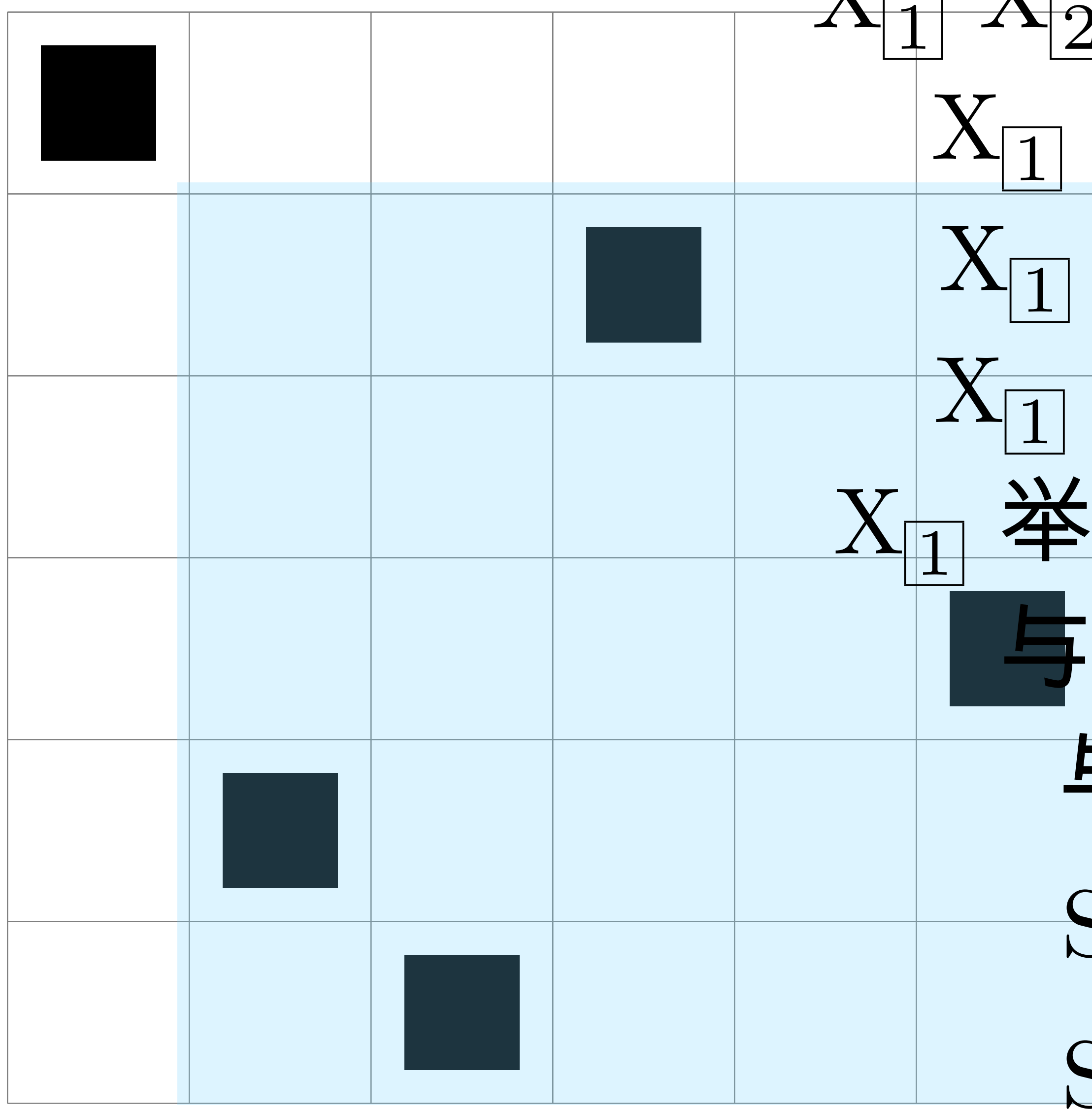
⟨with Sharon, 与 沙龙⟩

⟨held, 举行⟩

VP → VBD a talk PP, PP VBD 了 会谈

- Borrow syntactic categories either from source/ target side, aka SAMT(Zollman and Venugopal, 2006)

Exhaustive Extraction



$X_{\boxed{1}}\ X_{\boxed{2}}$ 了 会谈 $\quad X_{\boxed{2}}$ a talk $X_{\boxed{1}}$

$X_{\boxed{1}}\ X_{\boxed{2}}$ 会谈 $\quad X_{\boxed{2}}$ a talk $X_{\boxed{1}}$

$X_{\boxed{1}}\ X_{\boxed{2}}$ 会谈 $\quad X_{\boxed{2}}$ talk $X_{\boxed{1}}$

$X_{\boxed{1}}$ 举行 $X_{\boxed{2}} \quad$ held $X_{\boxed{2}}\ X_{\boxed{1}}$

$X_{\boxed{1}}$ 举行 了 $X_{\boxed{2}} \quad$ held a $X_{\boxed{2}}\ X_{\boxed{1}}$

与 沙龙 $X_{\boxed{1}} \quad X_{\boxed{1}}$ with Sharon

与 $X_{\boxed{1}}\ X_{\boxed{2}} \quad X_{\boxed{2}}$ with $X_{\boxed{1}}$

$$S \rightarrow \langle S_{\boxed{1}}\ X_{\boxed{2}}, S_{\boxed{1}}\ X_{\boxed{2}} \rangle$$

$$S \rightarrow \langle X_{\boxed{1}}, X_{\boxed{1}} \rangle$$

- Exhaustively extract rules as in phrase-based MT
- + glue rules

Features from Rules

$$
\begin{aligned}
\log p_r(\bar{\alpha}|\bar{\beta}) &= \log \frac{\mathrm{count}(\bar{\beta}, \bar{\alpha})}{\sum_{\bar{\alpha}'} \mathrm{count}(\bar{\beta}, \bar{\alpha}')} \\
\log p_r(\bar{\beta}|\bar{\alpha}) &= \log \frac{\mathrm{count}(\bar{\beta}, \bar{\alpha})}{\sum_{\bar{\beta}'} \mathrm{count}(\bar{\beta}', \bar{\alpha})}
\end{aligned}
$$

- Collect all the rules (α, β) from the data:
  - α = source side string, β = target side string
- Maximum likelihood estimates by relative frequencies
- Employ scores in two directions

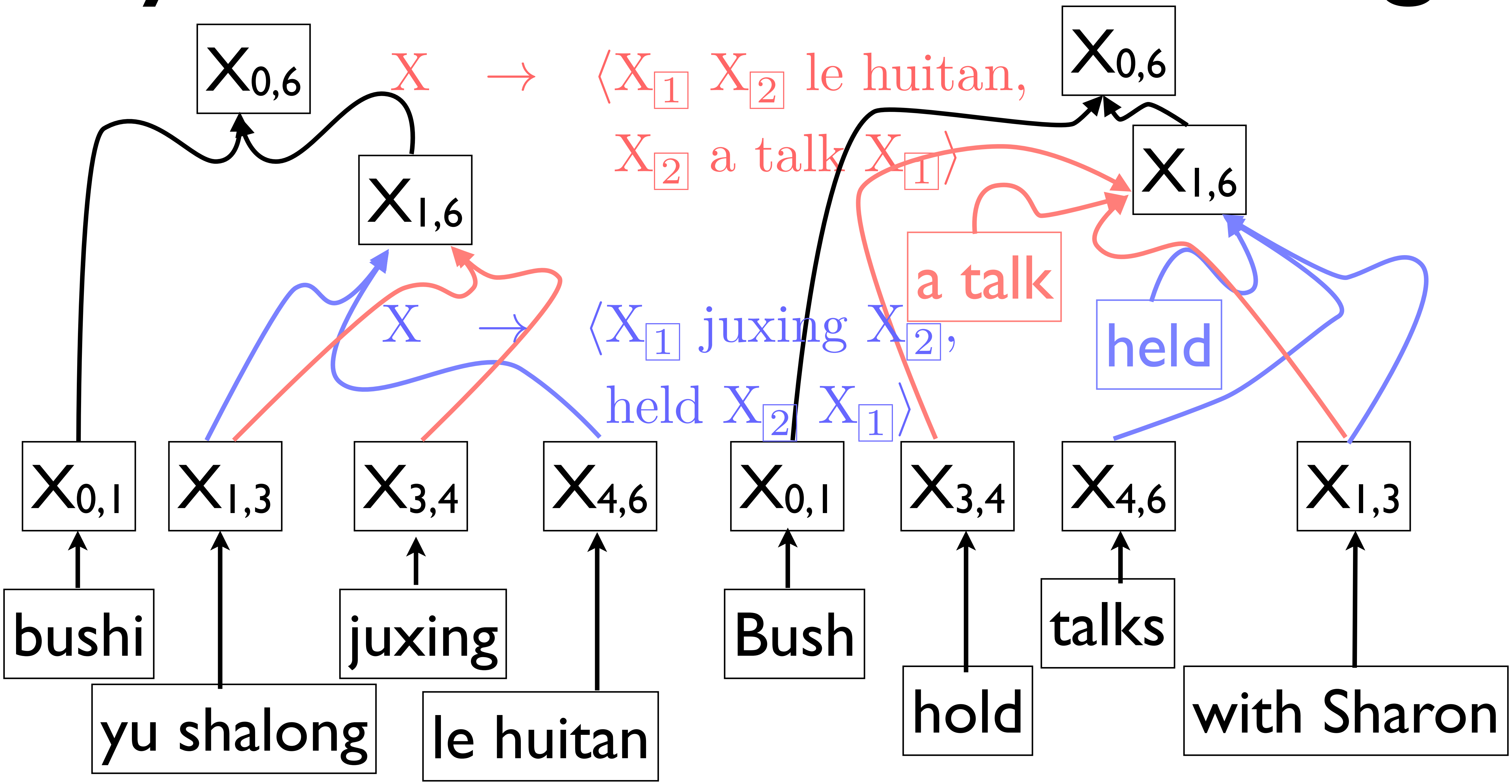
Remarks on Rules

- Too many rules extracted (Chiang, 2007):
  - at most two non-terminal symbols
  - at least one terminal between non-terminals in the source side
  - Span at most 15 words for “holes”
- Fractional counts (Chiang, 2007):
  - Each phrases counted in phrase-based MT
  - Fractional counts for rules sharing the same source/target span

Other Features

- Lexical weights as used in phrase-based MT
- ngram language model(s)
- word count: bias for ngram language model(s)
- rule count: shorter or longer phrases
- glue-rule counts: bias for monotonic glue rules

Synchronous-CFG: Parsing



- Parse input sentence using the source side, and construct a translation forest by target side

Synchronous-CFG: Parsing

- Translation by SCFG = monolingual parsing using the source side grammar
- Construct forest by the projected target side
- From forests, compute the best derivation (Huang and Chiang, 2005)
- Complexity: $O(n^3)$ as in monolingual CKY

Non-Local Features

$$X \rightarrow \langle X_{\boxed{1}} \text{ juxing } X_{\boxed{2}}, \; \text{held } X_{\boxed{2}} \; X_{\boxed{1}} \rangle$$

$X_{1,6}$

$X_{3,4}$ $X_{1,3}$

held a talk with Sharon

held talks with Sharon

held a talk and Sharon

held meeting Sharon with

Update boundary words only

p(talk | a)

a talk

talks

meeting

meetings

with Sharon — p(Sharon | with)

and Sharon — p(Sharon | and)

Sharon with — p(with | Sharon)

Sharon and — p(and | Sharon)

