

机器翻译原理与方法

第五讲 基于句法的统计机器翻译方法

刘群

中国科学院计算技术研究所

liuqun@ict.ac.cn

中国科学院计算技术研究所2011年秋季课程

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

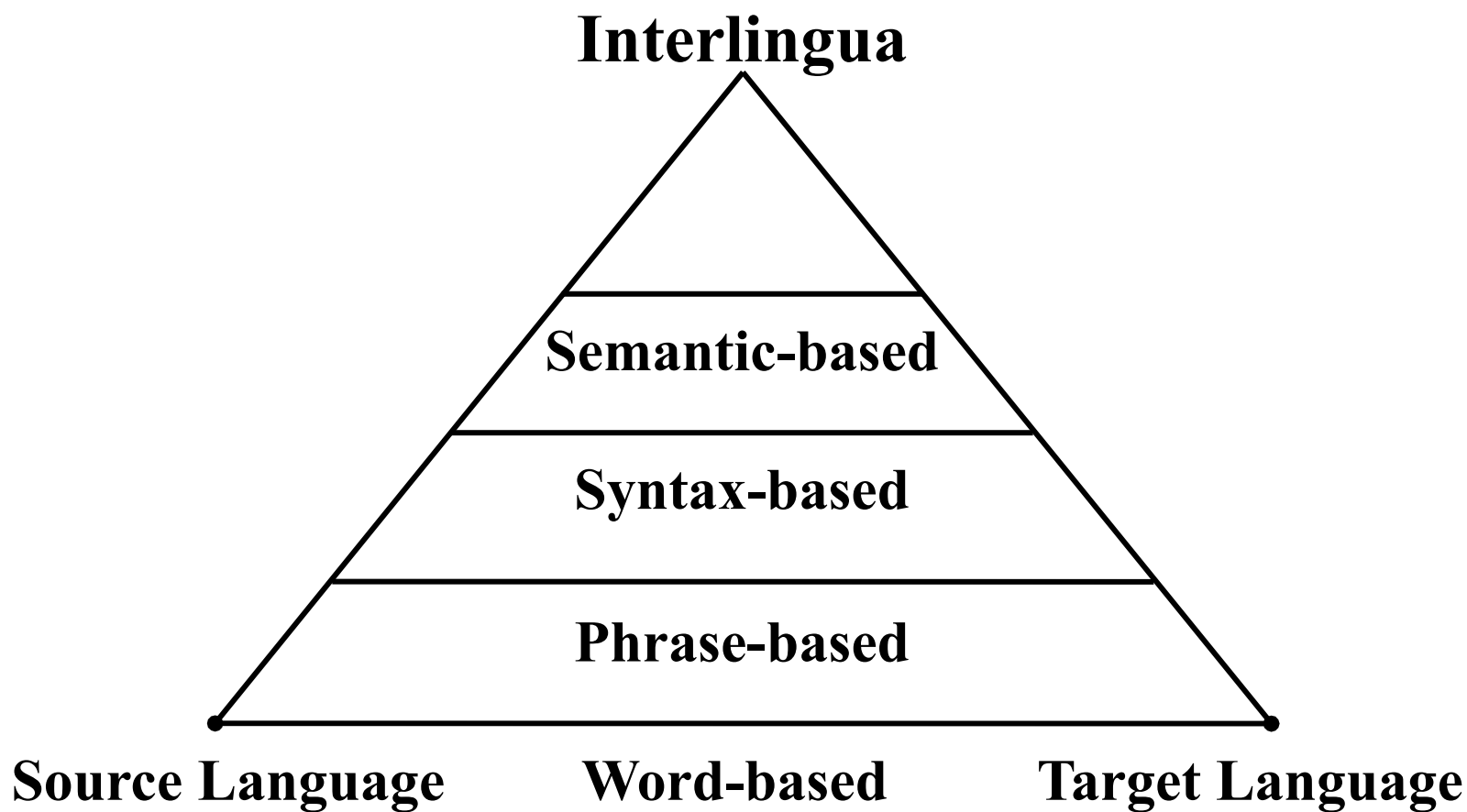
概述

- 基于短语的统计翻译方法的问题
- 基于句法的统计翻译方法的分类
- 目前的进展

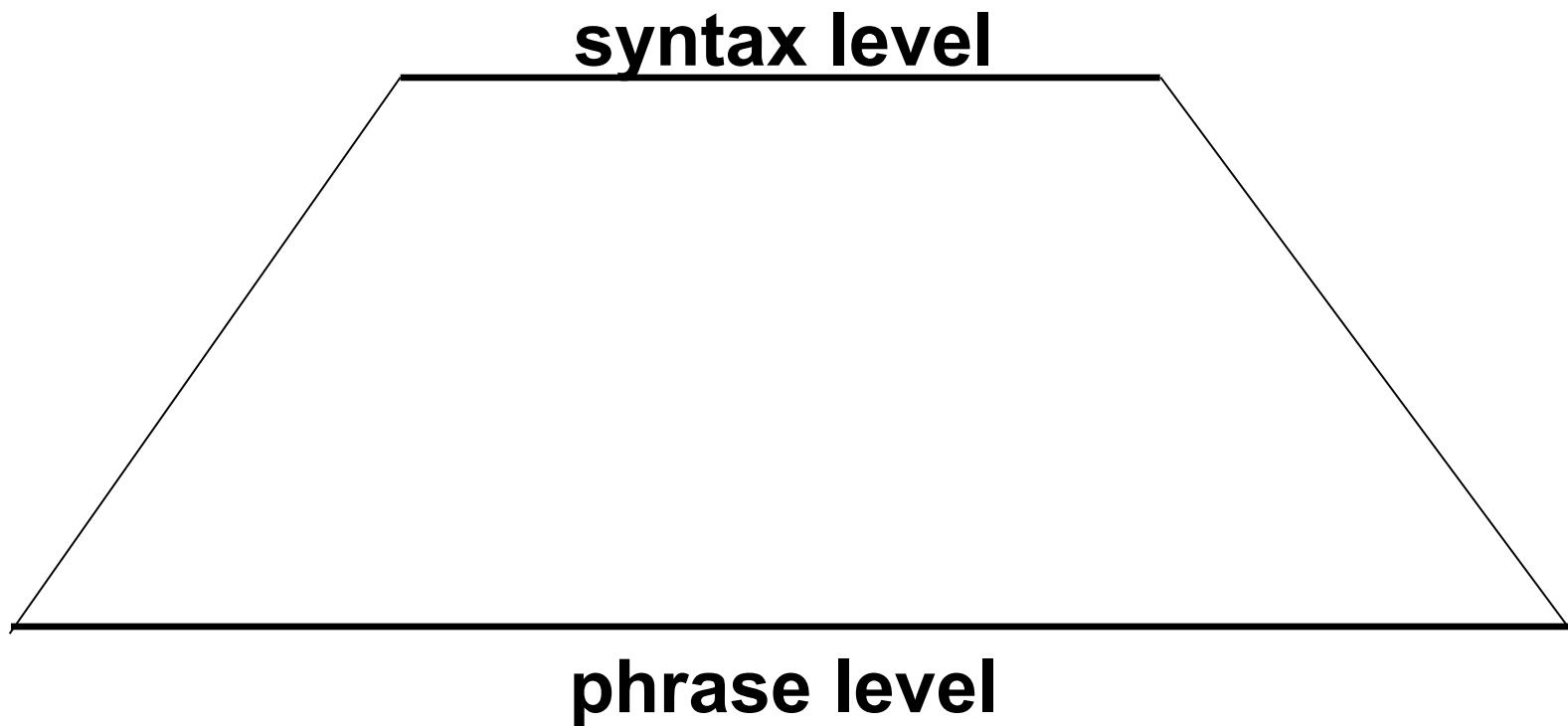
基于短语的统计翻译方法的问题

- 泛化能力差
 - 中国大使馆、美国大使馆 → 月球大使馆？
- 产生的句子不符合语法
 - 短语的简单组合，没有句法结构
- 无法表示不连续的短语搭配的翻译
 - 召开了一次关于…的会议 **hold a meeting on ...**
- 无法进行长距离的语序调整
- 解决办法：引入句法结构！

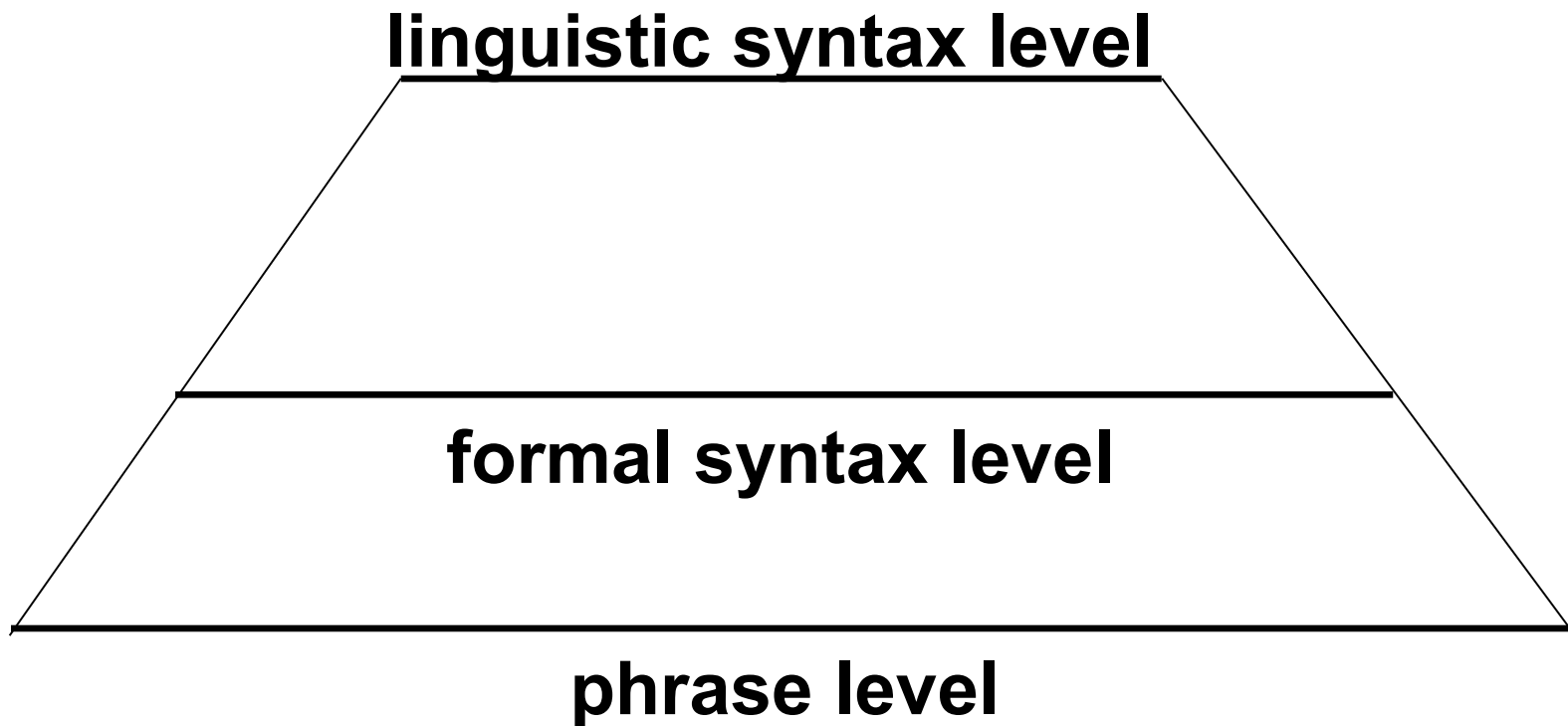
统计机器翻译方法的金字塔



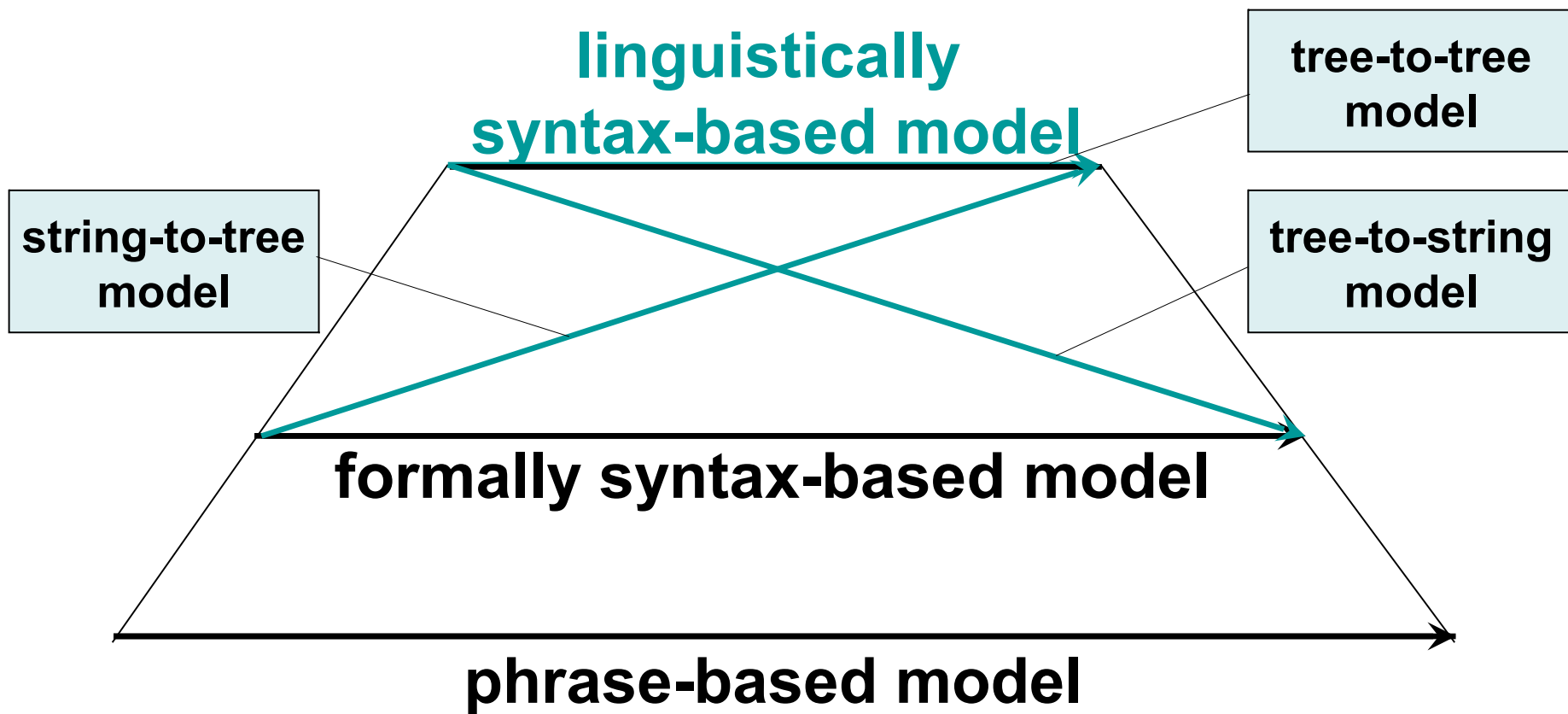
基于句法的统计机器翻译模型(1)



基于句法的统计机器翻译模型(1)



基于句法的统计机器翻译模型(1)



基于句法的统计机器翻译模型(2)

- 形式上基于句法的模型
 - 不使用任何语言学知识
 - 所有句法结构直接从未标注的语料库中自动学习得到
- 语言学上基于句法的模型
 - 使用语言学知识
 - 语言通常要从句法树库训练得到
 - 树到串模型：只在源语言端使用语言知识
 - 串到树模型：只在目标语言端使用语言知识
 - 树到树模型：在源语言端和目标语言端都使用语言知识

形式上基于句法的模型

- 反向转录语法（ITG）和括号转录语法（BTG）
Inversion (Bracketing) Transduction Grammar (ITG,BTG), Wu 1997
- 有限状态中心词转录机
Finite-State Head Transducer, Alshawi 2000
- 基于层次短语的翻译模型
Hierarchical Phrase-based Model, Chiang 2005
- 最大熵括号转录语法的翻译模型
Maximal Entropy Bracket Transduction Grammar (ME-BTG), Xiong 2006