- non-local features which requires out-of-span context, i.e. bigram LM

Bigram Features

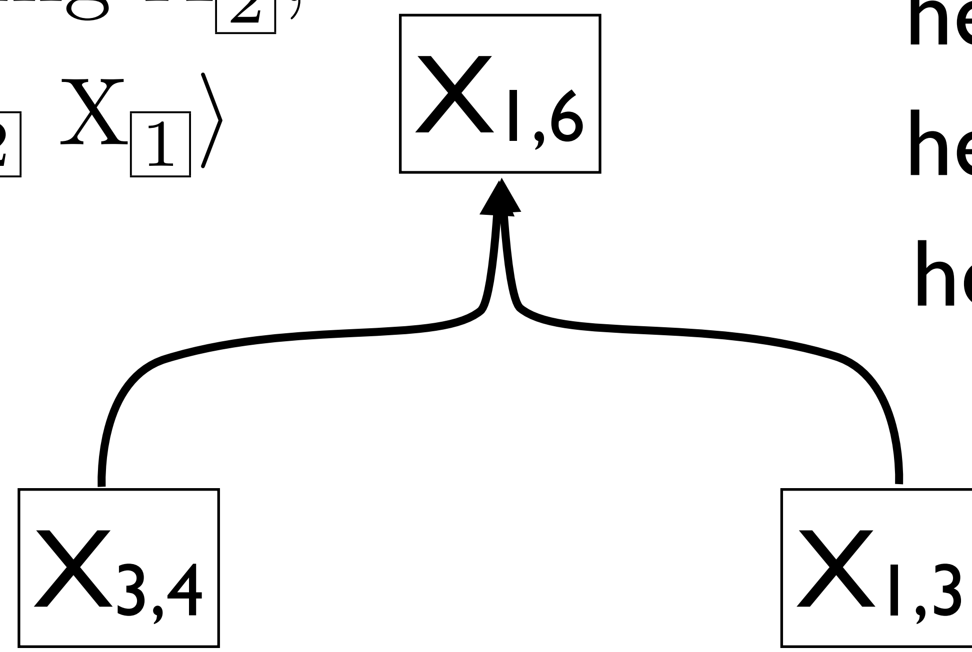
$$X \rightarrow \langle X_{\boxed{1}} \text{ juxing } X_{\boxed{2}}, \; \text{held } X_{\boxed{2}} \; X_{\boxed{1}} \rangle$$

held * Sharon

held * Sharon

held * Sharon

held * with



a * talk

talks

meeting

meetings

with * Sharon

and * Sharon

Sharon * with

Sharon * and

- We keep only bigram states: (Why 2 words?)

Language Model Updates

- Each hypothesis keeps two contexts:
  - Prefix: ngrams to be scored with antecedents
  - Suffix: contexts for future ngrams (i.e. Phrase-based MT)
- Complexity: $O(n^3V^{2(m-1)})$
- Very inefficient: we need to explicitly enumerate all the hypotheses in antecedents

Forest Rescoring

- Translation by SCFG = monolingual parsing using the source side grammar
- Construct forest by the projected target side + Rescore with non-local features
- From forests, compute the best derivation (Huang and Chiang, 2005)
- ~~Complexity: $O(n^3)$ as in monolingual CKY~~

Cube Pruning

$$X \rightarrow \langle X_{\boxed{1}} \text{ juxing } X_{\boxed{2}},\ \text{held } X_{\boxed{2}}\ X_{\boxed{1}} \rangle$$

		with * Sharon	and * Sharon	Sharon * with	Sharon * and
		1.5	1.7	2.6	3.2
a * talk	1.0	2.5	2.7	3.6	4.2
talks	1.3	2.8	3.0	3.9	4.5
meeting	2.2	3.7	3.9	4.8	5.4
meetings	2.6	4.1	4.3	5.2	5.8

- For each hyperedge, create a "cube" representing combinations of antecedents (Huang and Chiang, 2007)

Cube Pruning

$$X \rightarrow \langle X_{\boxed{1}} \text{ juxing } X_{\boxed{2}}, \text{ held } X_{\boxed{2}} X_{\boxed{1}} \rangle$$

		with * Sharon	and * Sharon	Sharon * with	Sharon * and
		1.5	1.7	2.6	3.2
a * talk	1.0	2.5 +0.5	2.7 +1.0	3.6 +1.5	4.2 +1.5
talks	1.3	2.8 +0.3	3.0 +1.5	3.9 +2.0	4.5 +2.0
meeting	2.2	3.7 +0.5	3.9 +1.0	4.8 +1.5	5.4 +1.5
meetings	2.6	4.1 +0.3	4.3 +1.5	5.2 +2.0	5.8 +2.0

- Bigrams require contexts from antecedents: non-monotonic scoring

Cube Pruning

queue: (0,0)

k-best:

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0			
talks	1.3				
meeting	2.2				
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue:

k-best: (0,0)

		with * Sharon	and * Sharon	Sharon * with	Sharon * and
		1.5	1.7	2.6	3.2
a * talk	1.0	3.0			
talks	1.3				
meeting	2.2				
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (0,1)(1,0)

k-best: (0,0)

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0	3.7		
talks	1.3	3.1			
meeting	2.2				
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (1,0)

k-best: (0,0)(0,1)

		with * Sharon	and * Sharon	Sharon * with	Sharon * and
		1.5	1.7	2.6	3.2
a * talk	1.0	3.0	3.7		
talks	1.3	3.1			
meeting	2.2				
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (1,0) (0,2) (1,1)

k-best: (0,0) (0,1)

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0	3.7		
talks	1.3	3.1	4.5		
meeting	2.2	4.2			
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (0,2) (1,1)

k-best: (0,0) (0,1) (1,0)

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0	3.7		
talks	1.3	3.1	4.5		
meeting	2.2	4.2			
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (0,2) (1,1) (3,0)

k-best: (0,0) (0,1) (1,0)

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0	3.7	5.1	
talks	1.3	3.1	4.5		
meeting	2.2	4.2			
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

queue: (1,1) (3,0)

k-best: (0,0) (0,1) (1,0) (0,2)

		with * Sharon 1.5	and * Sharon 1.7	Sharon * with 2.6	Sharon * and 3.2
a * talk	1.0	3.0	3.7	5.1	
talks	1.3	3.1	4.5		
meeting	2.2	4.2			
meetings	2.6				

- Starting from the upper-left corner, enumerate antecedent combinations

Cube Pruning

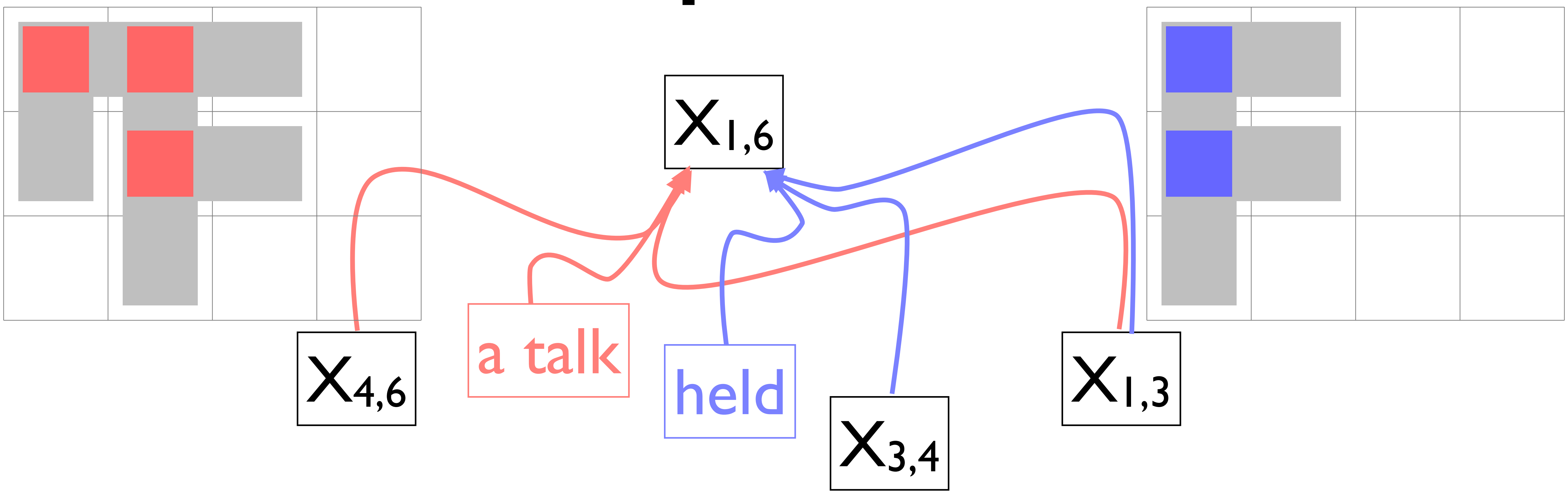
queue: (0,4) (1,1)(1,2) (3,0)

k-best: (0,0)(0,1) (1,0) (0,2)

		with * Sharon	and * Sharon	Sharon * with	Sharon * and
		1.5	1.7	2.6	3.2
a * talk	1.0	3.0	3.7	5.1	
talks	1.3	3.1	4.5		
meeting	2.2	4.2	4.9		
meetings	2.6	4.4			

- Starting from the upper-left corner, enumerate antecedent combinations

Multiple Rules



- Multiple rules sharing the same span are queued
  - Each rule is associated with a cube
  - hypothesis = hyperedge + cube-position

Further Faster Pruning

- Cube Growing (Huang and Chiang, 2007)
  - Top-down pruning combined with heuristic estimates
- Faster Cube Pruning (Gesmundo and Henderson, 2010)
  - Eliminate bookkeeping for inserted hypotheses by determining the ordering of cube enumerations
  - Push minimum hypotheses by looking up ancestors
- Incremental (Huang and Mi, 2010)
  - Top-down decoding as in (Watanabe et al., 2006)

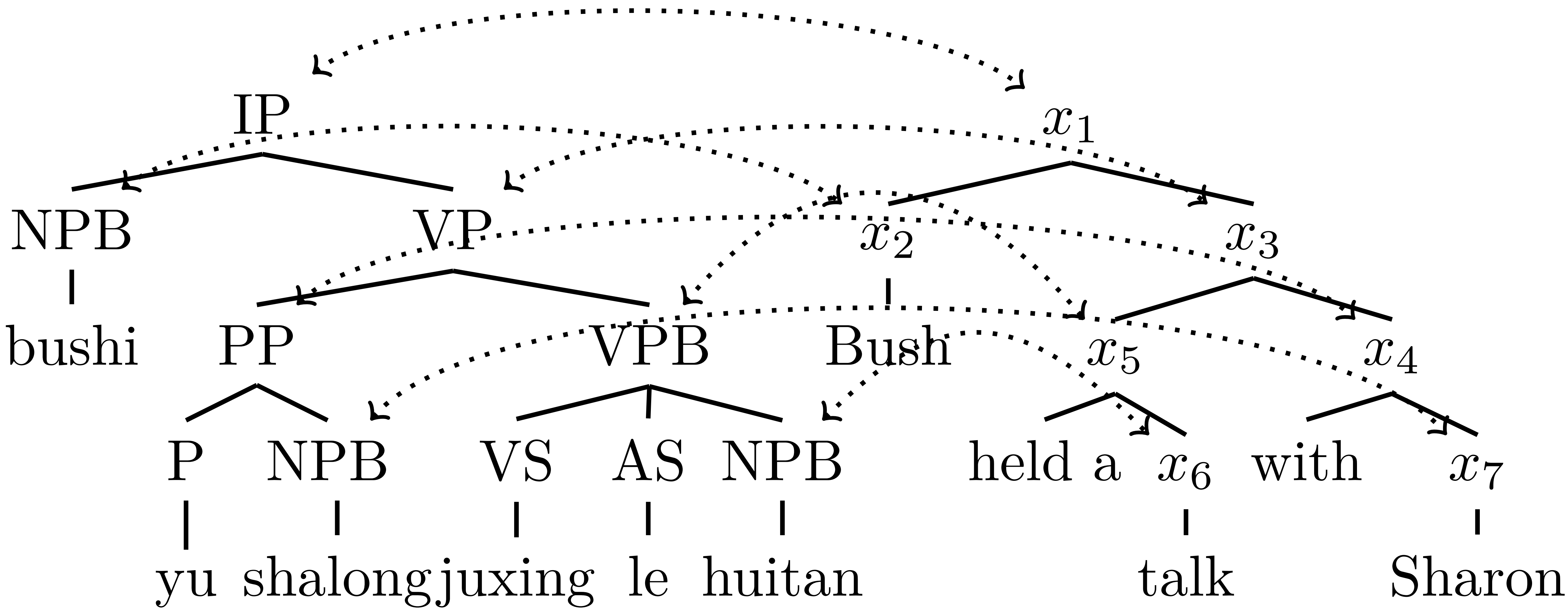
Conclusion

- Synchronous-CFG
  - paired CFG + shared non-terminal symbols
- Training is based on phrase-based MT by treating sub-phrase as a non-terminal
- Decoding: monolingual parsing
  - An efficient antecedent combination via cube-pruning

Tree-based MT

- Backgrounds
  - CFG, parsing, hypergraph, deductive system semirings
- Tree-based SMT
  - Synchronous-CFG
  - String-to-Tree, Tree-to-String

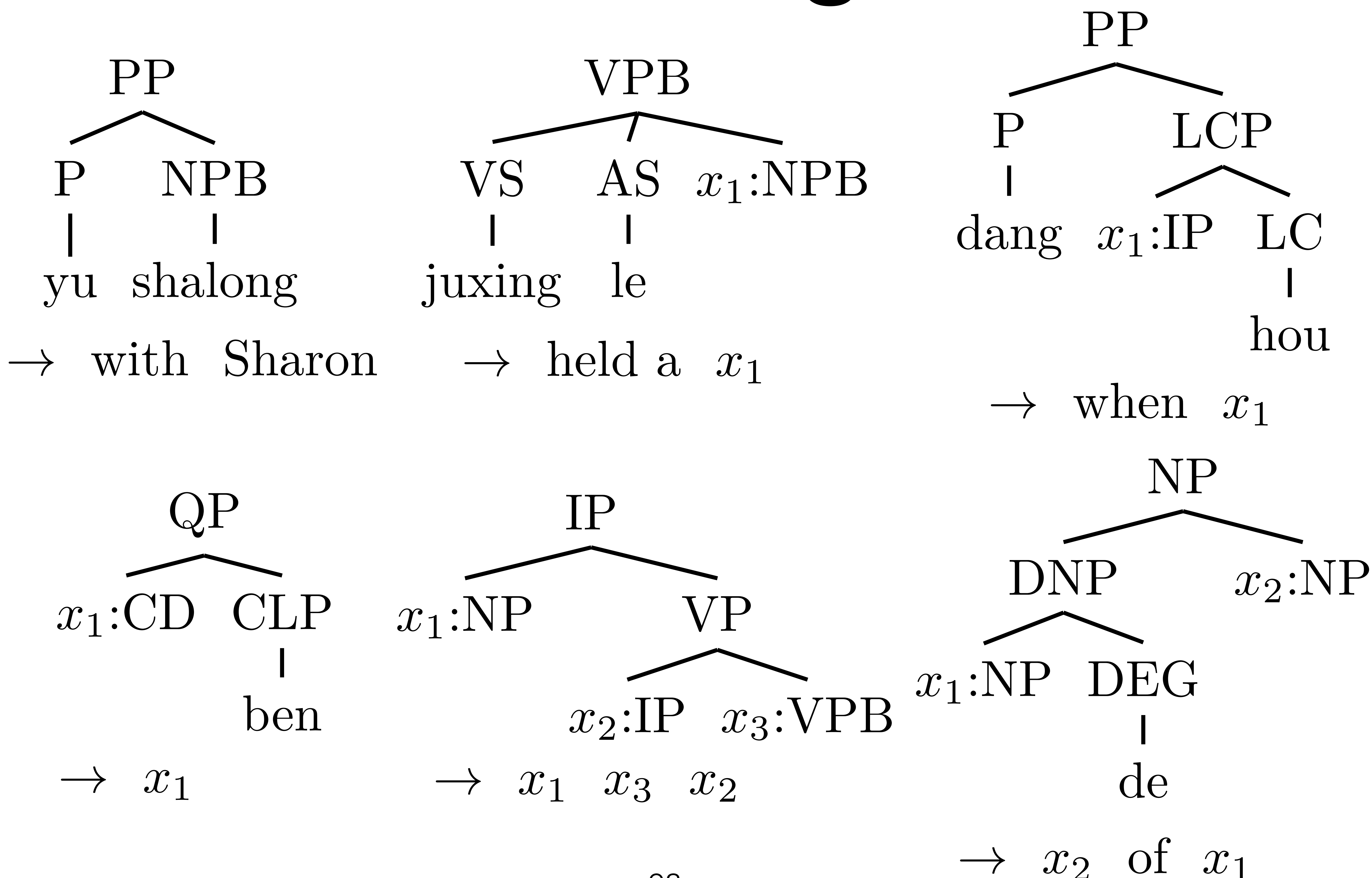
{Tree,String}-to-{Tree,String}



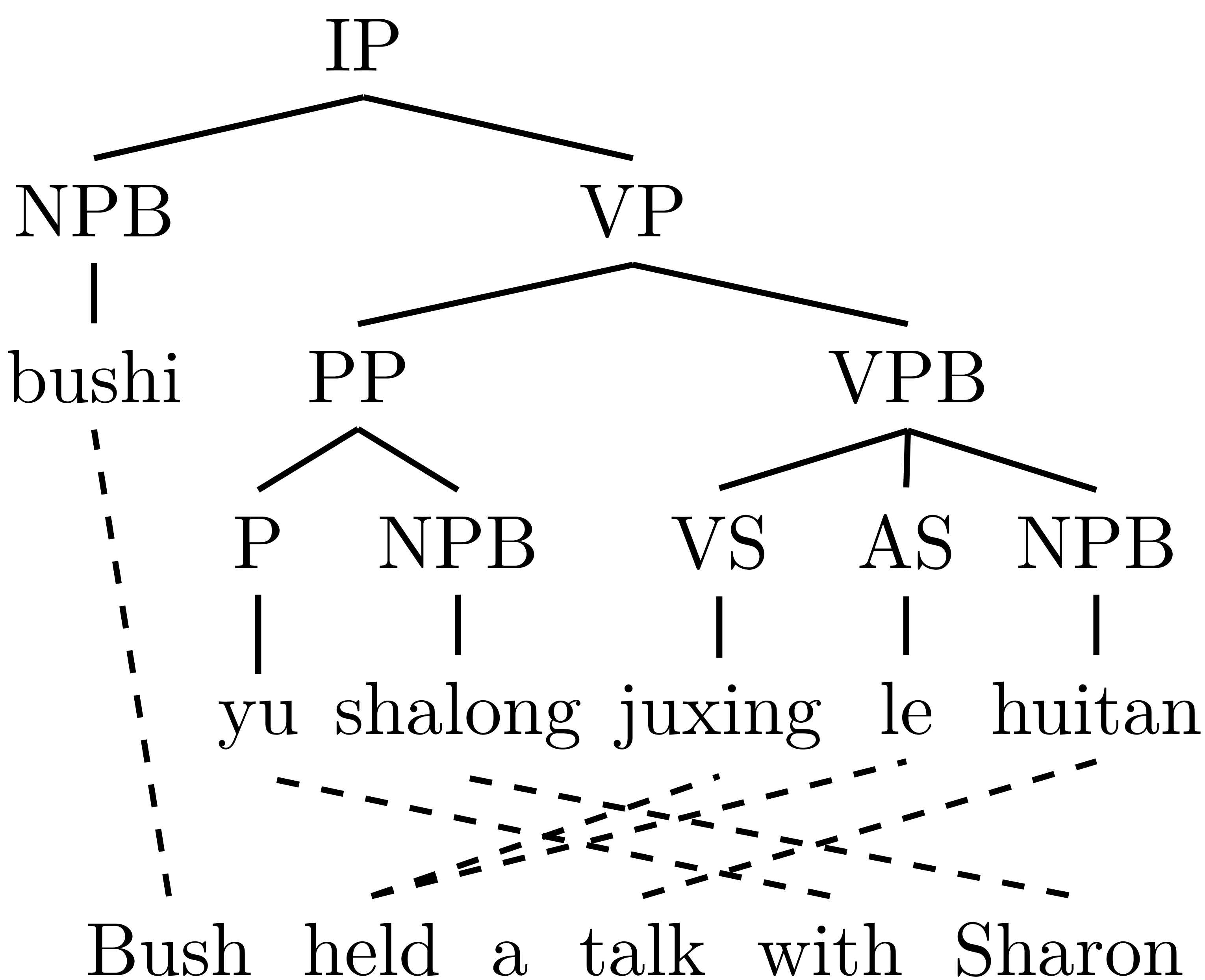
(Galley et al., 2004)

- Each synchronous rule has a subtree structure
- Flat structure + sharing the same non-terminal symbols = synchronous-CFG