语言学上基于句法的模型

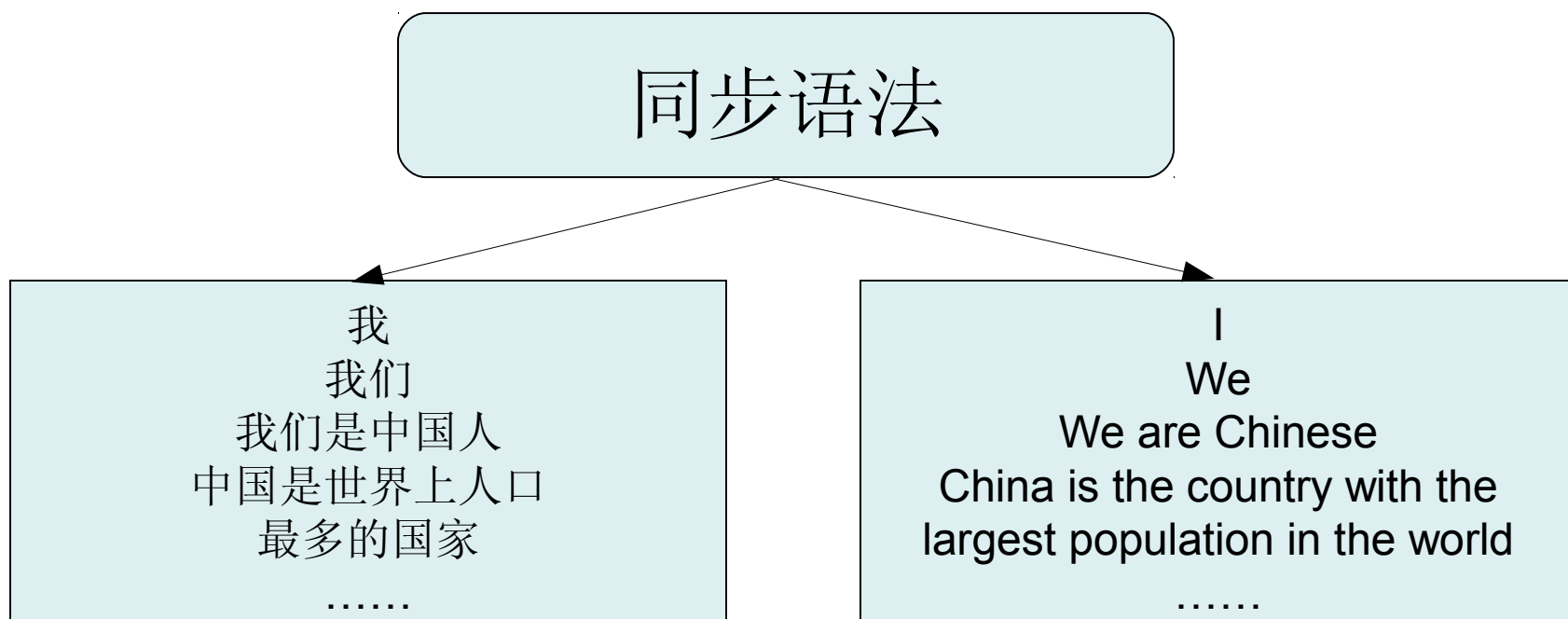
- 串到树模型 **String-to-Tree Model**
 - 美国南加州大学信息科学研究所（ISI/CSU）的工作
Yamada 2001, Galley 2006, Marcu 2006
- 树到串模型 **Tree-to-String Model**
 - 中科院计算所的工作
Tree-to-string Alignment Template Model (TAT),
Yang Liu ACL2006
 - 微软研究院的工作（依存模型）
Dependency Treelet Translation, Quirk 2005
- 树到树的模型 **Tree-to-Tree Model**

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

同步语法 (1)

- 定义：同步语法是一种形式语法，这种语法的每一次推导，都在两种或者两种以上语言中同步生成一个句子。



同步语法 (2)

- 同步语法的具体形式：
 - 同步上下文无关语法 (SCFG)
 - 反向转录语法 (ITG) 和括号转录语法 (BTG)
 - 同步树替换语法 (STSG)
 - 同步树粘接语法 (STAG)
 - 多文本语法 (MTG)
- 同步语法的应用：
 - 编译中的代码生成
 - 自然语言的语义解释
 - 自然语言的机器翻译
 - 双语语料库的对齐

同步语法 (2)

单语	双语	多语
乔姆斯基范式语法	反向转录语法	
上下文无关语法	同步上下文无关语法	多文本语法
树替换语法	同步树替换语法	
树粘接语法	同步树粘结语法	

同步语法 (3)

- 同步语法与统计机器翻译
 - 同步语法是很多基于句法的统计机器翻译模型的理论基础
 - 理论上说，如果采用同步语法，在完成源语言句法分析的同时，目标语言就生成了，因此可以利用各种成熟的句法分析算法进行机器翻译，而无需另外设计专门的翻译算法
 - 另一方面，采用同步语法对源语言进行句法分析时，要把目标语言的因素考虑进来，这不同于通常的句法分析

内容提要

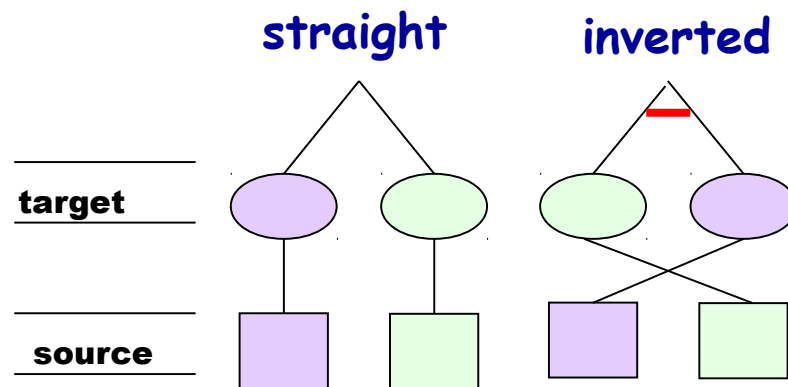
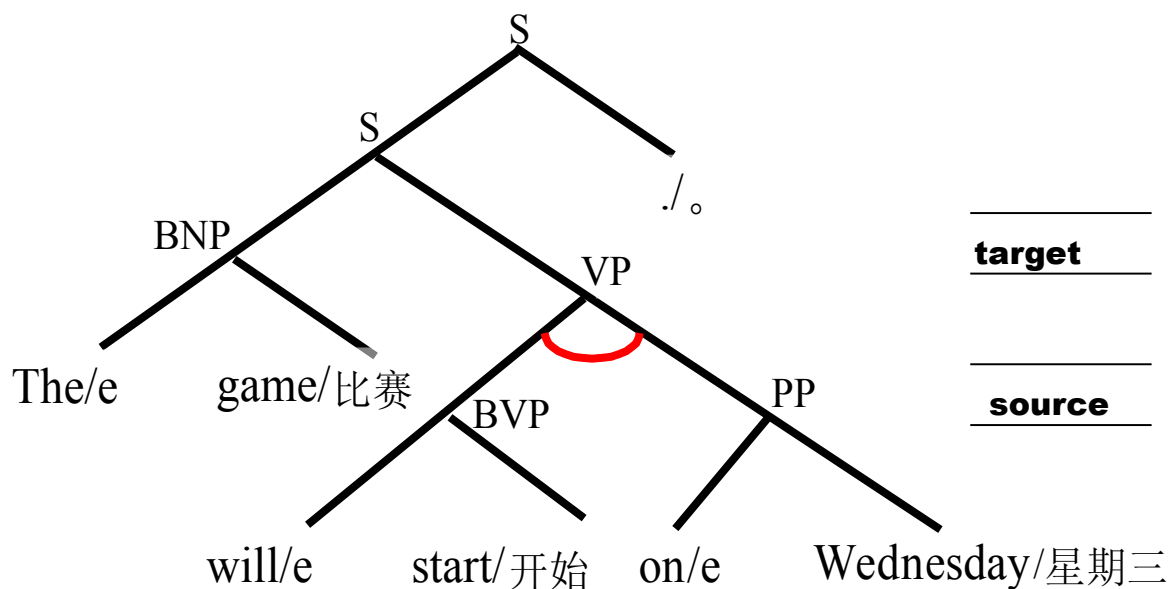
- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

反向转录语法

- Inversion Transduction Grammar (ITG)
吴德凯 (1997 onwards)
- ITG 是一种形式最简单的同步语法，可以并行地生成两颗对齐的句法树
 - ITG 的规则都是乔姆斯基范式形式的
 - 规则的右部或者全部是终结符，或者全部是非终结符
 - 非终结符规则都是二分的
 - ITG 的规则可以指定语序的变化：保序 或 逆序
 - ITG 中两种语言的规则使用同一套非终结符
- ITG 中对规则的二分限制降低了搜索的复杂度

反向转录语法

	ITG rules	Source	Target
非终结符规则	$A \rightarrow [B C]$	$A \rightarrow BC$	$A \rightarrow BC$
	$A \rightarrow < B C >$	$A \rightarrow BC$	$A \rightarrow CB$
终结符规则	$A \rightarrow x/y$	$A \rightarrow x$	$A \rightarrow y$



反向转录语法

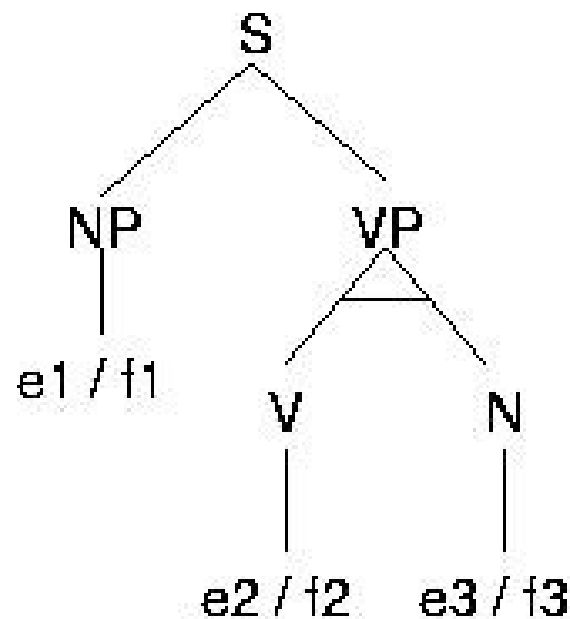
$S \rightarrow [NP VP]$

$NP \rightarrow e1 / f1$

$VP \rightarrow \langle N V \rangle$

$V \rightarrow e2 / f2$

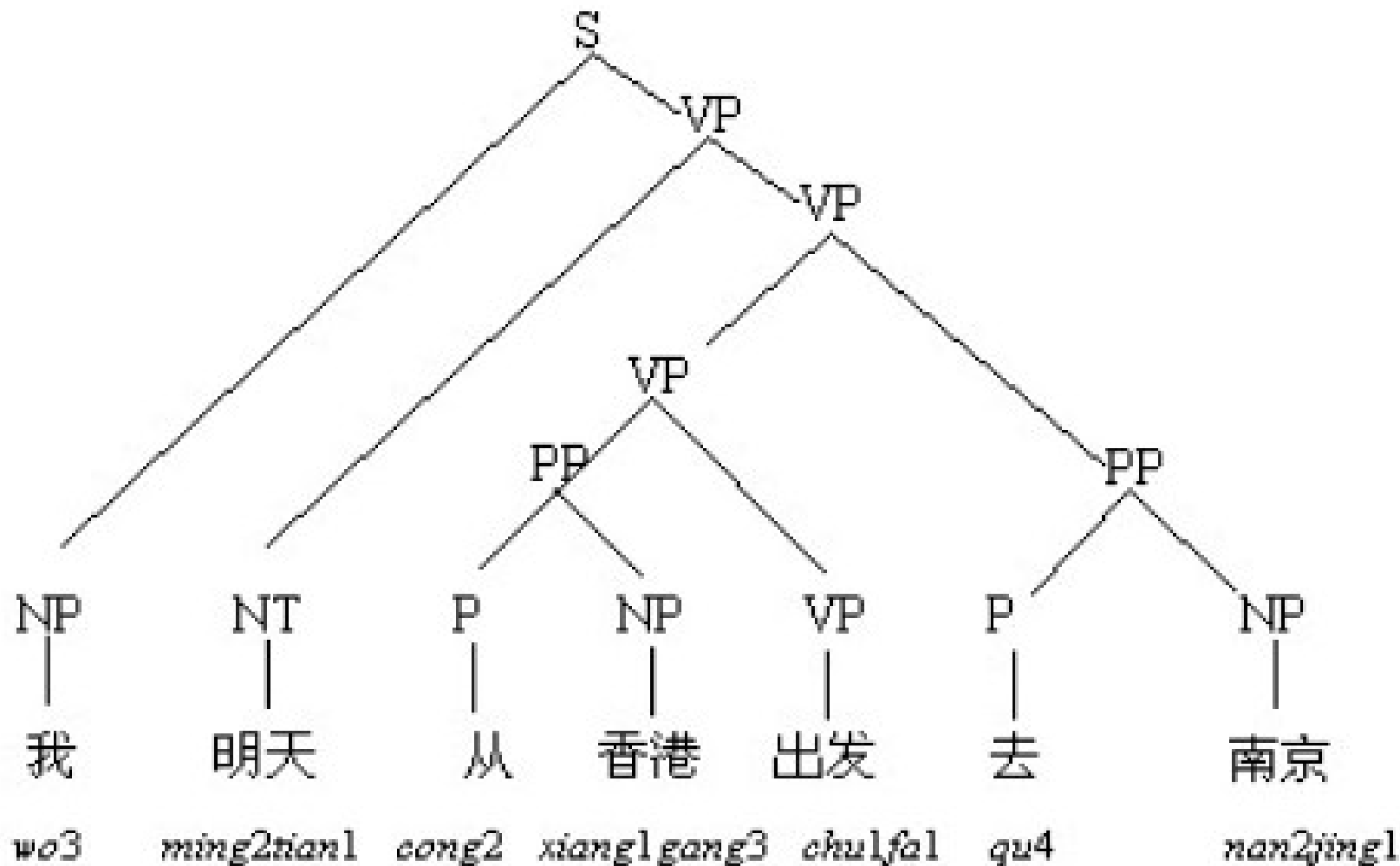
$N \rightarrow e3 / f3$



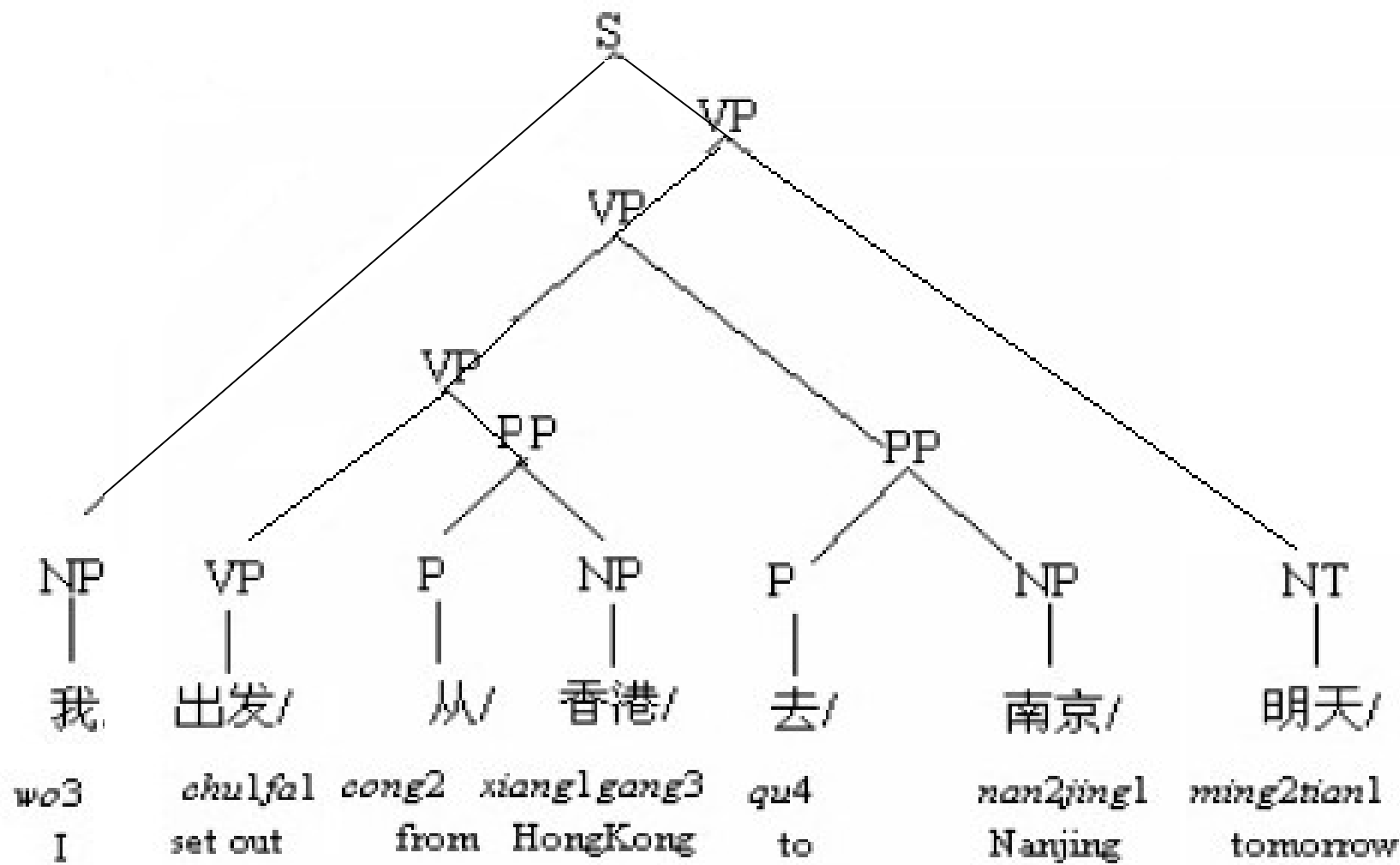
基于反向转录语法的统计机器翻译(1)