Tree-to-String Rules



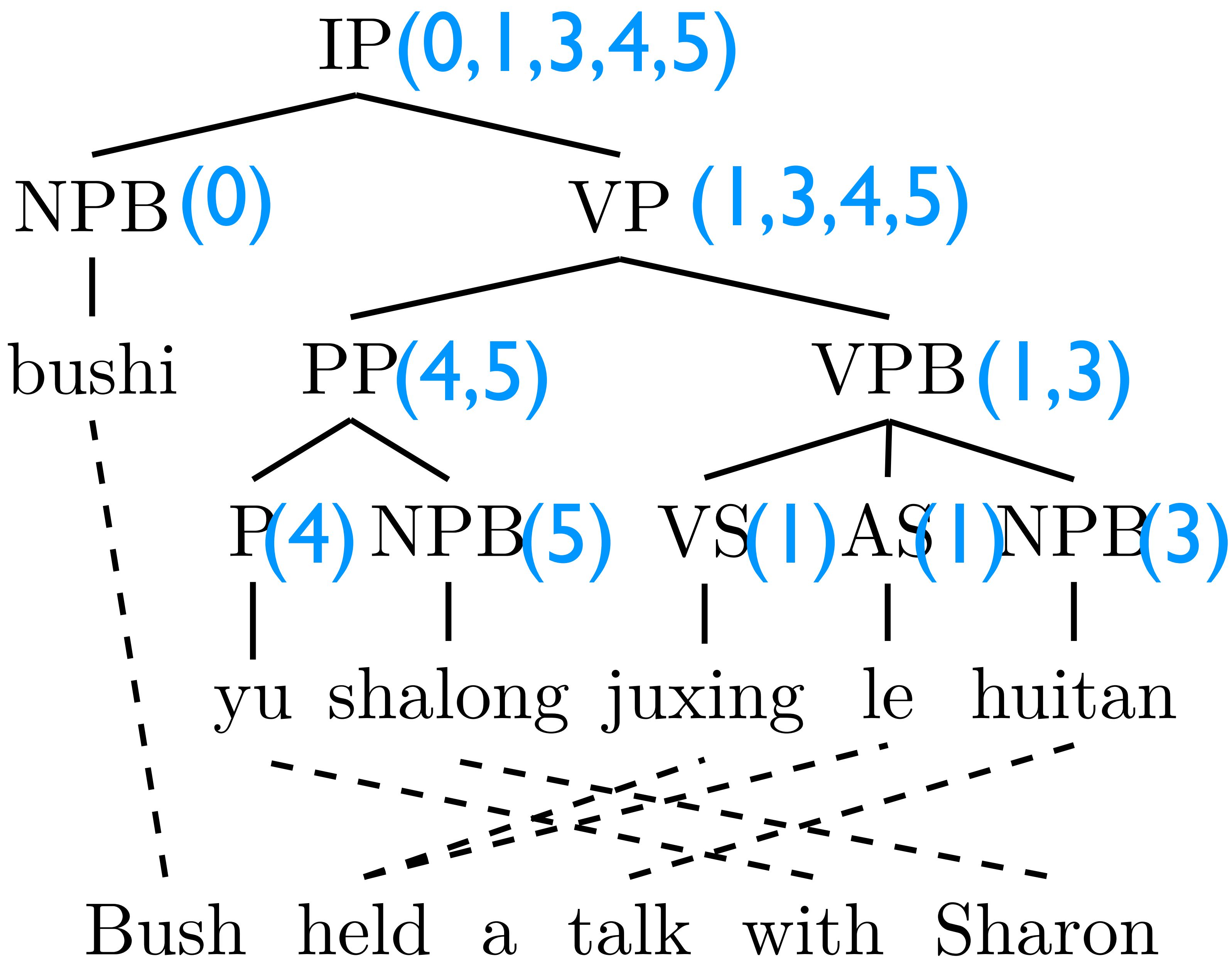
ルールの抽出



(Galley et al., 2004)

- Compute “minimum rules” as in phrase-based MT

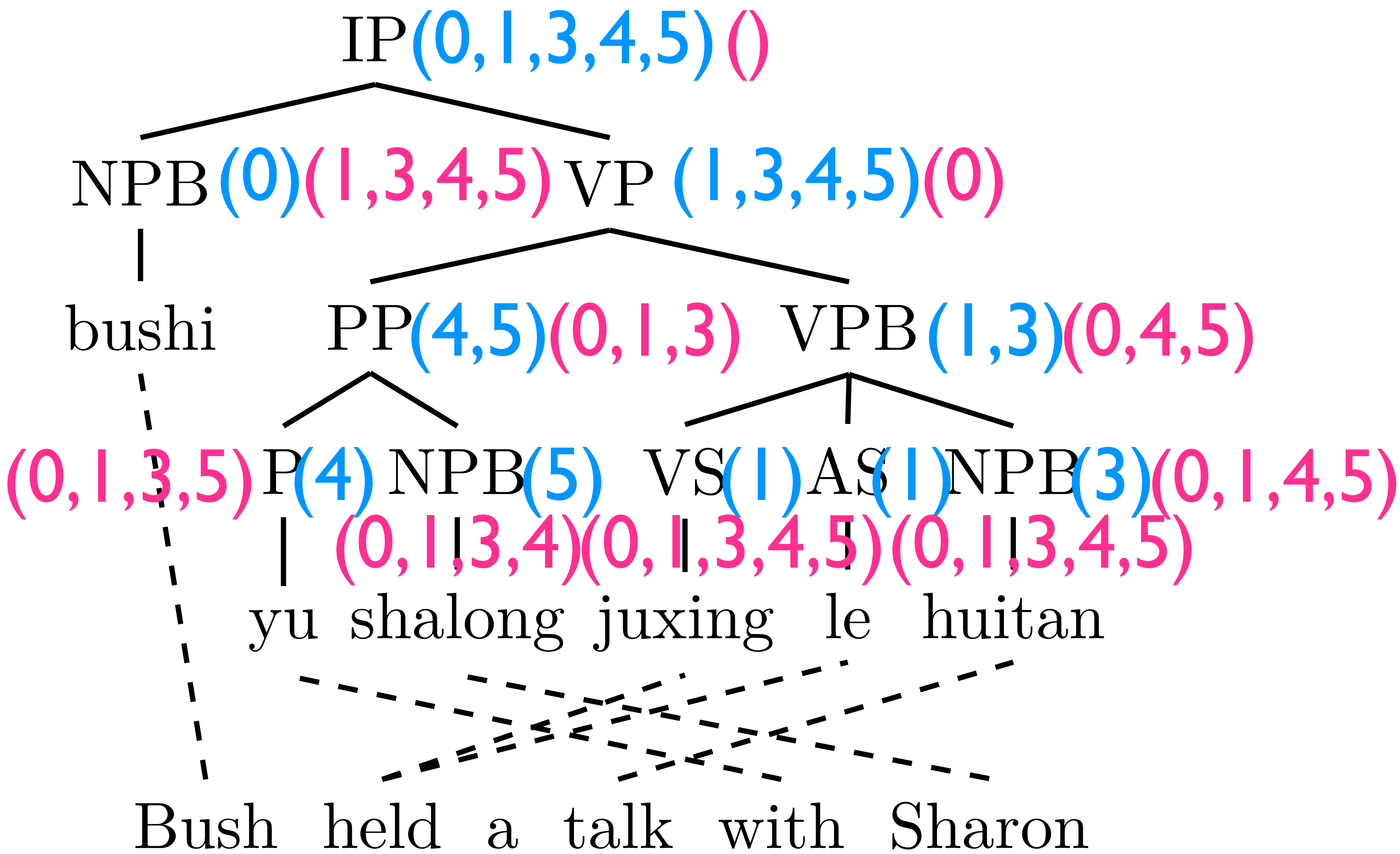
Rule Extraction



(Galley et al., 2004)

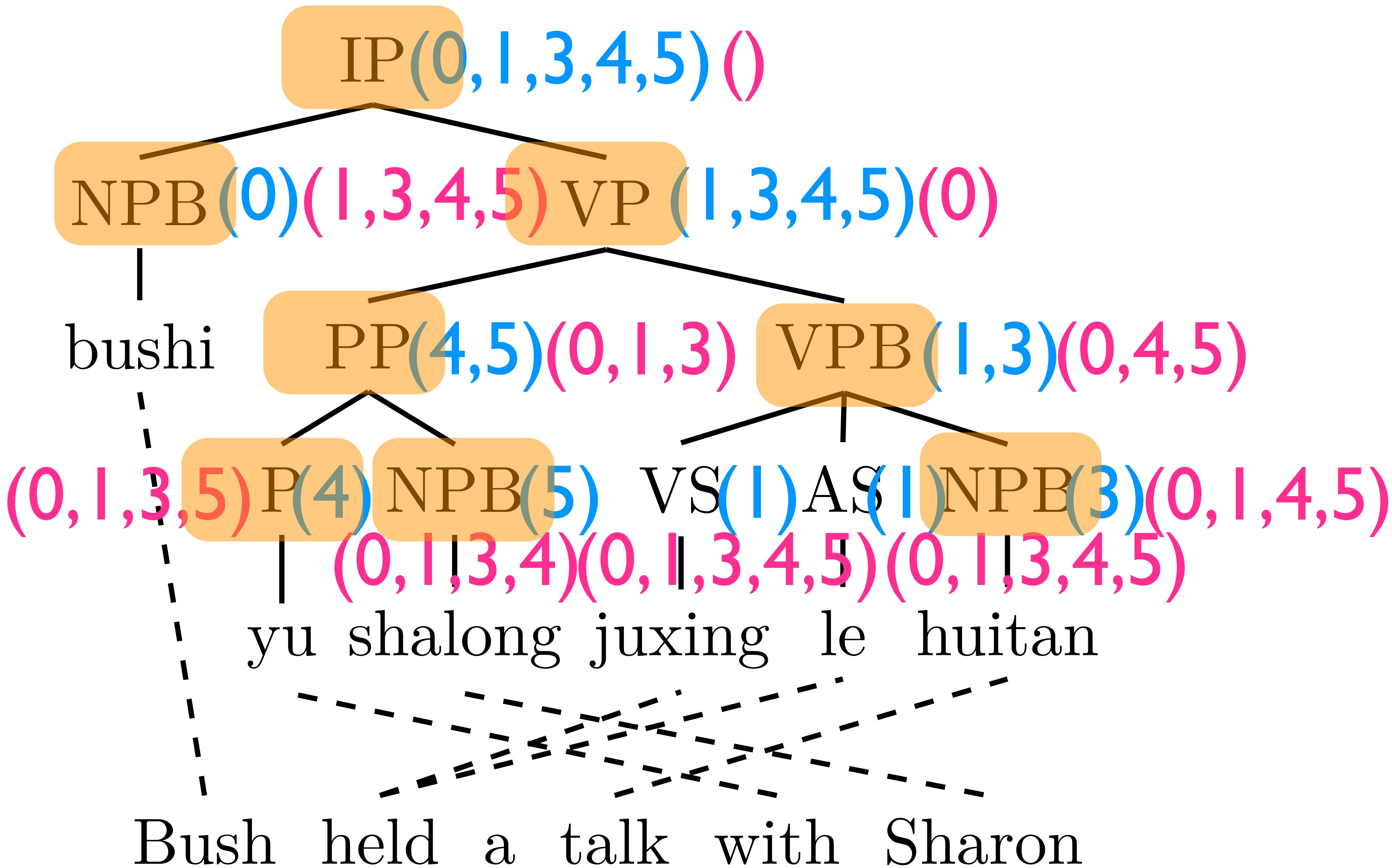
- Compute "spans" by propagating alignment in bottom-up