- 训练：从词语对齐的语料库中自动抽取规则
- 解码：类似于一个概率化句法分析的过程
 - 利用规则的源语言部分进行句法分析
 - 存在源语言部分相同而目标语言部分不同的规则（保序或逆序），这是不同于传统句法分析的地方
 - 句法分析时，对于源语言部分相同而目标语言部分不同的规则，需要通过概率计算进行评分，这相当于对译文语序进行选择
 - 句法分析完成的同时也就生成了译文句法结构和译文句子

基于反向转录语法的统计机器翻译(2)



基于反向转录语法的统计机器翻译(3)



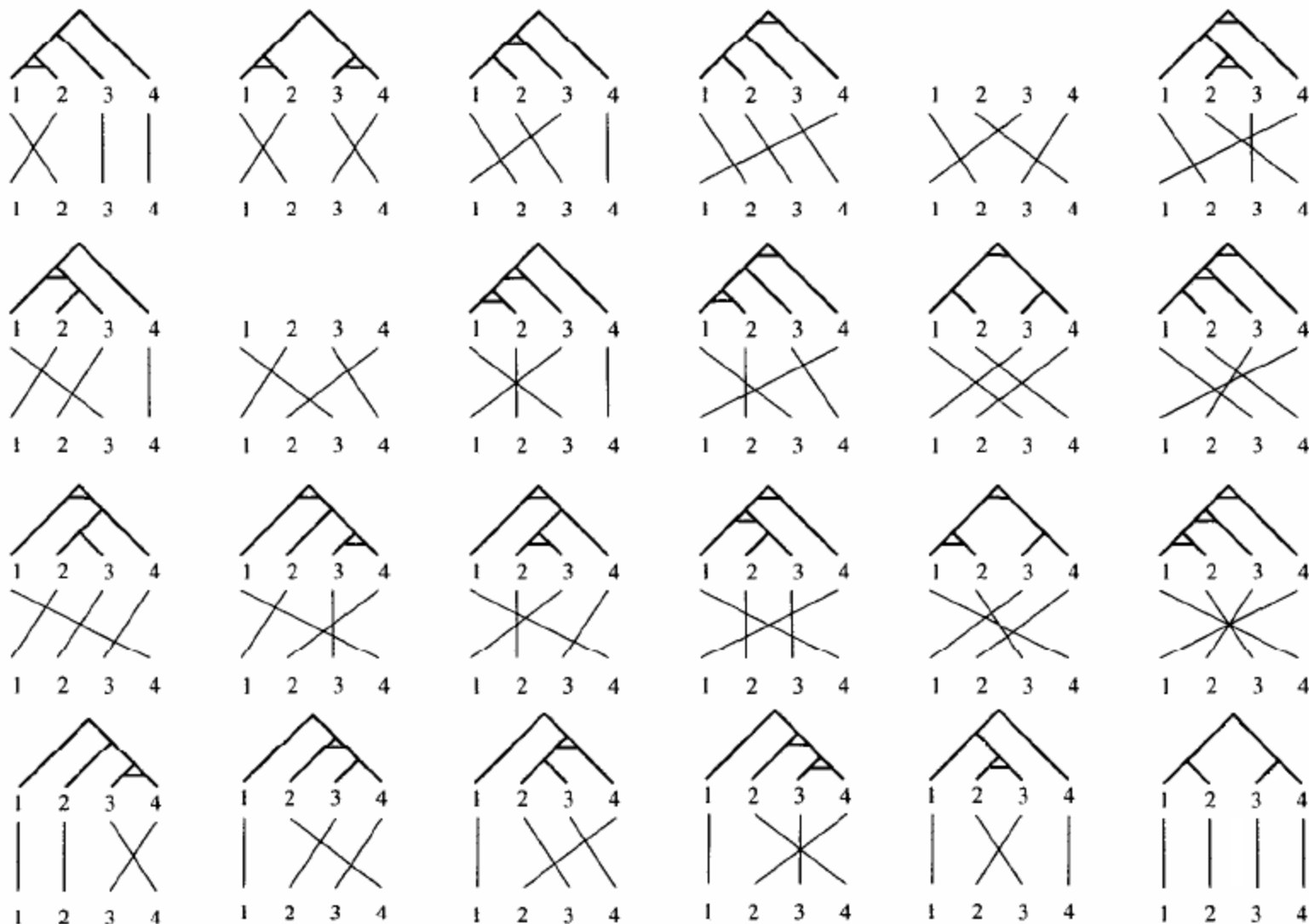
括号转录语法

- Bracketing Transduction Grammar: BTG
- BTG 是 ITG 的一个特例，其中只有唯一的一个非终结符 X
- 可以这么理解：BTG 仅仅给出了两种语言的句子结构结构之间的对应关系，没有任何句法标记信息（如 NP、VP 等等）

统计机器翻译中语序调整的方式

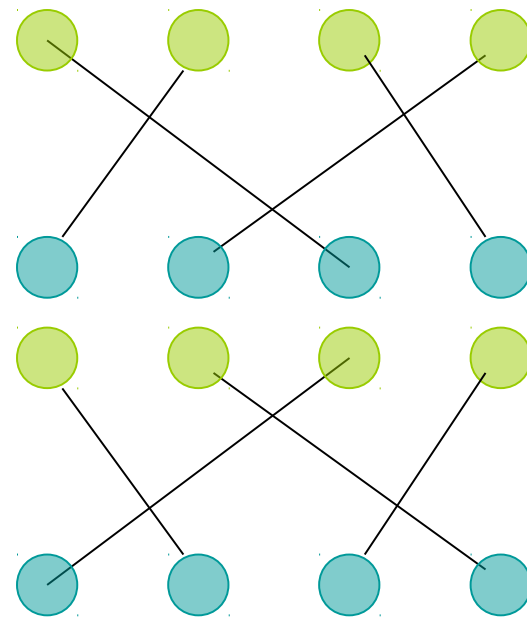
- 无约束（所有匹配都运行）
 - 所有语序调整都是允许的
 - 对于 N 个词（或短语），在 **IBM** 约束下，语序调整有 $N!$ 种可能性，搜索空间随着句子长度呈指数级增长，因此其搜索问题是 **NP** 问题
- **IBM 约束（IBM Constrains）**
 - 为了减少搜索空间，通常在从左到右的解码过程中都会采用 **IBM** 约束来限制语序调整的搜索空间，也就是说，每次只选择最左边若干个未被翻译的词语进行翻译（对 **Hypothesis** 进行扩展）
 - **IBM** 约束可以大大减少搜索空间，但依然存在大量非法语序调整
- **BTG 约束（BTG Constrains）**
 - 只有能够满足某种 **BTG** 映射的语序调整才是允许的
 - **BTG** 约束大大降低了搜索空间大小，确保搜索范围内的语序调整都满足语法约束，同时不在搜索范围内的约束都不满足语法约束
 - **BTG** 约束搜索使得长距离语序调整成为可能

这里给出了四个词的所有可能的调序方案以及对应的 **BTG** 转换模式。
 其中有两种方案在 **BTG** 约束下是不允许的（找不到对应的 **BTG** 转换模式）



BTG 约束导致搜索空间大大压缩

f	BTG	all matchings	ratio
0	1	1	1.000
1	1	1	1.000
2	2	2	1.000
3	6	6	1.000
4	22	24	0.917
5	90	120	0.750
6	394	720	0.547
7	1806	5040	0.358
8	8558	40320	0.212
9	41586	362880	0.115
10	206098	3628800	0.057
11	1037718	39916800	0.026
12	5293446	479001600	0.011
13	27297738	6227020800	0.004
14	142078746	87178291200	0.002
15	745387038	1307674368000	0.001
16	3937603038	20922789888000	0.000



**word reordering
which are not
permitted in BTG**

真实自然语言的翻译满足 BTG 约束吗？

对于汉语和英语之间的翻译，几乎满足一个例外（[出处？](#)）：



对于一些自由语序的语言，不一定满足

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

基于最大熵括号转录语法的翻译模型

- 基于最大熵括号转录语法的翻译模型
A Translation Model Based on Maximum Entropy Bracketing Transuction Grammar (ME-BTG)
- Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. COLING-ACL 2006, Sydney, Australia, July 17-21.
- Deyi Xiong, Min Zhang, Ai Ti Aw, Haitao Mi, Qun Liu and Shouxun Lin, Refinements in BTG-based Statistical Machine Translation, IJCNLP 2008, Hyderabad, India, January 7-12

BTG 的主要问题

- 两条主要合并规则

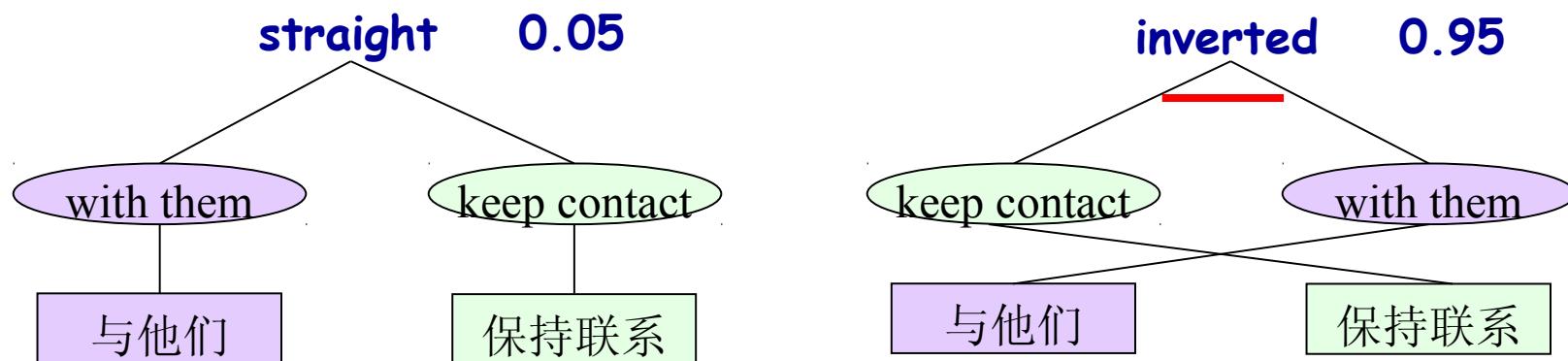
$$A \rightarrow [AA] \quad 0.8$$

$$A \rightarrow \langle AA \rangle \quad 0.2$$

- 如何使用这两条规则， **stochastic BTG** 给每条规则赋以先验概率
- 先验概率是一种非常粗糙、简单的处理方法，不能有效地处理重排序问题

ME-BTG: 基本思想

- 在 **BTG** 框架下，将重排序问题看作是一个2类分类问题：
 - 条件：各种与重排序短语相关的特征
 - 类别：相邻语块的顺序 { **straight**, **inverted** }
- 引入最大熵模型作为分类模型，根据实际上下文计算合并规则的概率



ME-BTG 模型

- 模型

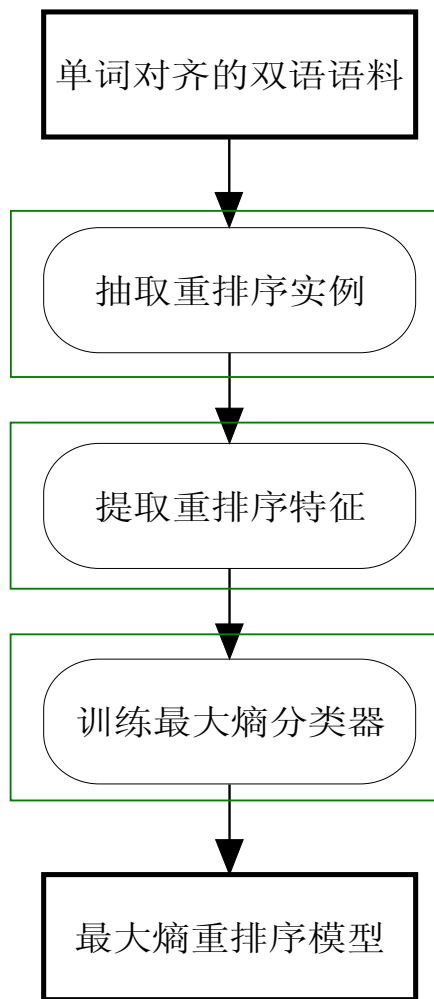
$$\Omega = p_{\theta}(o|A^1, A^2) = \frac{\sum_i \theta_i h_i(o, A^1, A^2)}{\sum_{o'} \exp(\sum_i \theta_i h_i(o', A^1, A^2))}$$

- 特征

$$h_i(o, A^1, A^2) = \begin{cases} 1 & \text{if } f(A^1, A^2) = T, o = O \\ 0 & \text{otherwise} \end{cases}$$

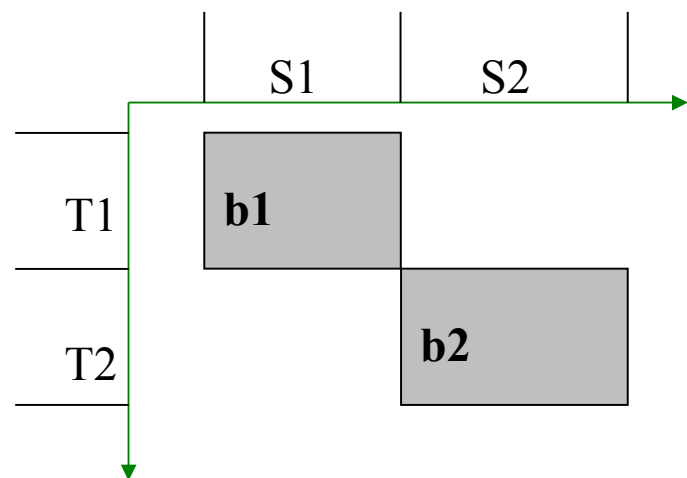
$$O \in \{straight, inverted\}$$

ME-BTG 训练



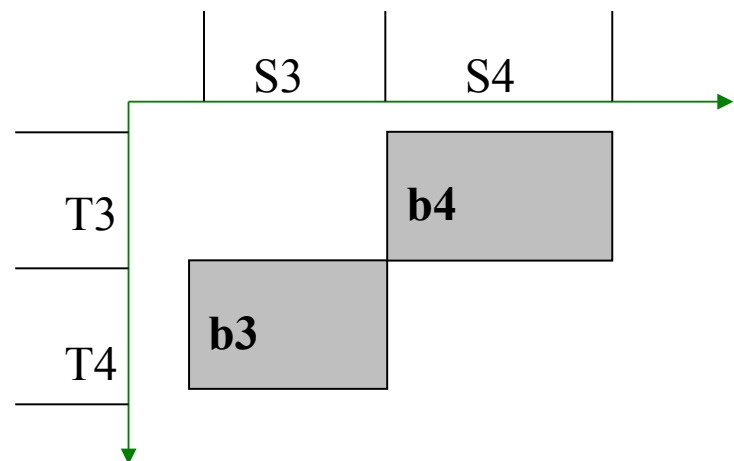
抽取重排序实例

在双语语料库中抽取所有如下两类
双语短语块：



$\langle b1; b2 \rangle \rightarrow \text{STRAIGHT}$

E.g. $\langle \text{今天 有 棒球 比赛} \mid \text{Are there any baseball games today; 吗 ?} \mid ? \rangle \rightarrow \text{STRAIGHT}$



$\langle b3; b4 \rangle \rightarrow \text{INVERTED}$

E.g. $\langle \text{澳门 政府} \mid \text{the Macao government; 有关 部门} \mid \text{related departments of} \rangle \rightarrow \text{INVERTED}$

重排序特征

- 单目特征：单个源/目标语言边界单词
- 双目特征：两个源/目标语言边界单词的组合

< 与 他们 | with them; 保持联系 | keep contact > → INVERTED

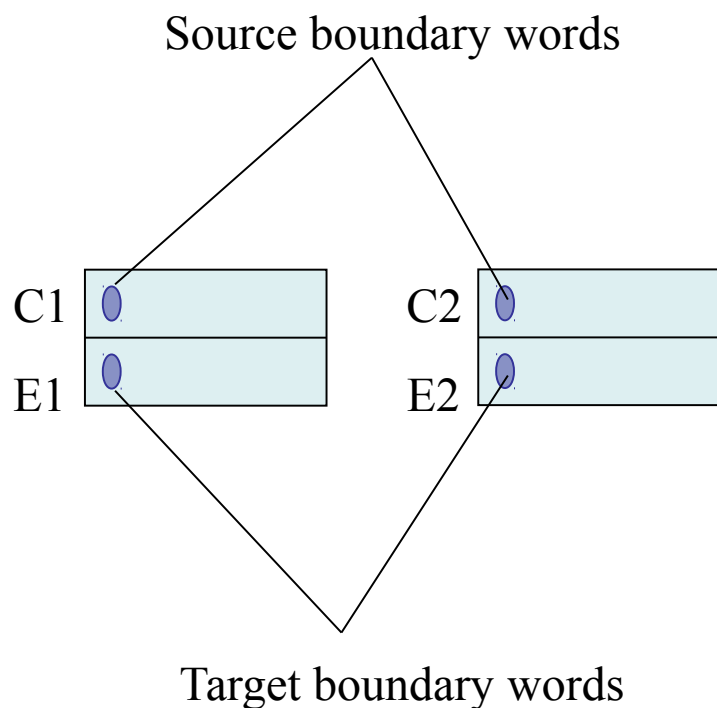


特征选择

$$h_{mono}(o, A^1, A^2) = \begin{cases} 1 & \text{if } A^2.t_1 = \text{keep}, o = \text{inverted} \\ 0 & \text{otherwise} \end{cases}$$

$$h_{bino}(o, A^1, A^2) = \begin{cases} 1 & \text{if } A^1.t_1 = \text{with}, A^2.t_1 = \text{keep}, o = \text{inverted} \\ 0 & \text{otherwise} \end{cases}$$

为什么使用边界单词作为特征？



feature	IGR
Phrases	.02655
C1C2E1E2	.0263687
E1E2	.0239286
C1C2	.023363
C2E2	.0192932
C1E1	.0153117
C2	.011371
E2	.00994372
E1	.00899752
C1	.00758598

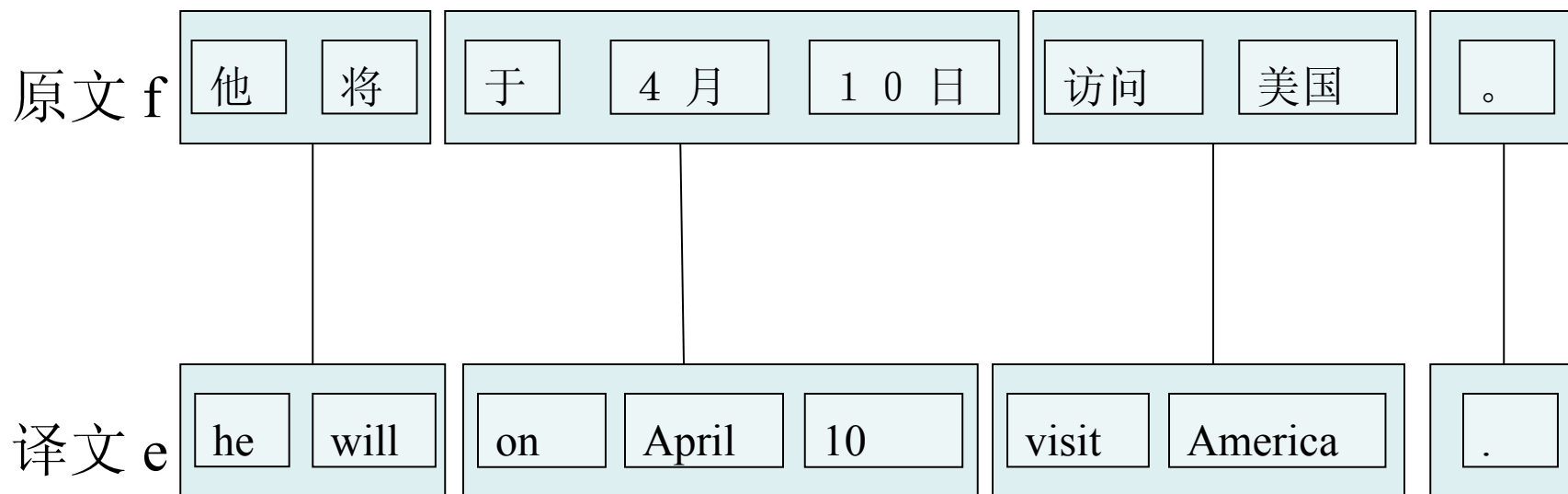
基于 ME-BTG 的统计机器翻译系统

- Bruin: 基于 ME-BTG 的统计机器翻译系统
- 解码算法
 - 基于 CKY 算法
 - 自底向上, 考虑每一个区间 (i,j) , 每个区间保留一个堆栈
 - 对于每个区间 (i,j) , 考虑其每一个分割 $(i,k)*(k+1,j)$
 - 对于每一个分割, 考虑其所有子节点的候选译文, 以及“保序”和“逆序”两种情况, 计算所有可能的候选译文
 - 采用柱搜索 (Beam Search) 策略, 对堆栈中的候选译文结点进行剪枝
 - 对于堆栈中的候选译文结点进行归并 (recombination): 如果结点的左右 $n-1$ 个单词都相同, 在归并为一个结点 (假设这里采用 n 元语法模型)

CKY 解码算法

基于 ME-BTG 模型的翻译过程

查短语表



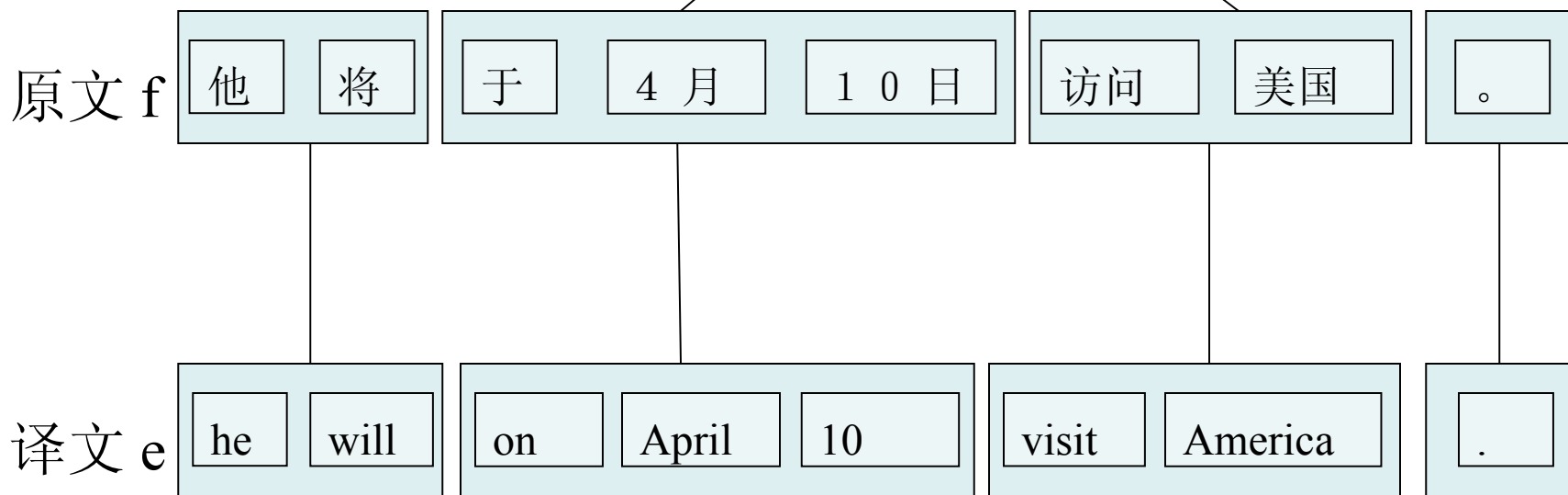
基于 ME-BTG 模型的翻译过程

利用边界词特征计算是否调序

原文边界词：“于” + “访问”

保序概率：0.05

译文边界词：“on” + “visit”



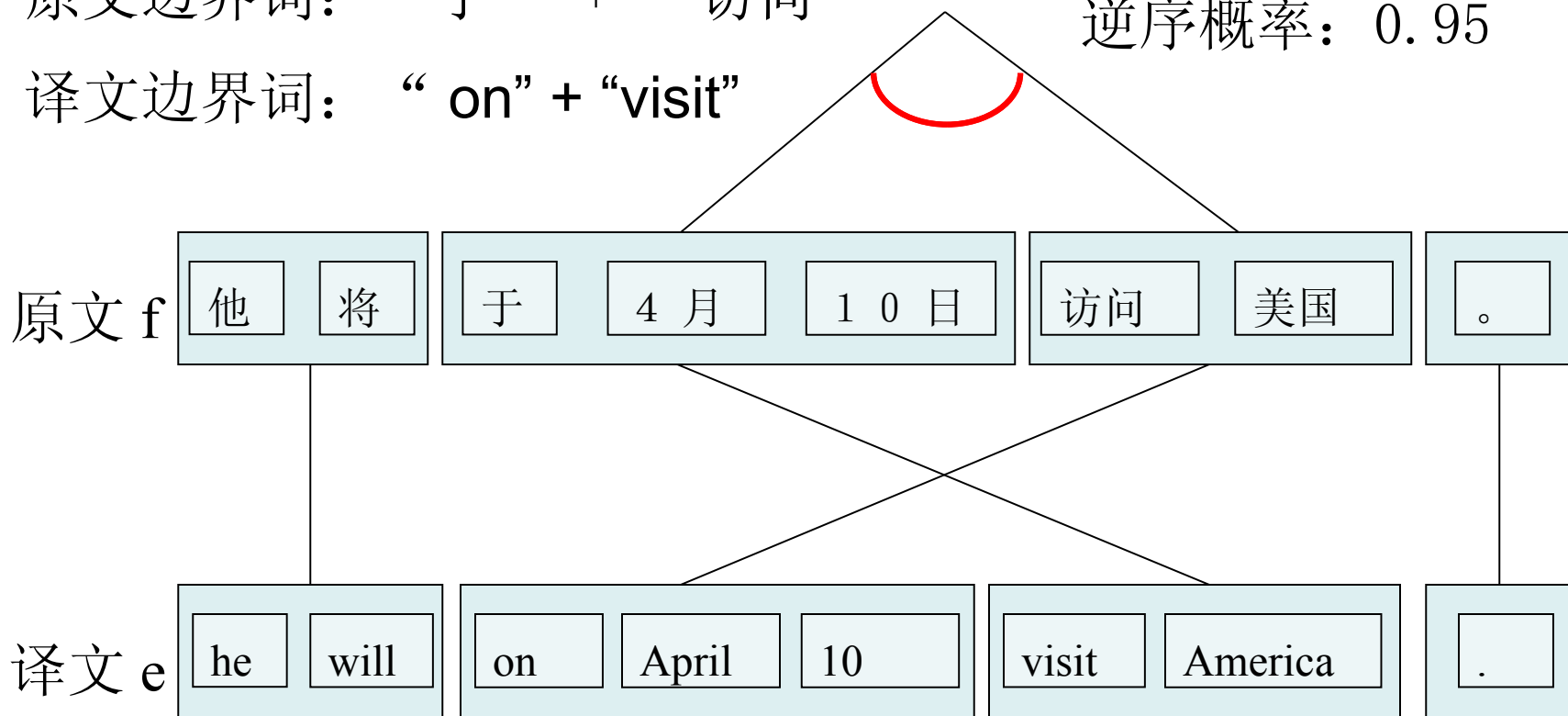
基于 ME-BTG 模型的翻译过程

利用边界词特征计算是否调序

原文边界词：“于” + “访问”

译文边界词：“on” + “visit”