Rule Extraction



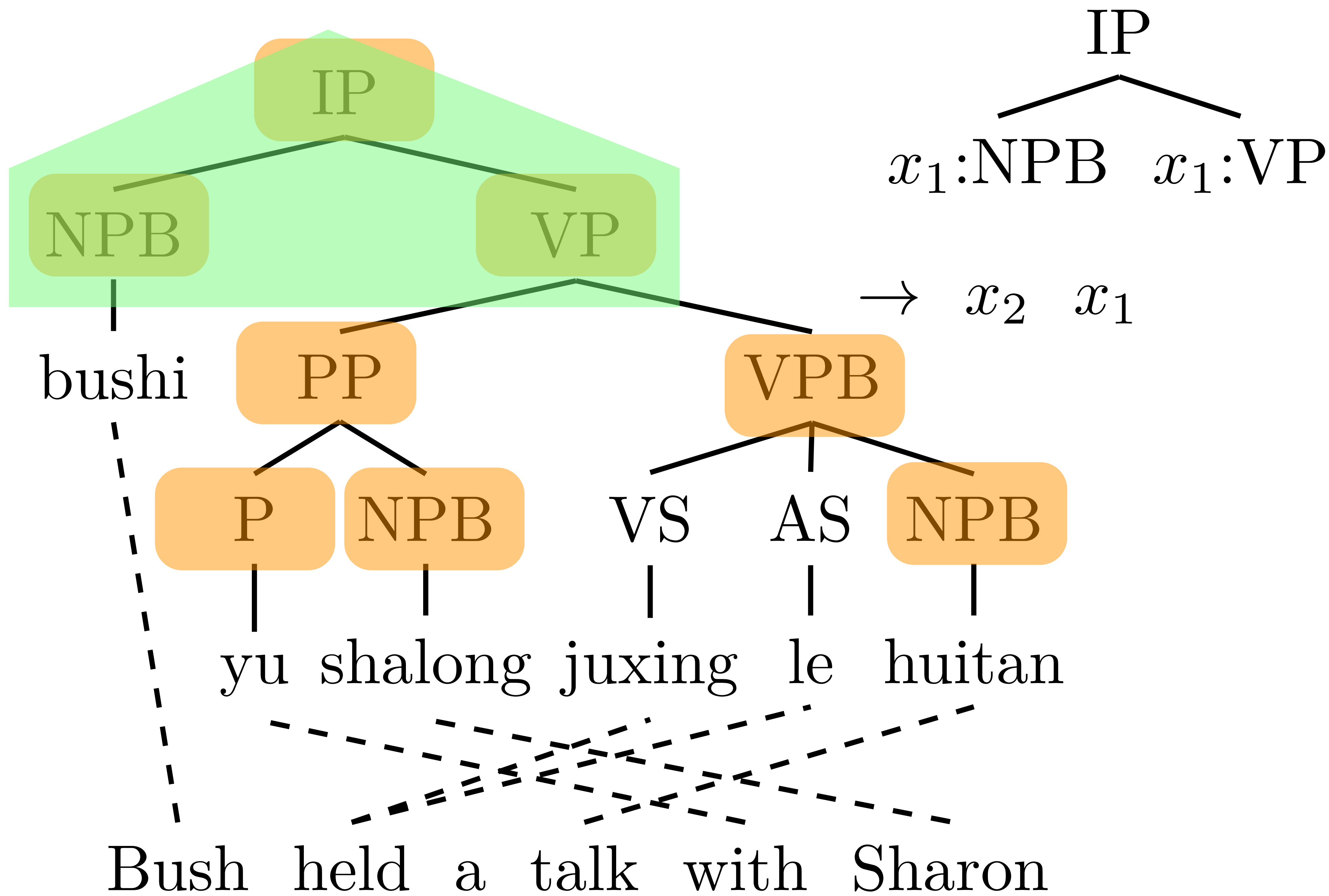
- Compute "complements" in top-down

Rule Extraction



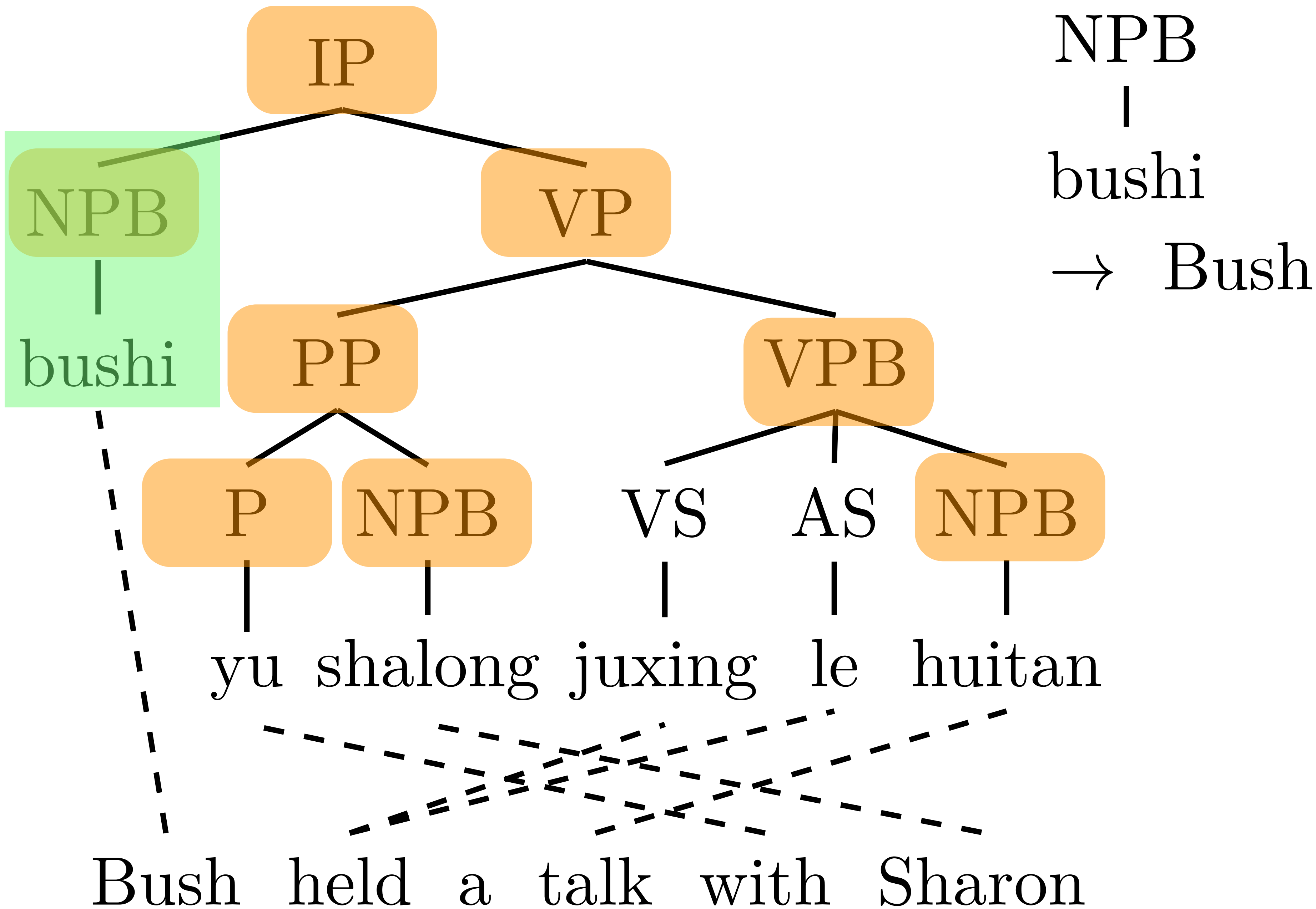
- Compute "frontiers": The nodes in which the intersection of "spans" and "complements" is empty

Rule Extraction



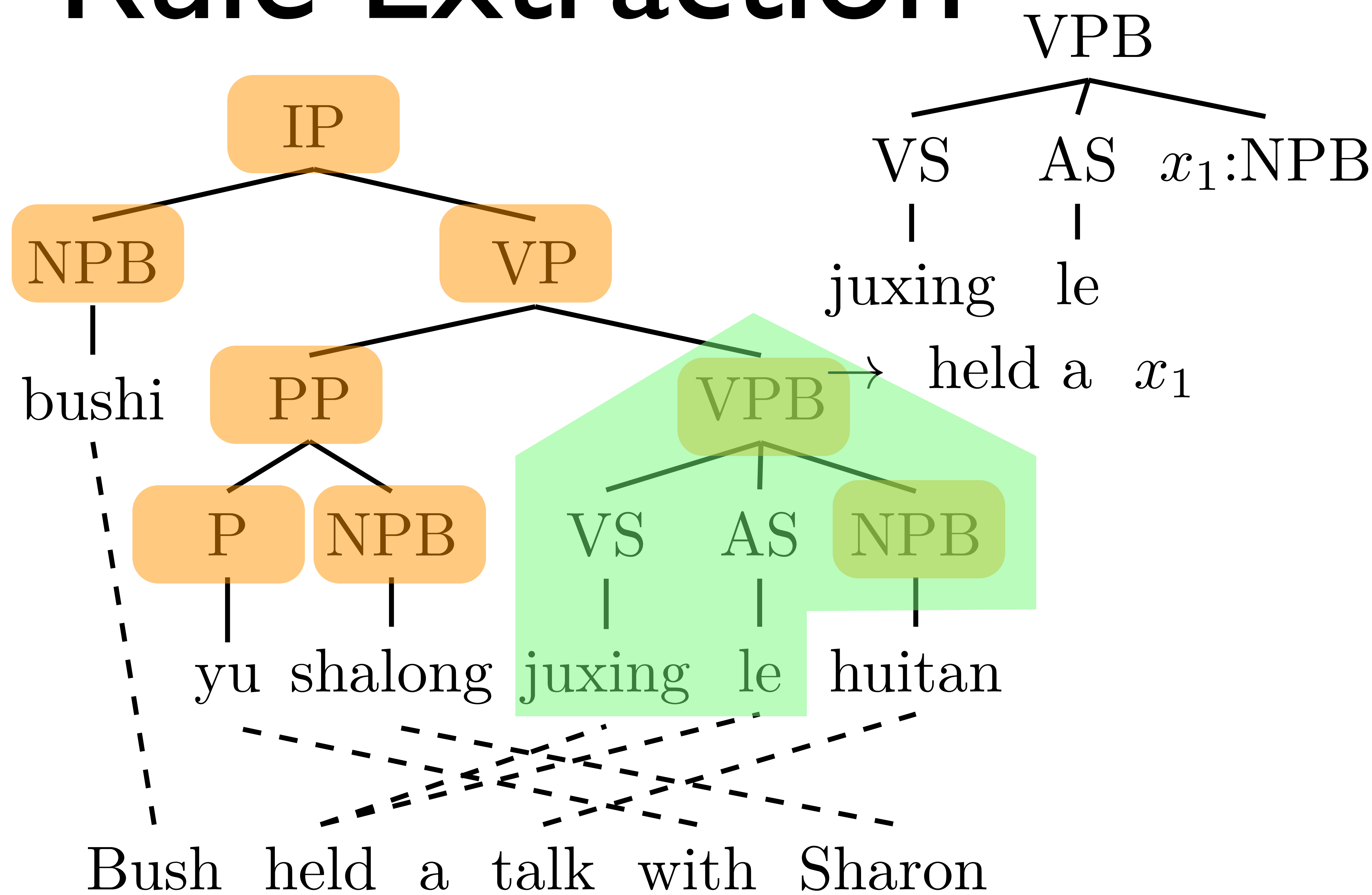
- Extract minimum rules using frontiers

Rule Extraction



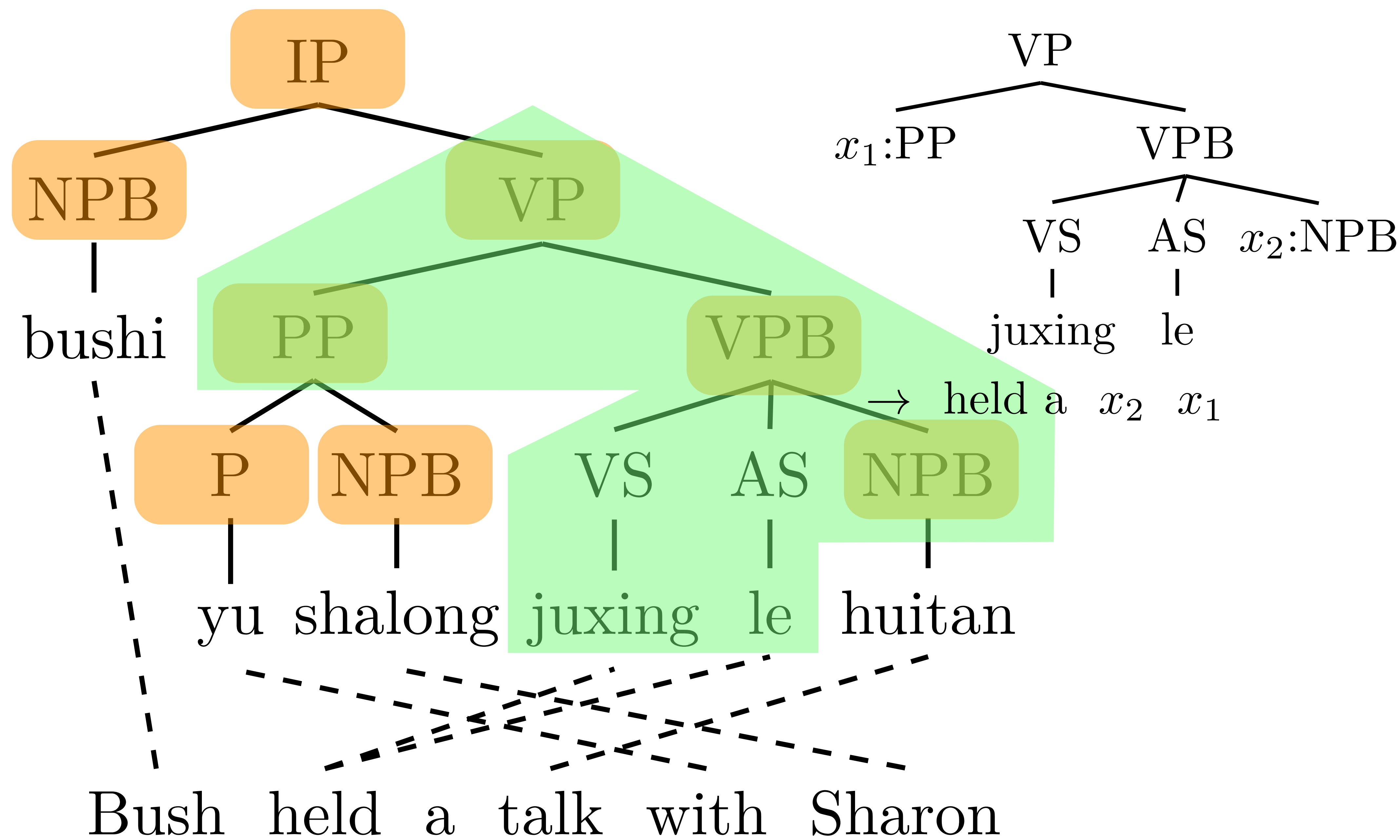
- Extract minimum rules using frontiers

Rule Extraction



- Extract minimum rules using frontiers

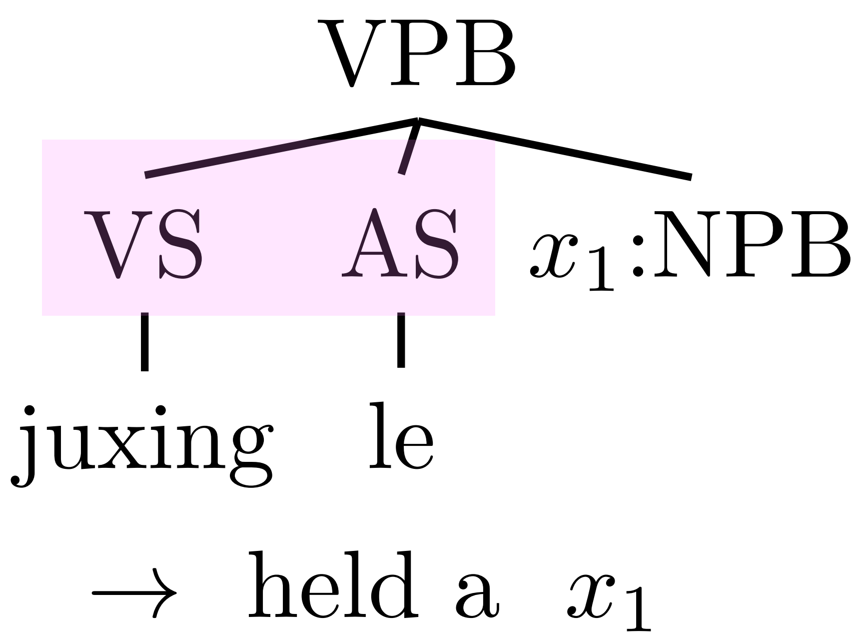
Rule Extraction

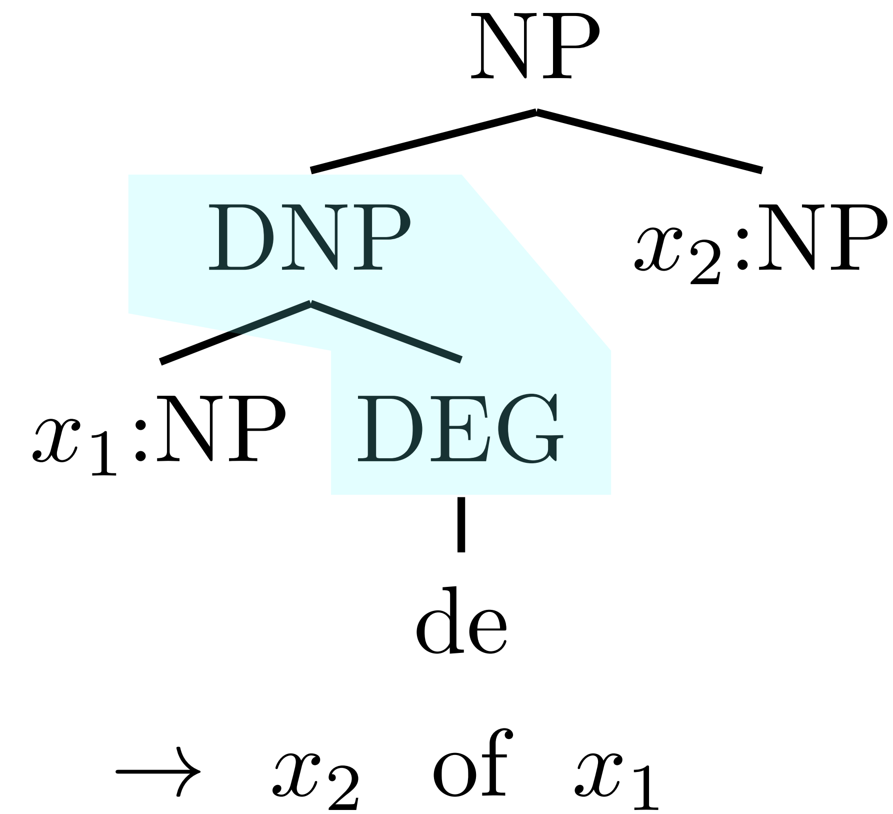


(Galley et al., 2006)

- Extract "compound rules" by combining minimum rules (i.e. longer phrases)

Decoding: String-{String, Tree}



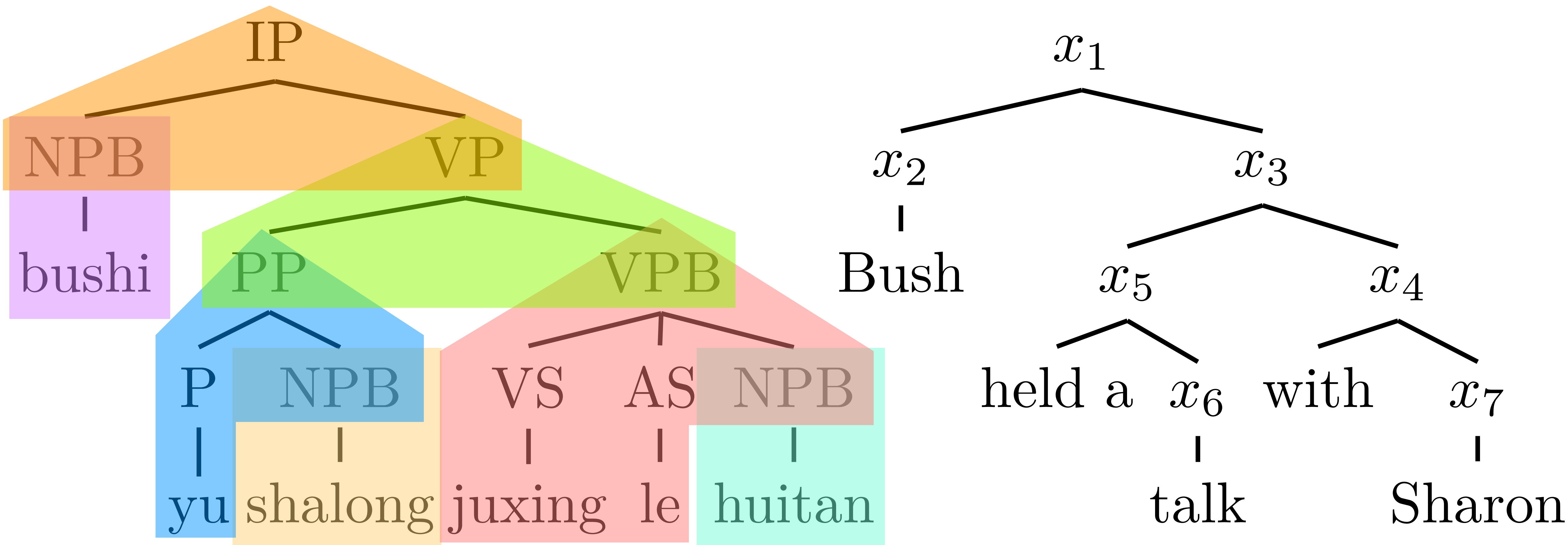


$$\begin{aligned} \langle \text{VPB} &\rightarrow \text{juxing le NPB}_1, \\ x &\rightarrow \text{hold a } x_1 \rangle \end{aligned} \qquad \begin{aligned} \langle \text{NP} &\rightarrow \text{NP}_1 \text{ de NP}_2, \\ x &\rightarrow x_2 \text{ of } x_1 \rangle \end{aligned}$$

(Galley et al., 2004)

- Similar to SCFG decoding: Use the “collapsed” source side rule to perform CKY parsing
- Construct a translation forest using the target side

Decoding: Tree-{String, Tree}



(Huang et al., 2006)

- First, an input sentence is parsed
- Input tree is transformed into a translation forest by tree rewriting

Forest Rescoring

- Translation by {tree,string}-to-{tree,string}
  - string-to-{tree,sting}: parsing using the source-side grammar
  - tree-to-{tree,string}: parse input sentences + tree-match-rewrite
- Construct forest by the projected target side
- From forests, compute the best derivation (Huang and Chiang, 2005)

Conclusion

- {String, Tree}-to-{String, Tree} translation models
- Rules extraction by GHKM (Galley et al., 2004)
  - Galley M, Hopkins M, Knight K, Marcu D, 2004
- Decoding:
  - String-to-{String, Tree} by CKY
  - Tree-to-{String, Tree} by tree-rewrite

More on Tree-based Models

- Forest-based approach: instead of 1-best parse, use forest encoding k-bests (Mi and Huang, 2008; Mi et al., 2008)
- “Binarized forest” as an alternative to represent multiple parses (Zhang et al., 2011)
- Fuzzy tree-to-tree as a way to overcome “stricktness” of tree-based models (Chiang, 2010)
- Use of dependency (Mi and Liu, 2010; Xie et al., 2011)

Tree-based MT

- Backgrounds
  - CFG, parsing, hypergraph, deductive system semirings
- Tree-based SMT
  - Synchronous-CFG
  - String-to-Tree, Tree-to-String

Structures in SMT

- Tutorial
  - Phrase-based MT
  - Tree-based MT
- Syntactic Structures in System Combination

MT System Combination by Confusion Forest

Taro Watanabe and Eiichiro Sumita
@ NICT

MT System Combination

- Better translation by combining multiple system outputs:
  - Sentence selection(Nomoto, 2004; etc.)
  - Phrasal combination (Frederking and Nirenburg, 1994; etc.)
  - Word level combination (Bangalore et al., 2001; Matusov et al., 2006; etc.)
- This Work: Syntactic combination, not word-wise combination