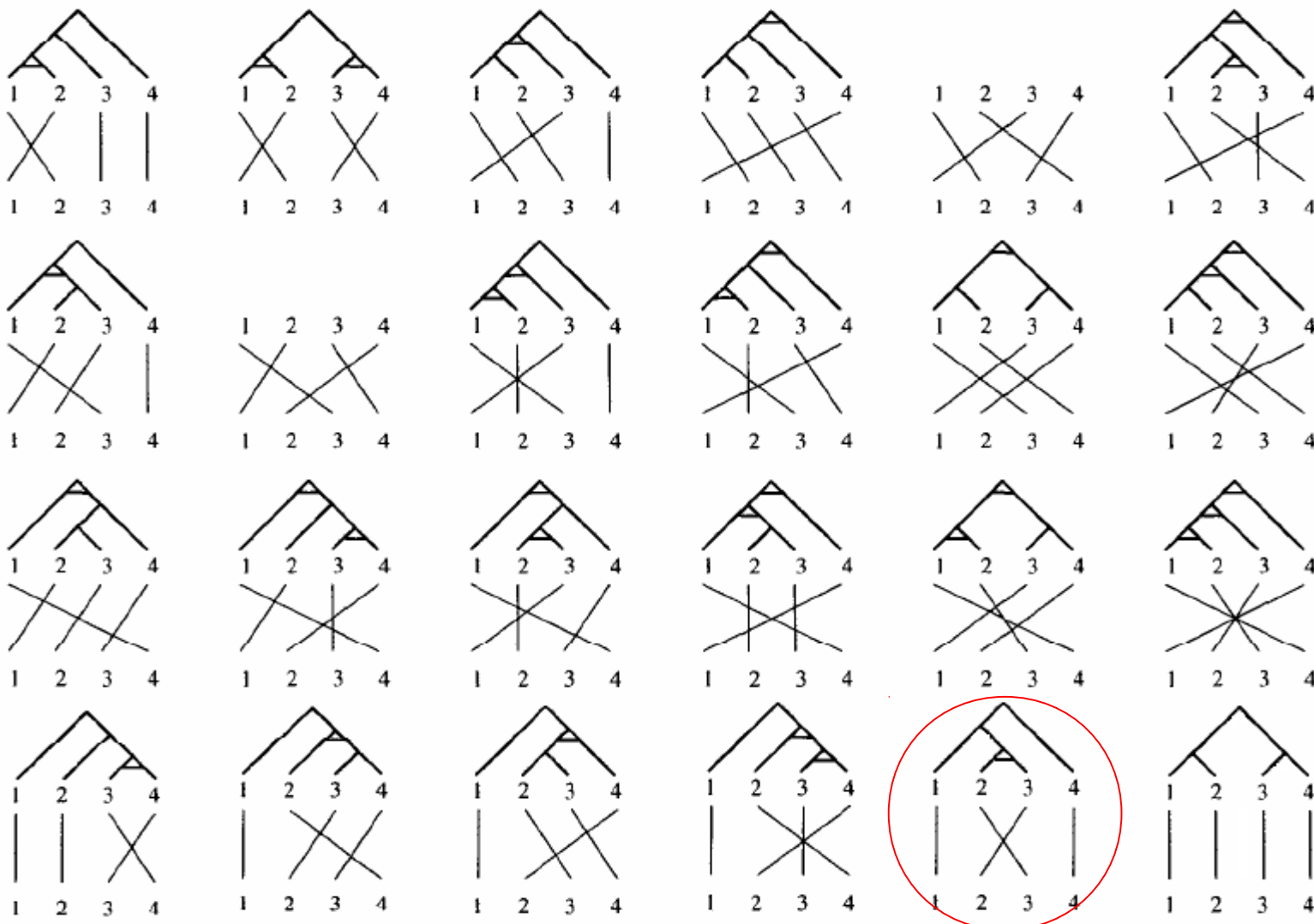
逆序概率：0.95



基于 ME-BTG 模型的翻译过程

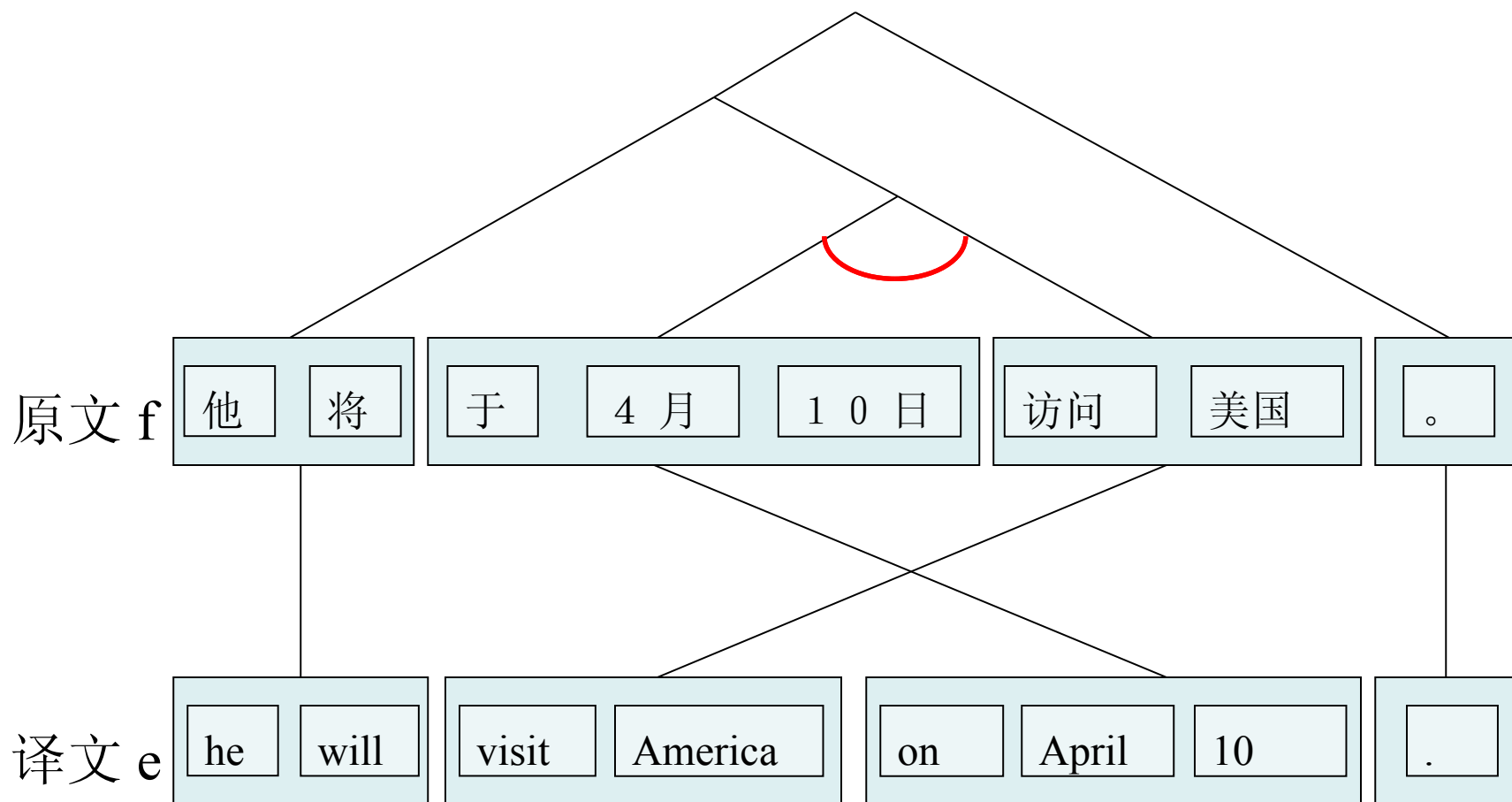
搜索

在所有可能的结构变换中搜索概率最大的形式



基于 ME-BTG 模型的翻译过程

得到结果



ME-BTG: 实验

Systems	NIST MT 05	IWSLT 04
Bruin with monotone search	20.1	37.8
Bruin with distance-based reordering	20.9	38.8
Bruin with flat reordering	20.5	38.7
Pharaoh	20.8	38.9
Bruin with MEBTG (单目)	22.0	42.4
Bruin with MEBTG (单目+双目)	22.2	42.8

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

同步上下文无关语法

- David Chiang. An introduction to synchronous grammars. In Proc. of ACL Tutorial, 2006.

本部分讲义引自 David Chiang 的上述 Tutorial 中的内容，特此说明，并向原作者表示感谢。

同步上下文无关语法 (1)

- 英语的语法:

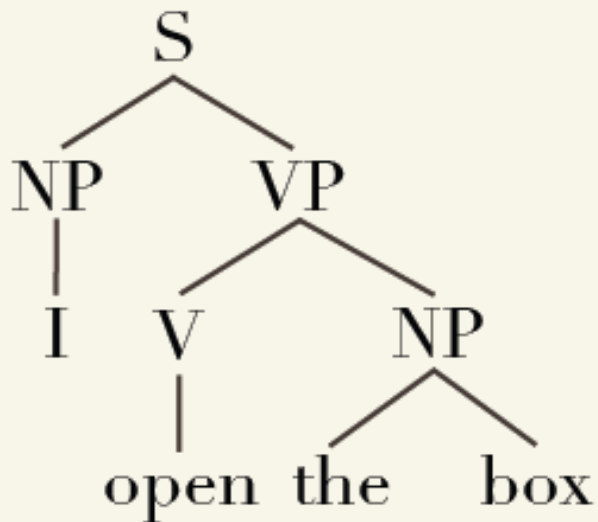
$S \rightarrow NP VP$

$NP \rightarrow I$

$NP \rightarrow \text{the box}$

$VP \rightarrow V NP$

$V \rightarrow \text{open}$



同步上下文无关语法 (2)

- 日语的语法

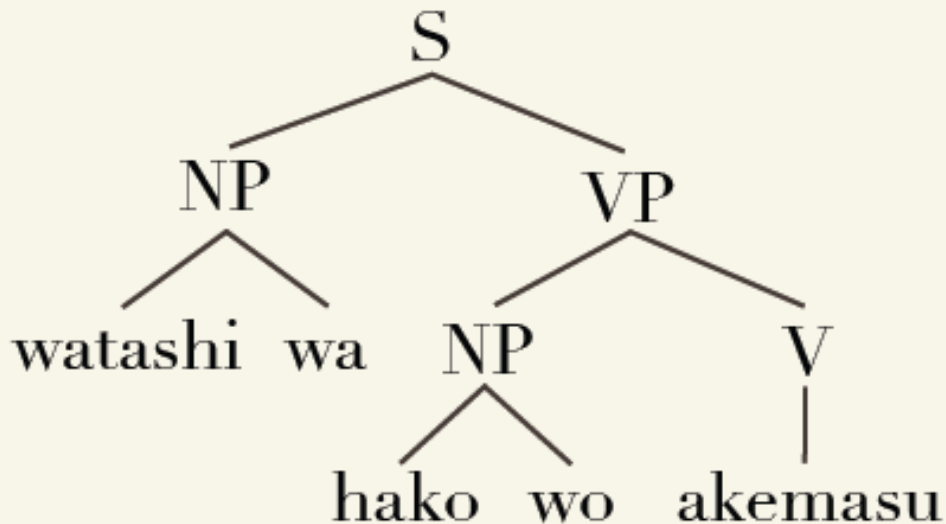
$S \rightarrow NP VP$

$NP \rightarrow watashi wa$

$NP \rightarrow hako wo$

$VP \rightarrow NP V$

$V \rightarrow akemasu$



同步上下文无关语法 (3)

- 两种语法的一一对应关系:

$S \rightarrow NP \ VP$	$S \rightarrow NP \ VP$
$NP \rightarrow I$	$NP \rightarrow watashi \ wa$
$NP \rightarrow the \ box$	$NP \rightarrow hako \ wo$
$VP \rightarrow V \ NP$	$VP \rightarrow NP \ V$
$V \rightarrow open$	$V \rightarrow akemasu$

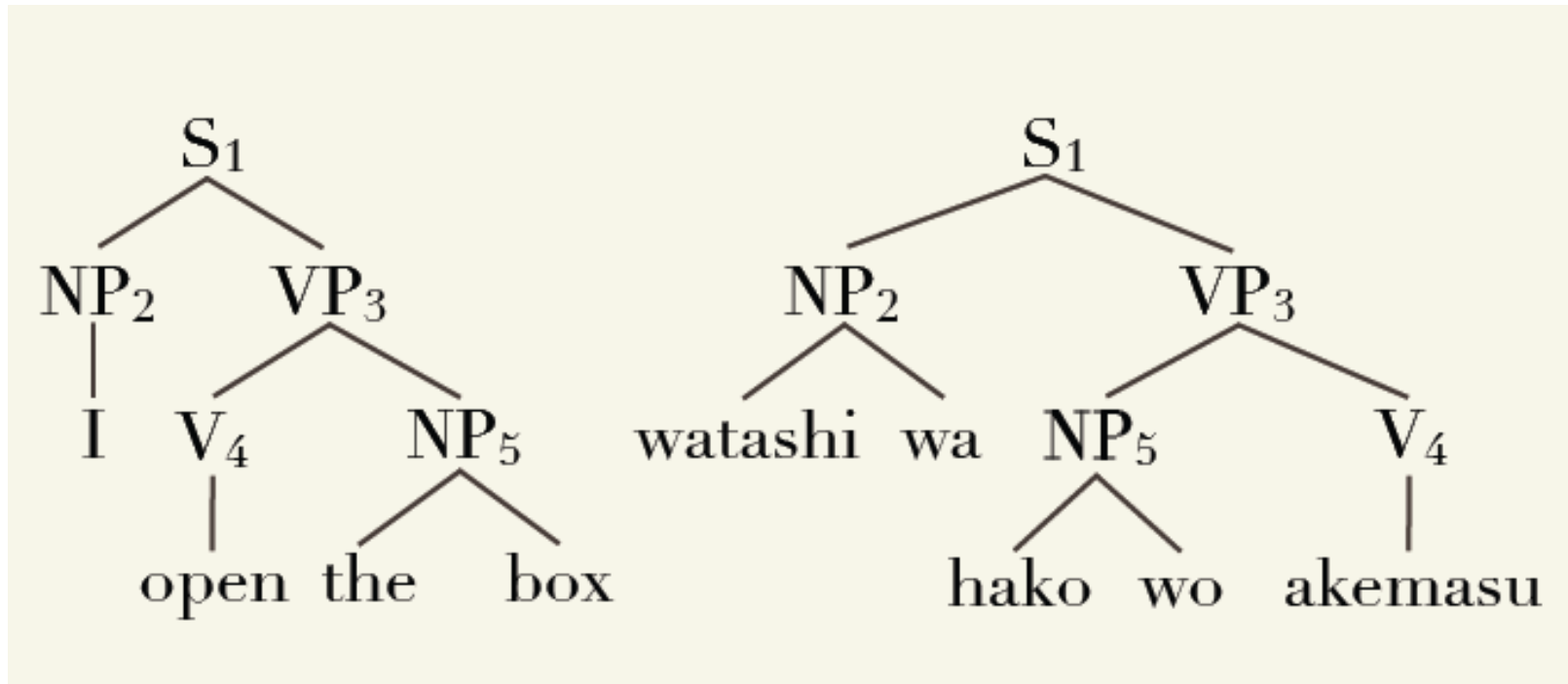
同步上下文无关语法 (4)

- 同步上下文无关语法:

$S \rightarrow NP_1 VP_2, NP_1 VP_2$
 $NP \rightarrow I, watashi\ wa$
 $NP \rightarrow the\ box, hako\ wo$
 $VP \rightarrow V_1 NP_2, NP_2 V_1$
 $V \rightarrow open, akemasu$

同步上下文无关语法 (5)

- 同步句法树:



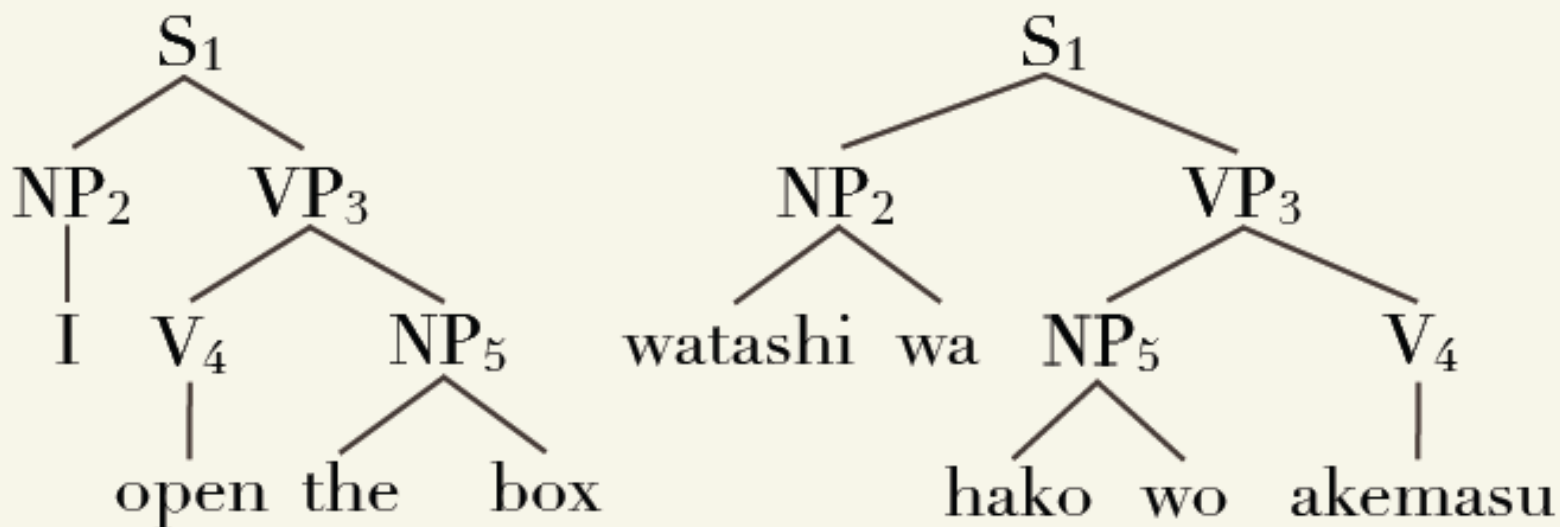
同步上下文无关语法 (6)

- 带概率的同步语法:

$\overset{0.3}{S} \rightarrow NP_1 VP_2, NP_1 VP_2$
 $\overset{0.1}{NP} \rightarrow I, watashi wa$
 $\overset{0.6}{NP} \rightarrow the\ box, hako wo$
 $\overset{0.5}{VP} \rightarrow V_1 NP_2, NP_2 V_1$
 $\overset{0.2}{V} \rightarrow open, akemasu$

同步上下文无关语法 (7)

- 同步句法树的概率:



Derivation probability: $0.3 \times 0.1 \times 0.5 \times 0.6 \times 0.2$

类似的表示形式

$$VP \rightarrow (V_1 NP_2, NP_2 V_1)$$

Syntax directed translation
schema (Aho and Ullman;
Lewis and Stearns)

$$(VP \rightarrow V_1 NP_2, VP \rightarrow NP_2 V_1)$$

$$VP \rightarrow \langle V NP \rangle$$

Inversion transduction
grammar (Wu)

$$VP \rightarrow \bowtie \begin{matrix} [1,2] \\ [2,1] \end{matrix} \left(\begin{matrix} V & NP \\ V & NP \end{matrix} \right)$$

Multitext grammar (Melamed)

同步上下文关语法的层次(1)

- 乔姆斯基范式:

$$X \rightarrow YZ$$

$$X \rightarrow a$$

同步上下文关语法的层次(2)

- 5 阶 \rightarrow 2阶

$A \rightarrow [[[B \ C] \ D] \ E] \ F$ rank 5

$A \rightarrow V1 \ F$

$V1 \rightarrow V2 \ E$

$V2 \rightarrow V3 \ D$

$V3 \rightarrow B \ C$

rank 2

同步上下文关语法的层次(3)

- 3 阶 \rightarrow 2阶:

$$A \rightarrow (B_1 [C_2 D_3], [C_2 D_3] B_1) \quad \text{rank 3}$$

$$\begin{aligned} A &\rightarrow (B_1 V1_2, V1_2 B_1) \\ V1 &\rightarrow (C_1 D_2, C_1 D_2) \end{aligned} \quad \text{rank 2}$$

同步上下文关语法的层次(4)

· 4 阶 \rightarrow 2阶?

$$A \rightarrow (B_1 C_2 D_3 E_4, C_2 E_4 B_1 D_3) \quad \text{rank 4}$$

$$A \rightarrow ([B_1 C_2] D_3 E_4, [C_2 \textcolor{red}{E}_4 B_1] D_3)$$

$$A \rightarrow (B_1 [C_2 D_3] E_4, [C_2 \textcolor{red}{E}_4 \textcolor{red}{B}_1 D_3])$$

$$A \rightarrow (B_1 C_2 [D_3 E_4], C_2 [E_4 \textcolor{red}{B}_1 D_3])$$

同步上下文关语法的层次(5)

- 4 阶 \rightarrow 2阶?

$$A \rightarrow (B_1 \ C_2 \ D_3, C_2 \ D_3 \ B_1)$$

	1	2	3
1			B
2	C		
3		D	

$$A \rightarrow (B_1 \ C_2 \ D_3 \ E_4, C_2 \ E_4 \ B_1 \ D_3)$$

	1	2	3	4
1			B	
2	C			
3				D
4		E		

同步上下文关语法的层次(5)

- 表达能力:

$$1\text{-CFG} \subsetneq 2\text{-CFG} = 3\text{-CFG} = 4\text{-CFG} = \dots$$

$$1\text{-SCFG} \subsetneq 2\text{-SCFG} = 3\text{-SCFG} \subsetneq 4\text{-SCFG} \subsetneq \dots$$

$$\begin{array}{c} \cong \quad \cong \\ \text{ITG} \\ (\text{Wu}, 1997) \end{array}$$

算法复杂度

- 机器翻译复杂度：
 - 分析: $O(n^3)$
 - 转换: $O(n)$
 - 生成: $O(n)$
- 同步句法分析复杂度：
 - $O(n^{10})$