Confusion Network

★ I saw saw the the forest forest

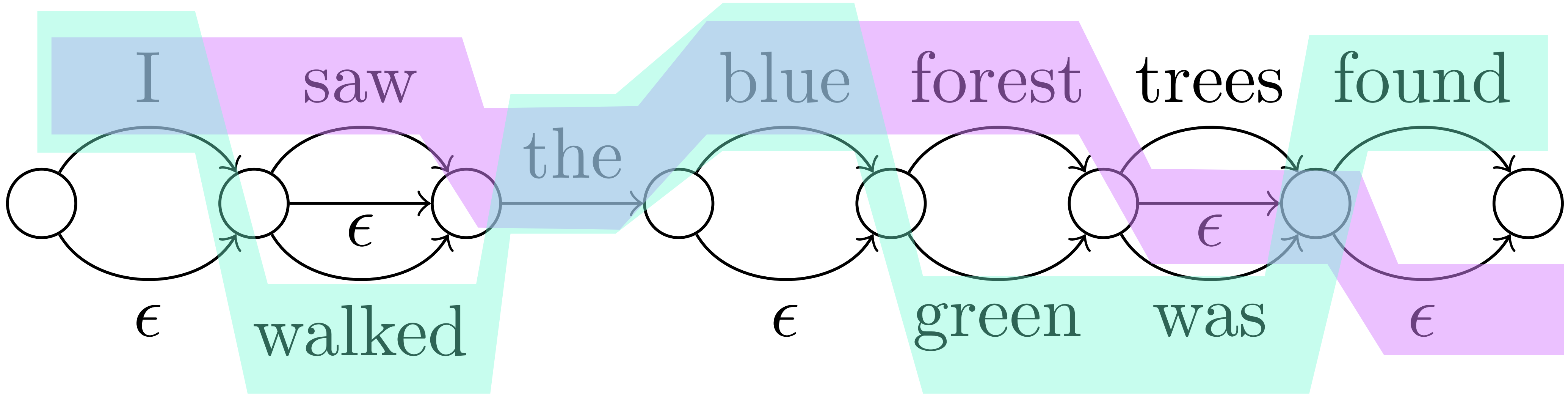
I walked walked the the blue blue forest forest

I saw saw the the green green trees trees

the forest the was found forest was found

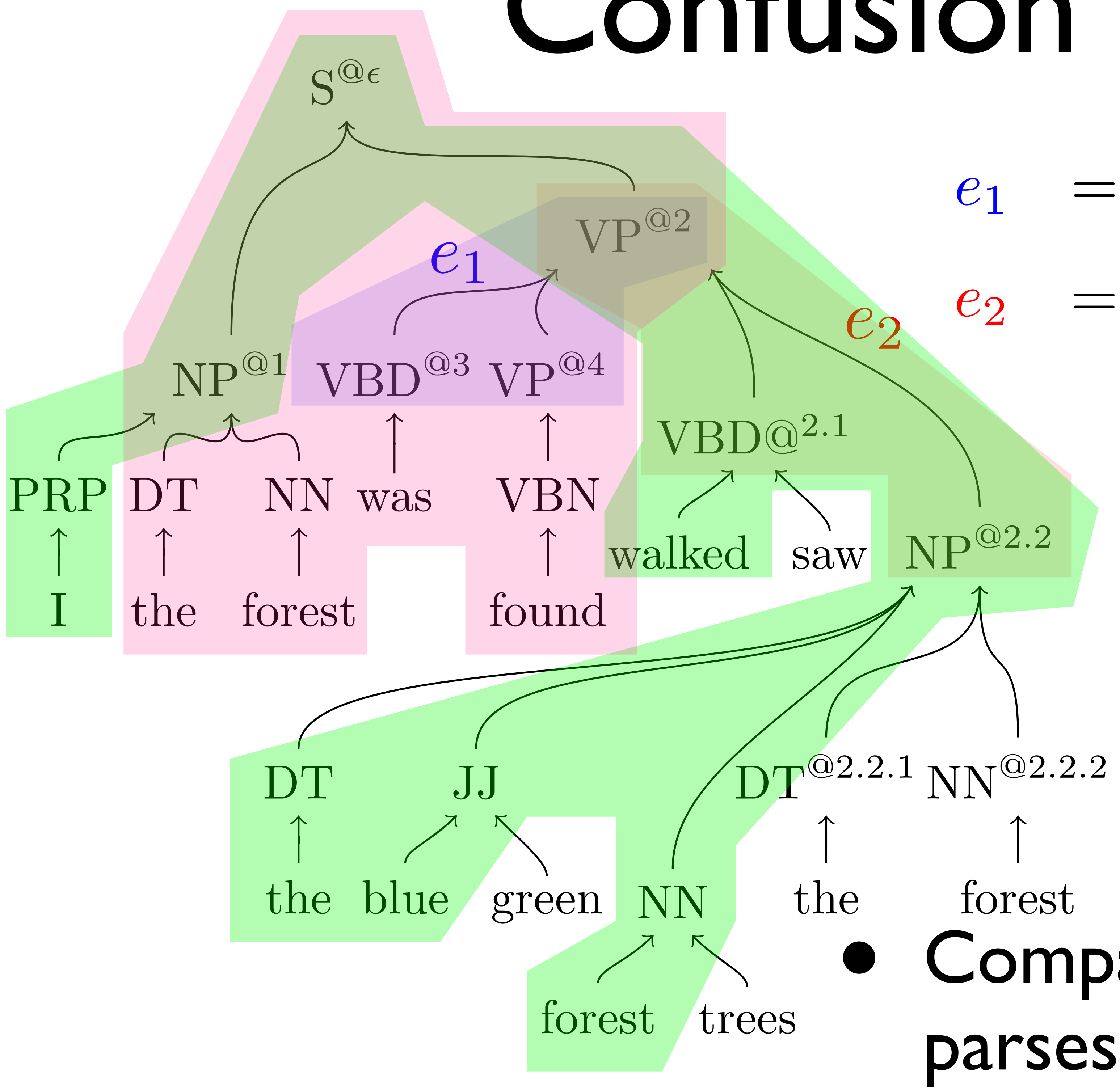
- State-of-the-art: Confusion Network
- Choose a skeleton, compute word alignment against the skeleton
  - Edit-distance-based alignment (TER etc.) (Sim et al., 2007)
  - Model-based alignment(GIZA++ etc.) (Matsov et al., 2006)

Confusion Network



- Construct a network with each arc representing alternative translation
  - Best path = Best translation
  - Syntactically different language pairs: i.e. active/passive voices
  - Spurious insertion/repetition due to alignment error
  - Incremental alignment/construction + merge multiple networks into one (Rosti et al., 2008)

Confusion Forest

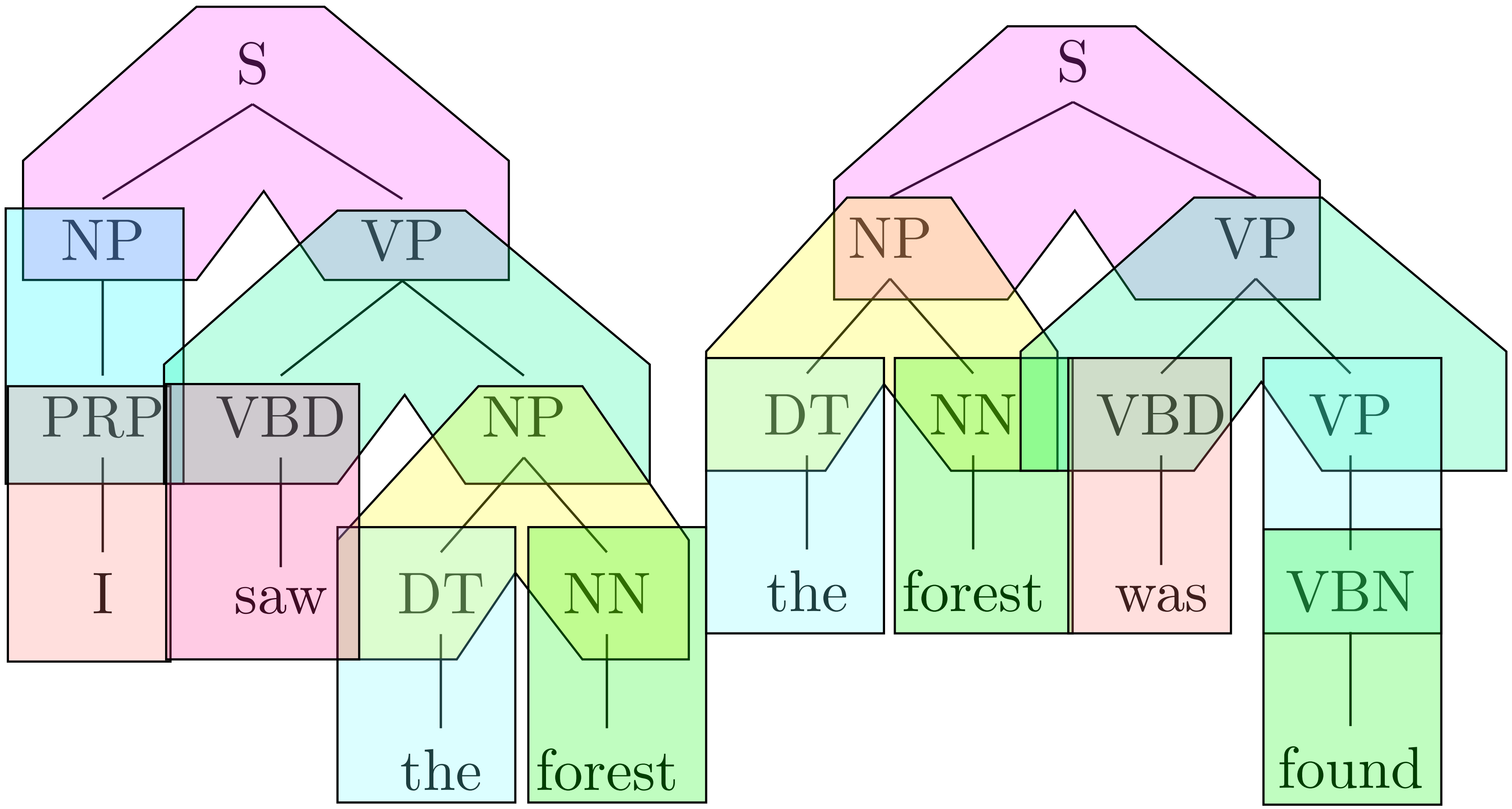


$$e_1 = \langle VP^{@2}, \{VBD^{@3}, VP^{@4}\}\rangle$$

$$e_2 = \langle VP^{@2}, \{VBD^{@2.1}, NP^{@2.2}\}\rangle$$

- Compactly represent multiple parses by sharing nodes
- Represented by "hypergraph"

Rule Extraction



S	$\rightarrow$	NP	VP
NP	$\rightarrow$	PRP	
PRP	$\rightarrow$	I	
VP	$\rightarrow$	VBD	NP
VBD	$\rightarrow$	saw	
NP	$\rightarrow$	DT	NN
DT	$\rightarrow$	the	
NN	$\rightarrow$	forest	
VP	$\rightarrow$	VBD	VP
VBD	$\rightarrow$	was	
VP	$\rightarrow$	VBN	
VBN	$\rightarrow$	found	

- Parse each system output by a parser
- Extract rules from parsed trees: local grammar