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

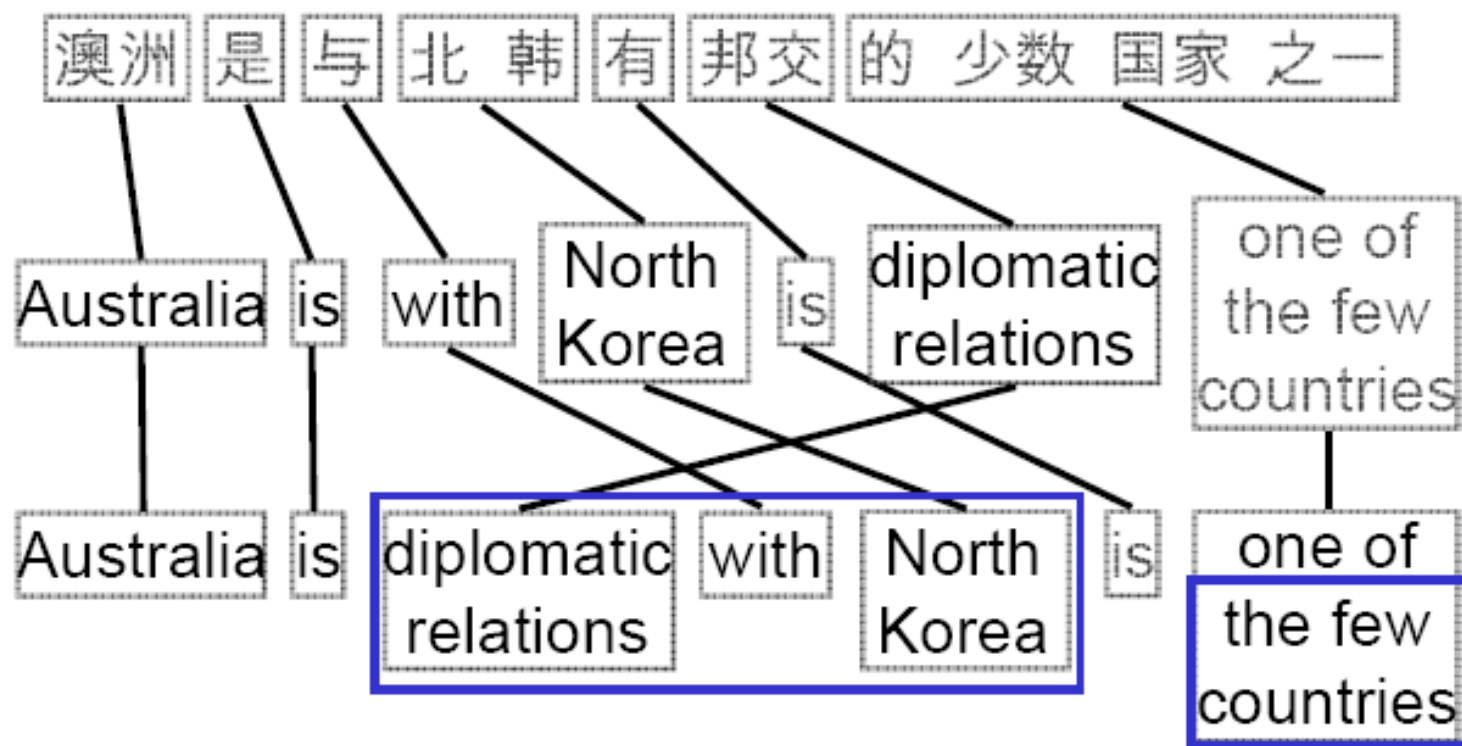
层次短语模型 (1)

- 层次化基于短语的翻译模型（蒋伟， UMD）
Hierarchical Phrase-Based Translation Model
- David Chiang. A Hierarchical Phrase-Based Model for Statistical Machine Translation. ACL2005. (Best Paper Award)
- 本讲义这一部分内容直接引用了以下讲义的部分内容，特此说明并向原作者表示感谢：
 - David Chiang, Hiero: Finding Structure in Statistical Machine Translation, in National University of Singapore

层次短语模型 (2)

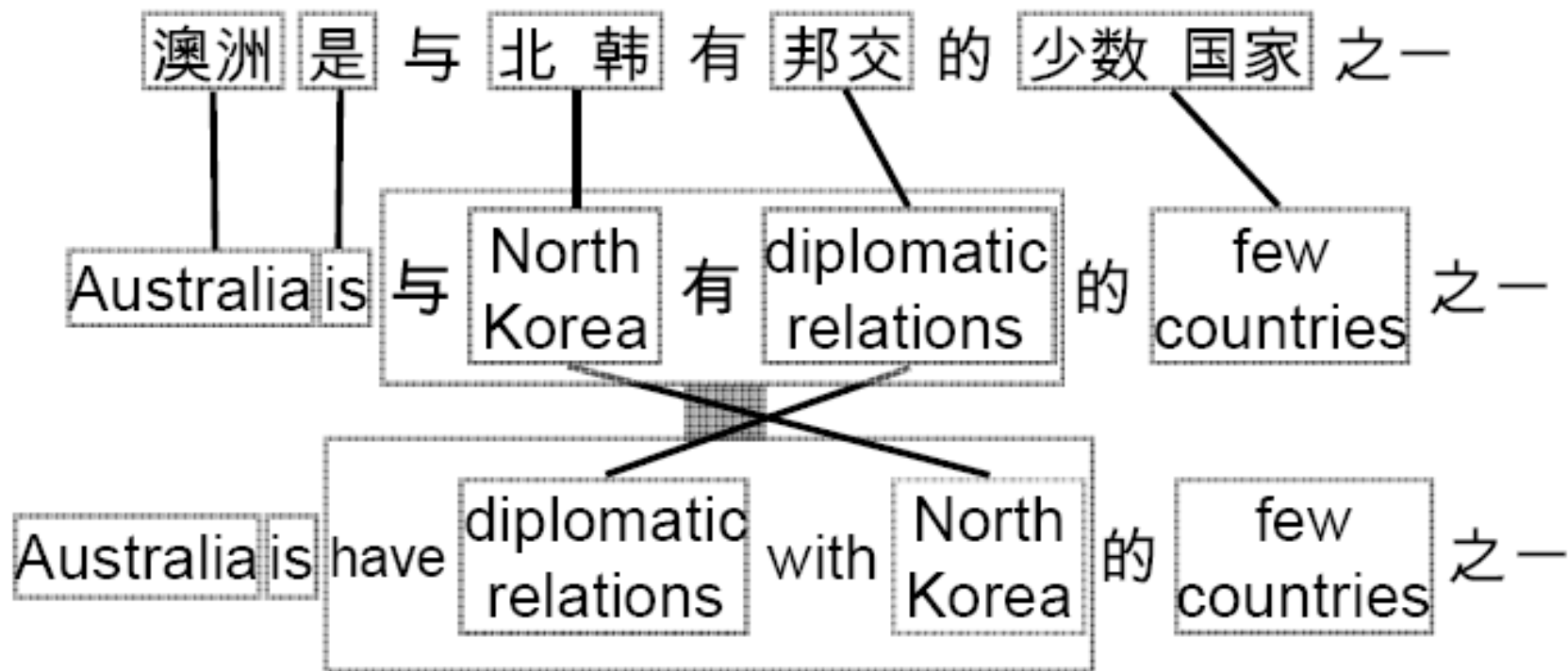
- 传统的基于短语的翻译模型中，短语是平面的，不能嵌套
- 在层次短语模型中，引入了嵌套的层次短语
- 采用平行上下文无关语法作为理论基础，但只使用唯一的非终结符标记
- 效果比传统的短语模型有很大提高

观察：短语的层次性

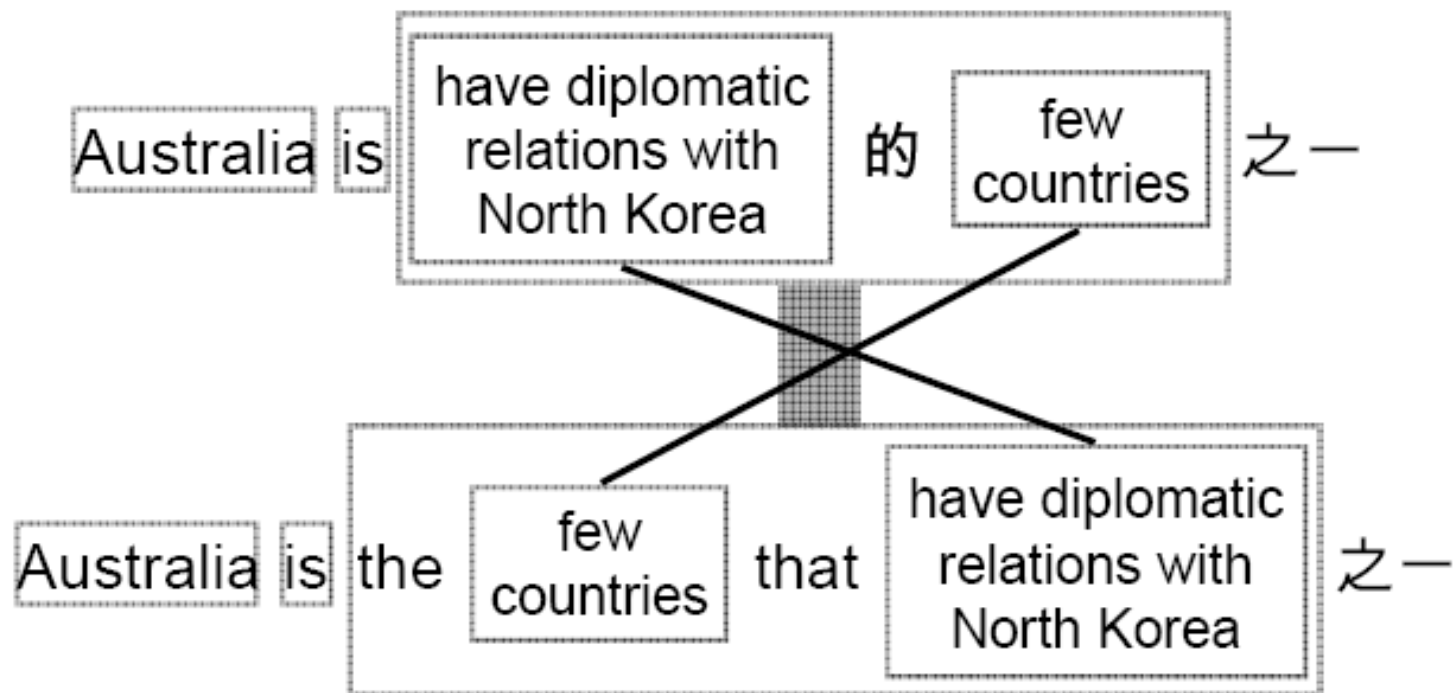


可以观察到短语是有层次的，短语之间可以嵌套。

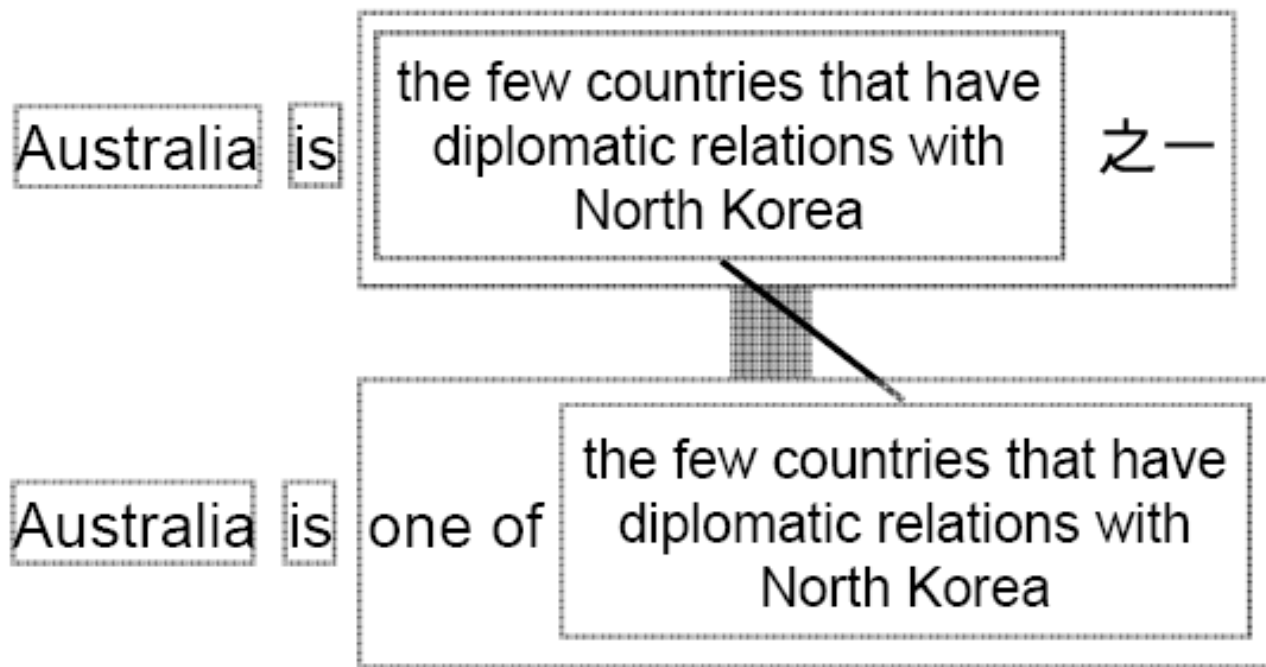
用层次短语进行翻译 (1)



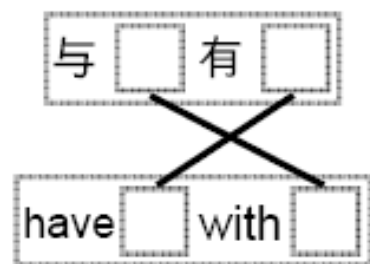
用层次短语进行翻译 (2)



用层次短语进行翻译 (3)



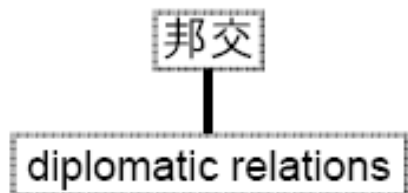
用同步语法表示层次短语 (1)



$(X \rightarrow \text{与 } X_1 \text{ 有 } X_2, X \rightarrow \text{have } X_2 \text{ with } X_1)$

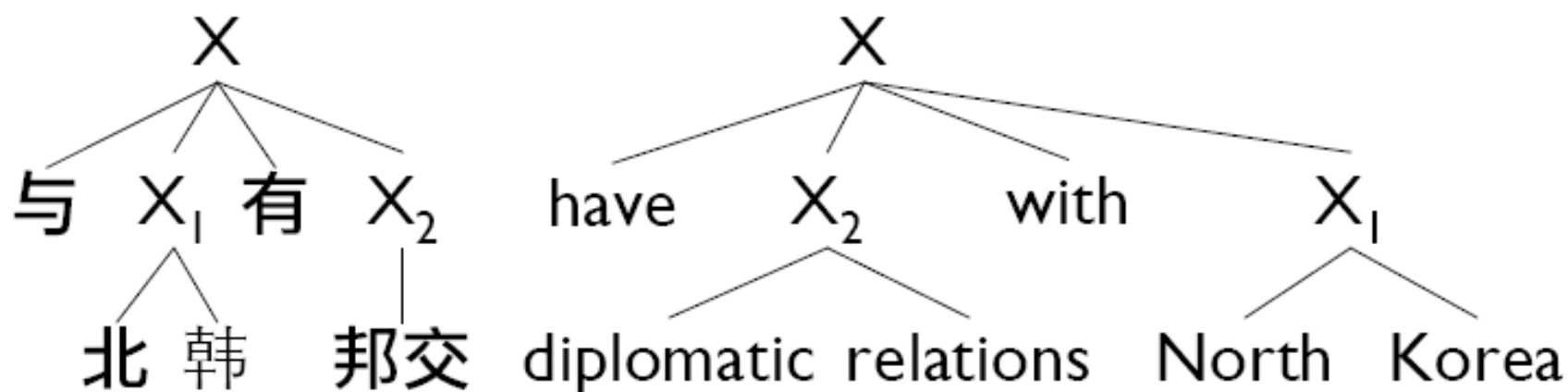


$(X \rightarrow \text{北 韩}, X \rightarrow \text{North Korea})$

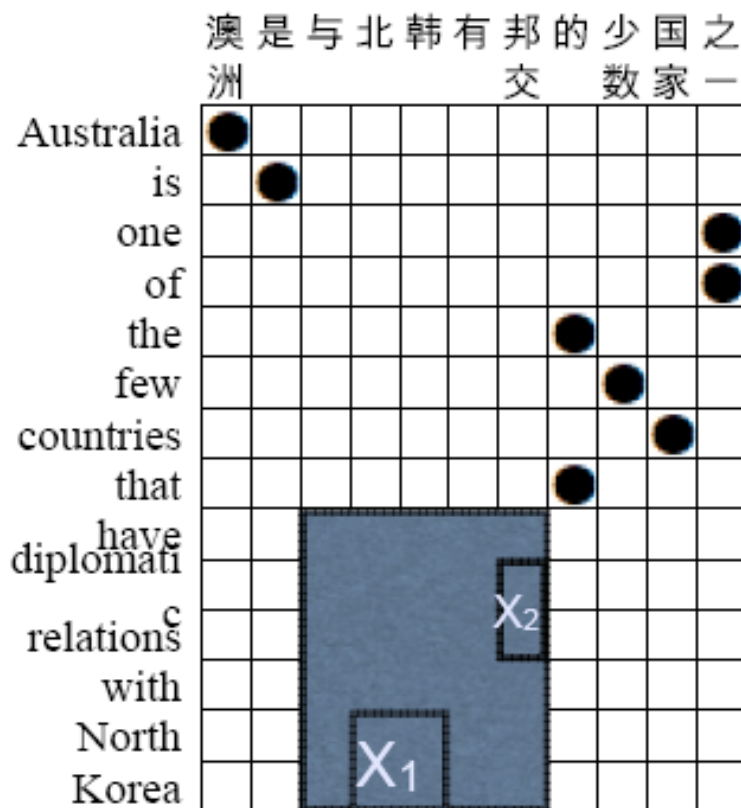


$(X \rightarrow \text{邦交}, X \rightarrow \text{diplomatic relations})$

用同步语法表示层次短语 (2)



层次短语的抽取



(与 北 韩 有 邦 交,
have diplomatic
relations with
North Korea)

(邦 交, diplomatic
relations)

(北 韩, North Korea)

($X \rightarrow$ 与 X_1 有 X_2 ,
 $X \rightarrow$ have X_2 with X_1)

约束：降低复杂度

- 用于抽取规则的短语长度 ($\leq 7 - 20$)
- 规则长度 ($\leq 5 - 6$)
- 规则中至少要有有一个终结符
- 最多有两个不相邻的非终结符
- 句法约束?

非终结符标记

- 到目前未知只采用一个非终结符 **X**
- 可能的扩展：
 - 句法类型
 - 其他信息，如命名实体标记（人名、地名等）

规则举例

$X \rightarrow \text{的}$	$X \rightarrow \text{'s}$
$X \rightarrow X_1 \text{ 的 } X_2$	$X \rightarrow \text{the } X_2 \text{ of } X_1$
$X \rightarrow X_1 \text{ 的 } X_2$	$X \rightarrow \text{the } X_2 \text{ that } X_1$
<hr/>	
$X \rightarrow \text{在}$	$X \rightarrow \text{in}$
$X \rightarrow \text{在 } X_1 \text{ 下}$	$X \rightarrow \text{under } X_1$
$X \rightarrow \text{在 } X_1 \text{ 前}$	$X \rightarrow \text{before } X_1$
<hr/>	
$X \rightarrow \text{今年 } X_1$	$X \rightarrow X_1 \text{ this year}$
$X \rightarrow X_1 \text{ 之一}$	$X \rightarrow \text{one of } X_1$
$X \rightarrow X_1 \text{ 总统}$	$X \rightarrow \text{president } X_1$

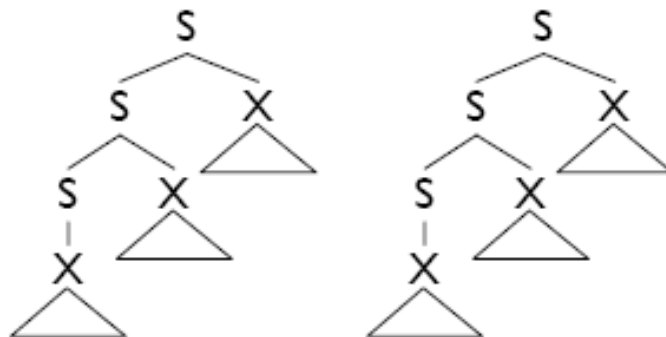
粘合规则 (Glue Rules)

- 找不到可用的规则时，引入粘合规则

$$(S \rightarrow S_1 X_2, S \rightarrow S_1 X_2)$$

$$(S \rightarrow X_1, S \rightarrow X_1)$$

- 粘合规则的作用在于将短语的译文从左到右依次顺序“粘合”成完整的译文：



特殊规则

- 实际的翻译系统中，通常需要一些特殊的翻译模块：
 - 数词
 - 时间词
 - 人名、地名、机构名
 - 新闻 **byline**
- 将以上模块翻译的结果处理成一条规则：
($X \rightarrow$ 一百二十三, $X \rightarrow 123$)

模型

- 直接利用同步上下文无关语法的概率模型
- 通过对数线性模型融合其他特征，如传统短语模型的各种特征

模型特征

- Language model $p(e)$
- Phrase translation probabilities $p(\bar{f} | \bar{e}), p(\bar{e} | \bar{f})$
- PCFG-like probability $p(\bar{f})$ (since all rules are $X \rightarrow \bar{f}$)
- Probability for glue rule $S \rightarrow SX$
- Word penalty, phrase penalty
- Constituent reward (optional)

模型特征

- Phrase translation:

$$p(X \rightarrow X_1 \text{ 之一} \mid X \rightarrow \text{one of } X_1)$$

$$p(X \rightarrow \text{one of } X_1 \mid X \rightarrow X_1 \text{ 之一})$$

- Lexical weighting (Koehn):

$$\frac{1}{2}[p(\text{之一} \mid \text{one}) + p(\text{之一} \mid \text{of})]$$

$$p(\text{one} \mid \text{之一}) \times p(\text{of} \mid \text{之一})$$

解码

- 类似于句法分析，在对源语言分析的同时，产生目标语言的结构。
- 算法复杂度 $O(n^3)$
- 为了减少搜索时间，只将抽取出来的规则用于比较短的串（如少于10-15个词），对于更长的串只使用粘合规则。

小结

- 形式上基于句法的模型
- 性能明显超过基于短语的模型
- 完全兼容基于短语的模型
- 所有规则可以自动抽取

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

树到串翻译模型

- 树到串翻译模型指这样一类翻译模型：
 - 在源语言端进行句法分析
 - 在目标语言端不进行句法分析
 - 从源语言端句法分析和词语对齐的语料库中抽取翻译规则并构造翻译模型
- 树到串翻译模型的发展经历了三个阶段：
 - 基于树的方法（**Tree-based Approach**）
 - 基于森林的方法（**Forest-base Approach**）
 - 基于串的方法，句法分析和解码联合方法（**String-based Approach, Joing Parsing and Translation**）

基于树的方法

Tree-to-String Model

- Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In Proceedings of COLING/ACL 2006, pages 609-616, Sydney, Australia, July.

Meritorious Asian NLP Paper Award

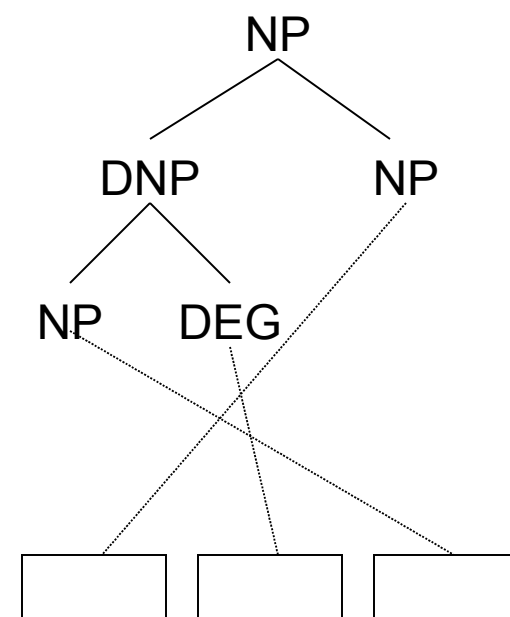
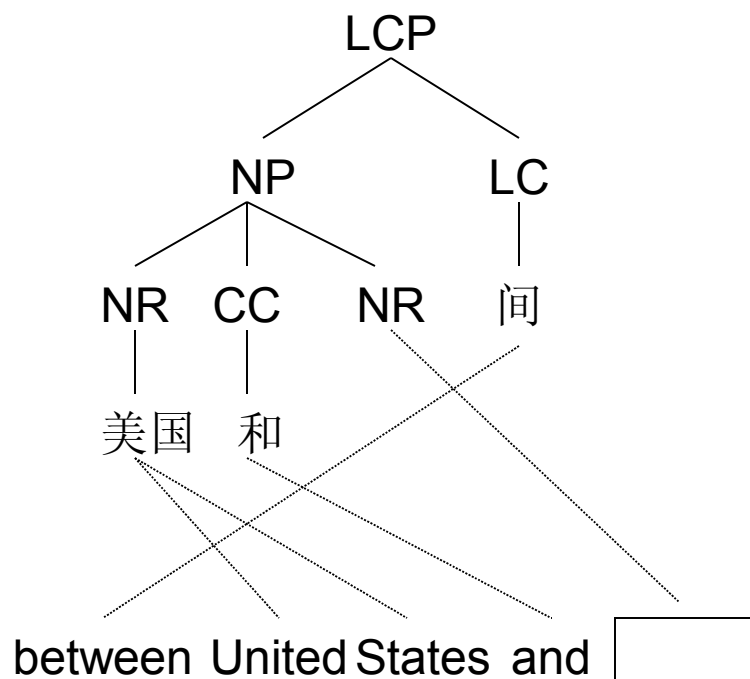
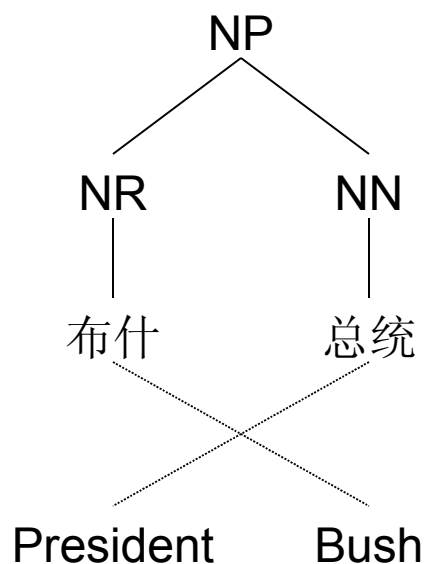
基于树到串对齐模板的翻译模型

- 基于树到串对齐模板的翻译模型（刘洋， ICT）
A Translation Model Based on Tree-to-String Alignment Template
- Yang Liu, Qun Liu, and Shouxun Lin. 2006.
Tree-to-String Alignment Template for Statistical Machine Translation. COLING-ACL 2006, Sydney, Australia, July 17-21.
- Yang Liu, Yun Huang, Qun Liu and Shouxun Lin, Forest-to-String Statistical Translation Rules, ACL2007, Prague, Czech, June 2007

基于树到串对齐模板的翻译模型

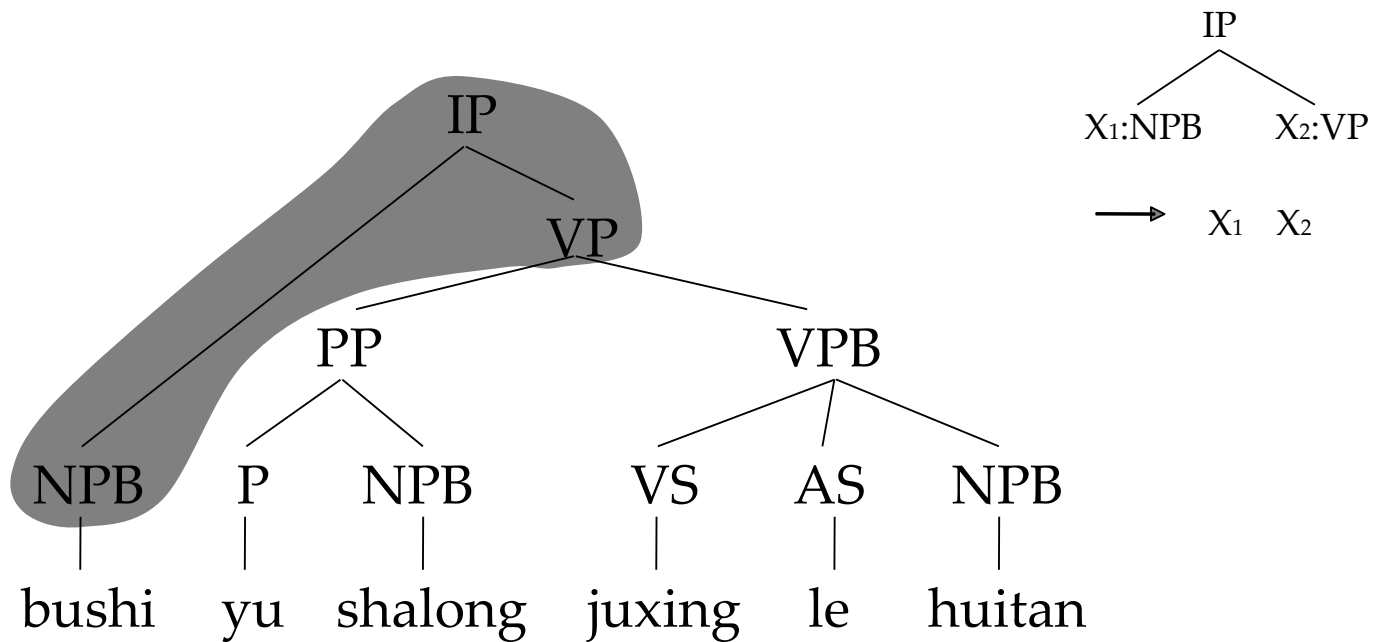
- 基于树到串对齐模板（简称 **TAT**）的统计翻译模型是一种在源语言进行句法分析的基于语言学句法结构的统计翻译模型
- 树到串对齐模板既可以生成终结符也可以生成非终结符，既可以执行局部重排序也可以执行全局重排序
- 从经过词语对齐和源语言句法分析的双语语料库上自底向上自动抽取 **TAT**
- 自底向上的柱搜索算法