Generation by Earley

Scan:

$$\frac{[\mathrm{X} \to \alpha \bullet x\beta, h] : u}{[\mathrm{X} \to \alpha x \bullet \beta, h] : u}$$

Predict:

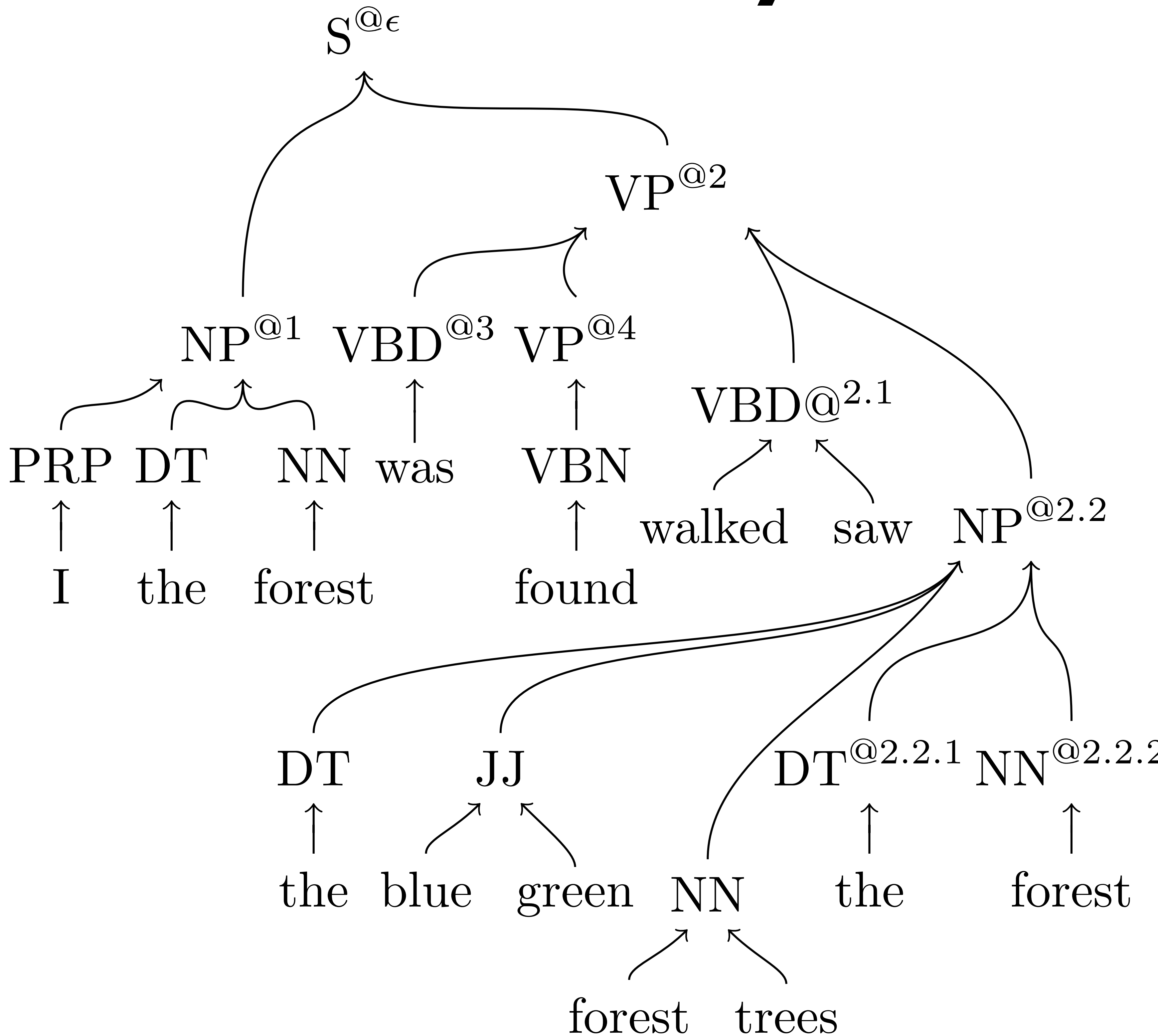
$$\frac{[\mathrm{X} \to \alpha \bullet \mathrm{Y}\beta, h]}{[\mathrm{Y} \to \bullet\gamma, h+1] : u} \quad \mathrm{Y} \xrightarrow{u} \gamma \in \mathcal{G}, h < H$$

Complete:

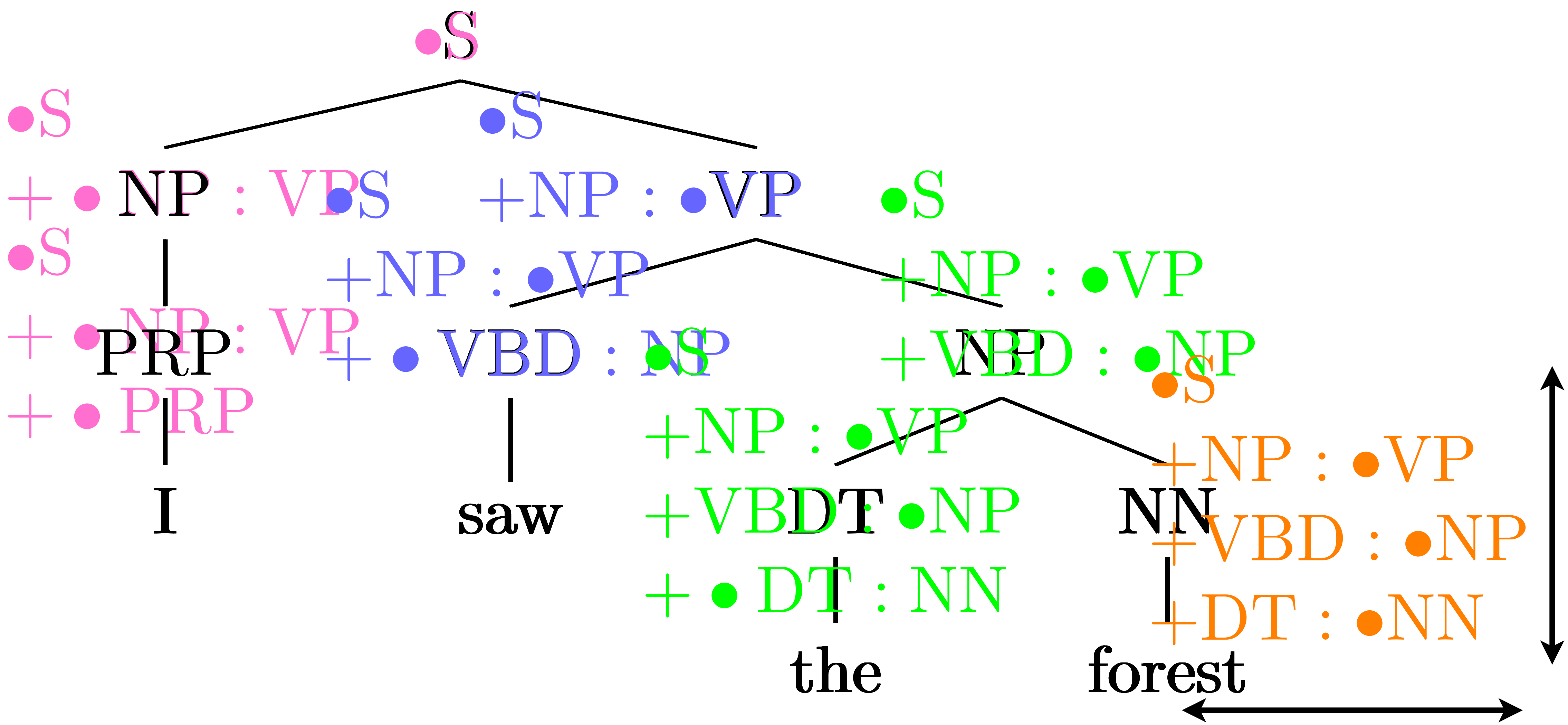
$$\frac{[\mathrm{X} \to \alpha \bullet \mathrm{Y}\beta, h] : u \quad [\mathrm{Y} \to \gamma\bullet, h+1] : v}{[\mathrm{X} \to \alpha\mathrm{Y} \bullet \beta, h] : u \otimes v}$$

- Generation from the extracted grammar
- Scanning always succeed: constraint by height

Generation by Earley



Spurious Ambiguity



- Memorize the (partial) tree structures in each node
- Employ the sequence of Ealrye state as a node
  - Horizontal/Vertical Markovization (Klein and Manning, 2003)

Forest Reranking

$$\hat{d} = \arg\max_{d \in D} \mathbf{w}^\top \cdot \mathbf{h}(d, F)$$

- Choose the best derivation d among all possible derivations D in a forest F
  - Terminal yield of the best derivation = the best translation
  - Approximately apply non-local features (ngram language models) by Cube Pruning (Huang and Chiang, 2007)
  - Efficient k-best by Algorithm 3 (Huang and Chiang, 2005)

Experiments

- WMT10 System Combination Task
  - Czech, German, Spanish, French → English
  - tune/test: 455/2,034 sentences

	cz-en	de-en	es-en	fr-en
systems	6	16	8	14
tune	10.6K	10.9K	10.9K	11.0K
test	50.5K	52.1K	52.1K	52.4K

Systems

- CF: Stanford parser + "cicada" (a hypergraph-based toolkit based on SEMIring parsing framework)
- CN: Single network by merging multiple networks + conversion into hypergraph by lattice parsing
- features: tuned by hypergraph-MERT(Kumar et al. 2009)
  - Language Models, # of terminals, # of hyperedges
  - # of rules in a derivation originally in n_{th} system output
  - BLEUs by treating each system output as a reference translation
  - Network distance (only used for CN)

BLEU

	cz-en	de-en	es-en	fr-en
system min	14.09	15.62	21.79	16.79
max	23.44	24.10	29.97	***29.17***
CN	23.70	24.09	***30.45***	***29.15***
CF, $v=\infty, h=\infty$	***24.13***	24.18	***30.41***	***29.57***
CF, $v=\infty, h=2$	***24.14***	***24.58***	***30.52***	28.84
CF, $v=\infty, h=1$	***24.01***	23.91	***30.46***	***29.32***

Oracle BLEU

	cz-en	de-en	es-en	fr-en
rerank	29.40	32.32	36.83	36.59
CN	38.52	34.97	47.65	46.37
CF, $v=\infty, h=\infty$	30.51	34.07	38.69	38.94
CF, $v=\infty, h=2$	30.61	34.25	38.87	39.10
CF, $v=\infty, h=1$	31.09	34.65	39.27	39.51

Hypergraph size

	cz-en	de-en	es-en	fr-en
CN	2,222.68	47,231.20	2,932.24	11,969.40
CF, $v=\infty$, $h=1$	230.08	540.03	262.30	386.79
CF, $v=5$, $h=1$	254.45	651.10	302.01	477.51
CF, $v=4$, $h=1$	286.01	802.79	349.21	575.17

- Average # of hyperedges
- (rough) estimates for speed

Conclusion

- System combination by Confusion Forest which employs syntactic distance, not word-level distance
- Forest construction by the grammar extracted from system outputs
    - Parser: assign tree structure to the similar expressions
- Compact date structure + comparable performance against Confusion Network

Structures in SMT

- Tutorial
  - Phrase-based MT
  - Tree-based MT
- Syntactic Structures in System Combination

Research on MT

- Reading: at least 50 papers are related to MT "every year"
- Specialist: solve a sub-problem
- Language Neutral: a solution which works only for a particular language pair is boring