树到串对齐模板



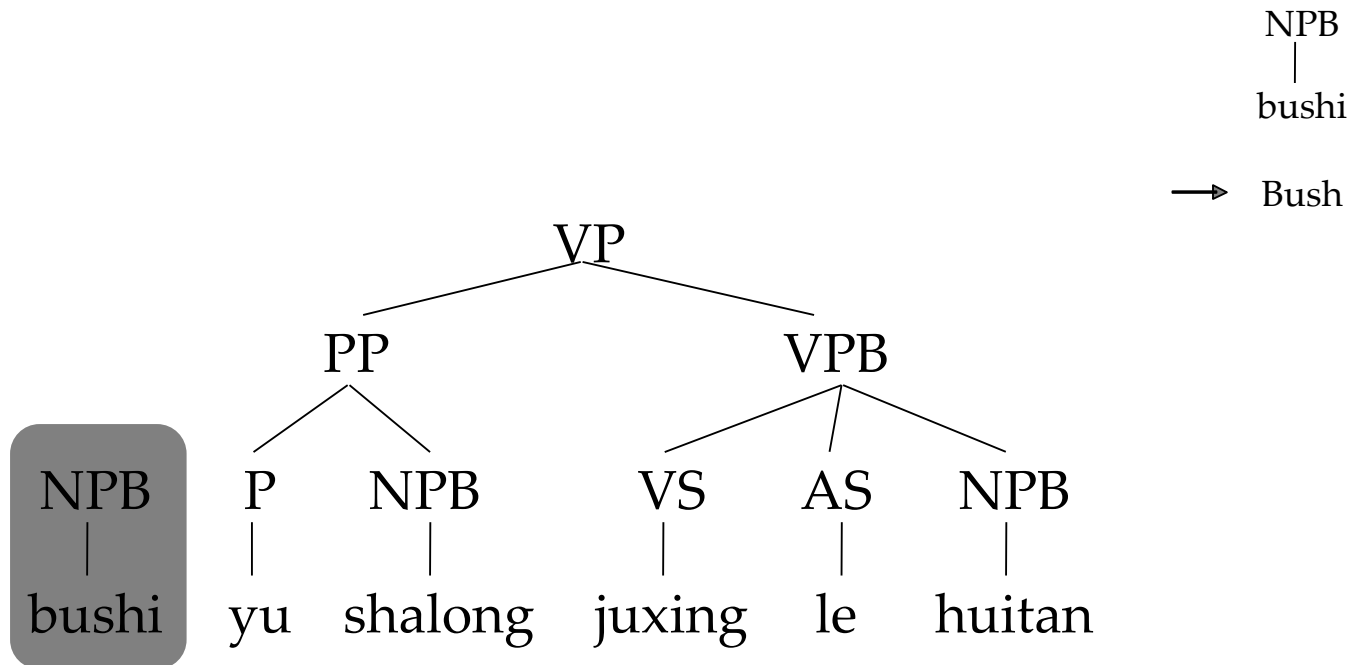
Tree-based Translation

- Recursive rewrite by pattern-matching



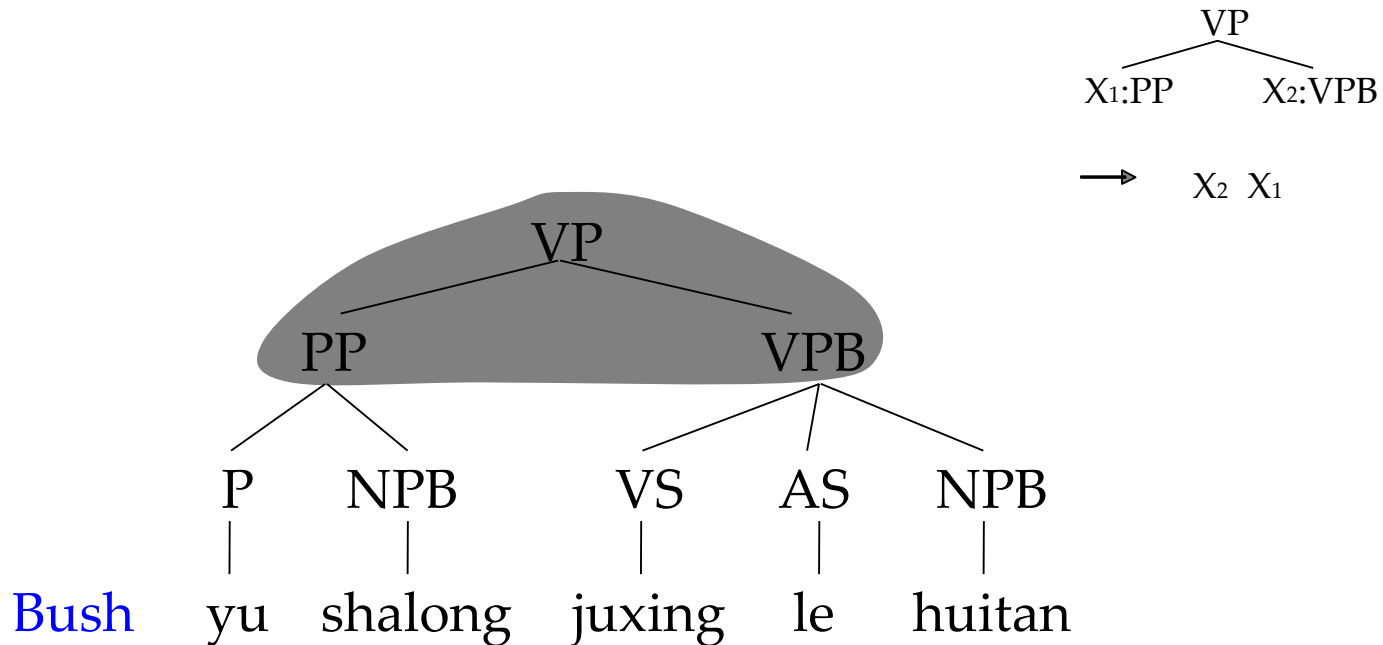
Tree-based Translation

- Recursive rewrite by pattern-matching



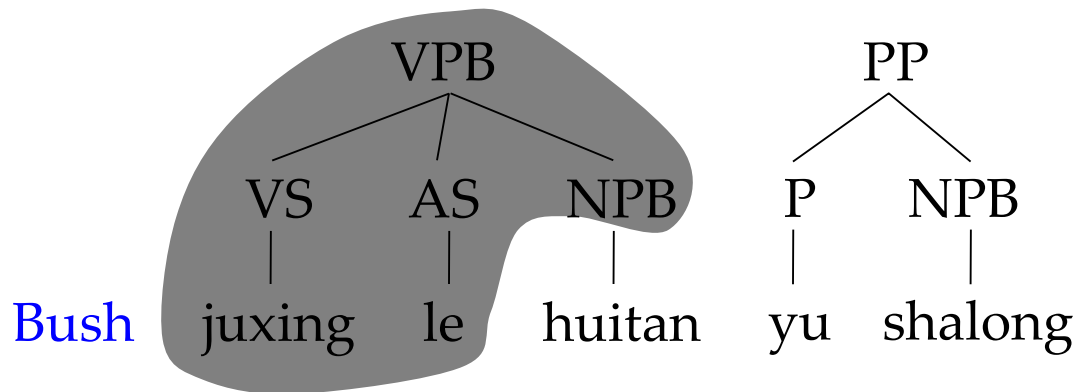
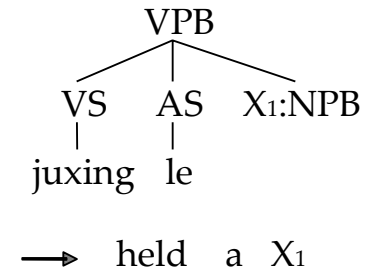
Tree-based Translation

- Recursive rewrite by pattern-matching



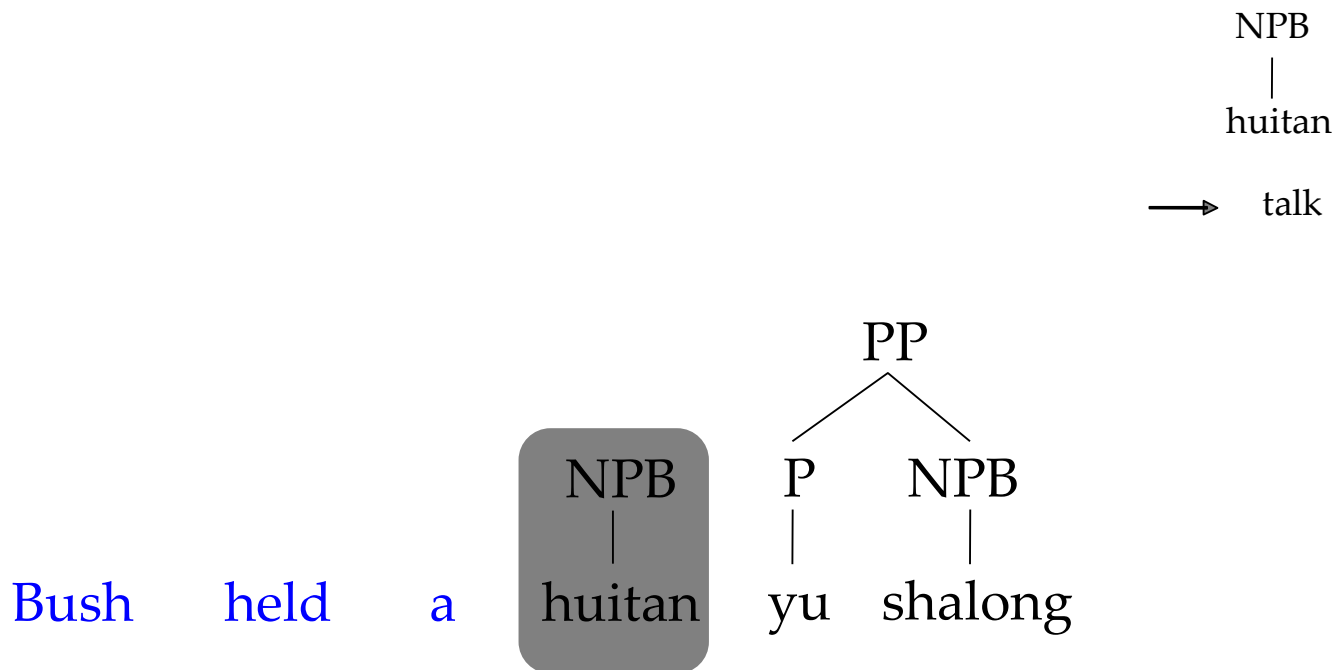
Tree-based Translation

- Recursive rewrite by pattern-matching



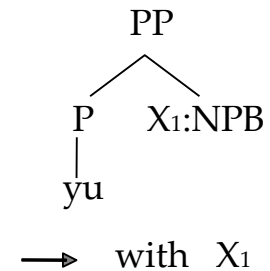
Tree-based Translation

- Recursive rewrite by pattern-matching

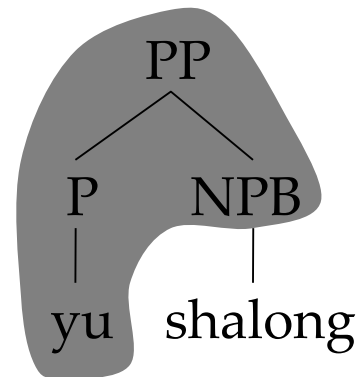


Tree-based Translation

- Recursive rewrite by pattern-matching




Bush held a talk



Tree-based Translation

- Recursive rewrite by pattern-matching

NPB
|
shalong
→ Sharon

Bush held a talk with 

Tree-based Translation

- Recursive rewrite by pattern-matching

Bush held a talk with Sharon

模型

$$\begin{aligned} & Pr(e_1^I, z_1^K | f_1^J) \\ &= \frac{\exp[\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, z_1^K)]}{\sum_{e_1^I, z_1^K} \exp[\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, z_1^K)]} \\ & \hat{e}_1^I = \operatorname{argmax}_{e_1^I, z_1^K} \left\{ \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J, z_1^K) \right\} \end{aligned}$$

模型特征

$$h_1(e_1^I, f_1^J) = \log \prod_{k=1}^K \frac{N(z) \cdot \delta(T(z), \tilde{T}_k)}{N(T(z))}$$

$$h_2(e_1^I, f_1^J) = \log \prod_{k=1}^K \frac{N(z) \cdot \delta(T(z), \tilde{T}_k)}{N(S(z))}$$

$$h_3(e_1^I, f_1^J) = \log \prod_{k=1}^K lex(T(z)|S(z)) \cdot \delta(T(z), \tilde{T}_k)$$

$$h_4(e_1^I, f_1^J) = \log \prod_{k=1}^K lex(S(z)|T(z)) \cdot \delta(T(z), \tilde{T}_k)$$

$$h_5(e_1^I, f_1^J) = K$$

$$h_6(e_1^I, f_1^J) = \log \prod_{i=1}^I p(e_i | e_{i-2}, e_{i-1})$$

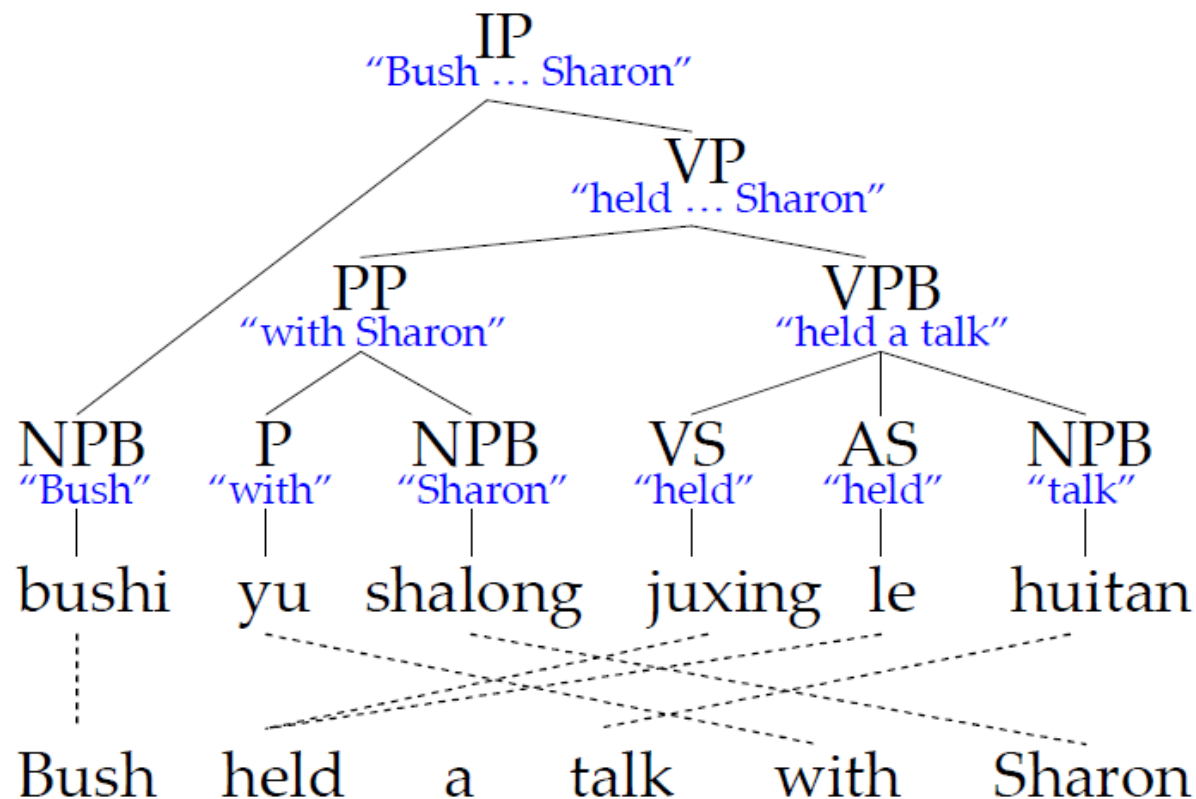
$$h_7(e_1^I, f_1^J) = I$$

训练

- 数据：源语言句法分析和词语对齐的双语语料库
- 自底向上抽取
- 为避免抽取的 **TAT** 数量过大，需要对抽取过程施加一些约束：
 - 树高度约束 $\text{height}(T) \leq h$
 - 子节点个数约束 $\text{number_of_children}(T) \leq c$
 - 目标语言的首尾词必须是对齐的

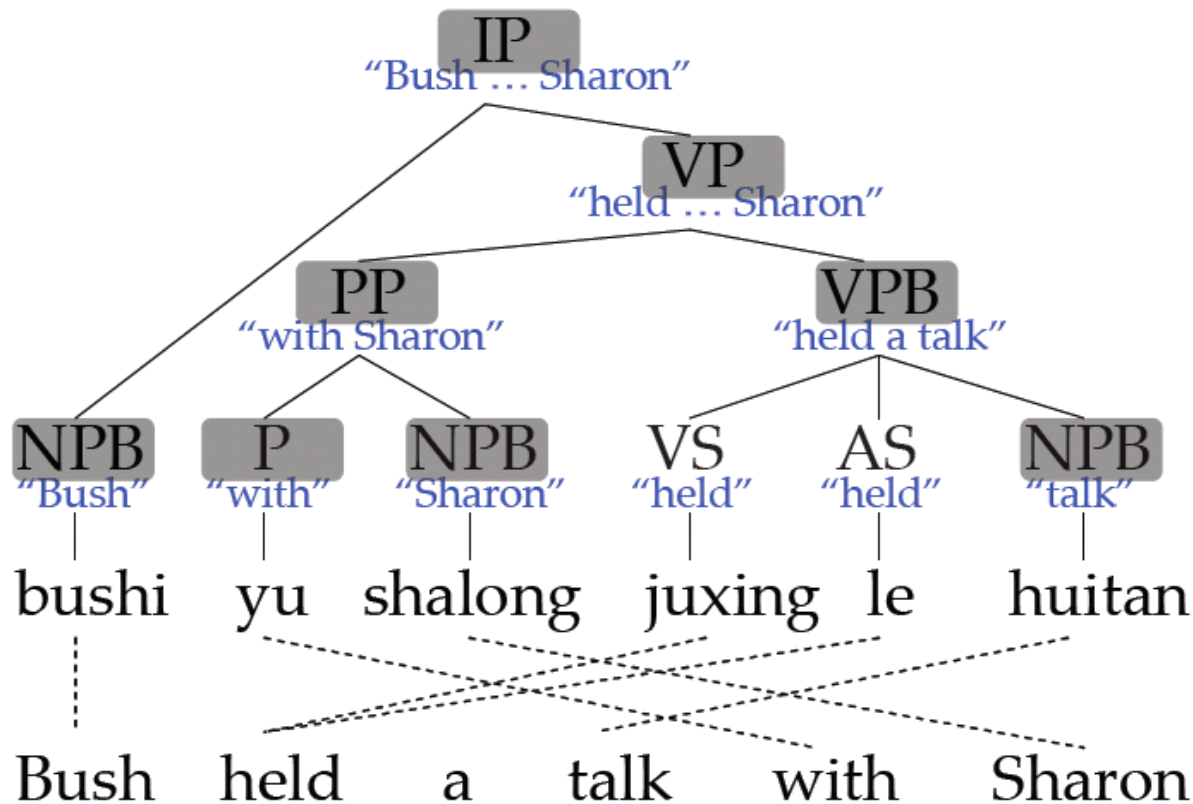
Tree-to-String Rule Extraction

- Compute target spans



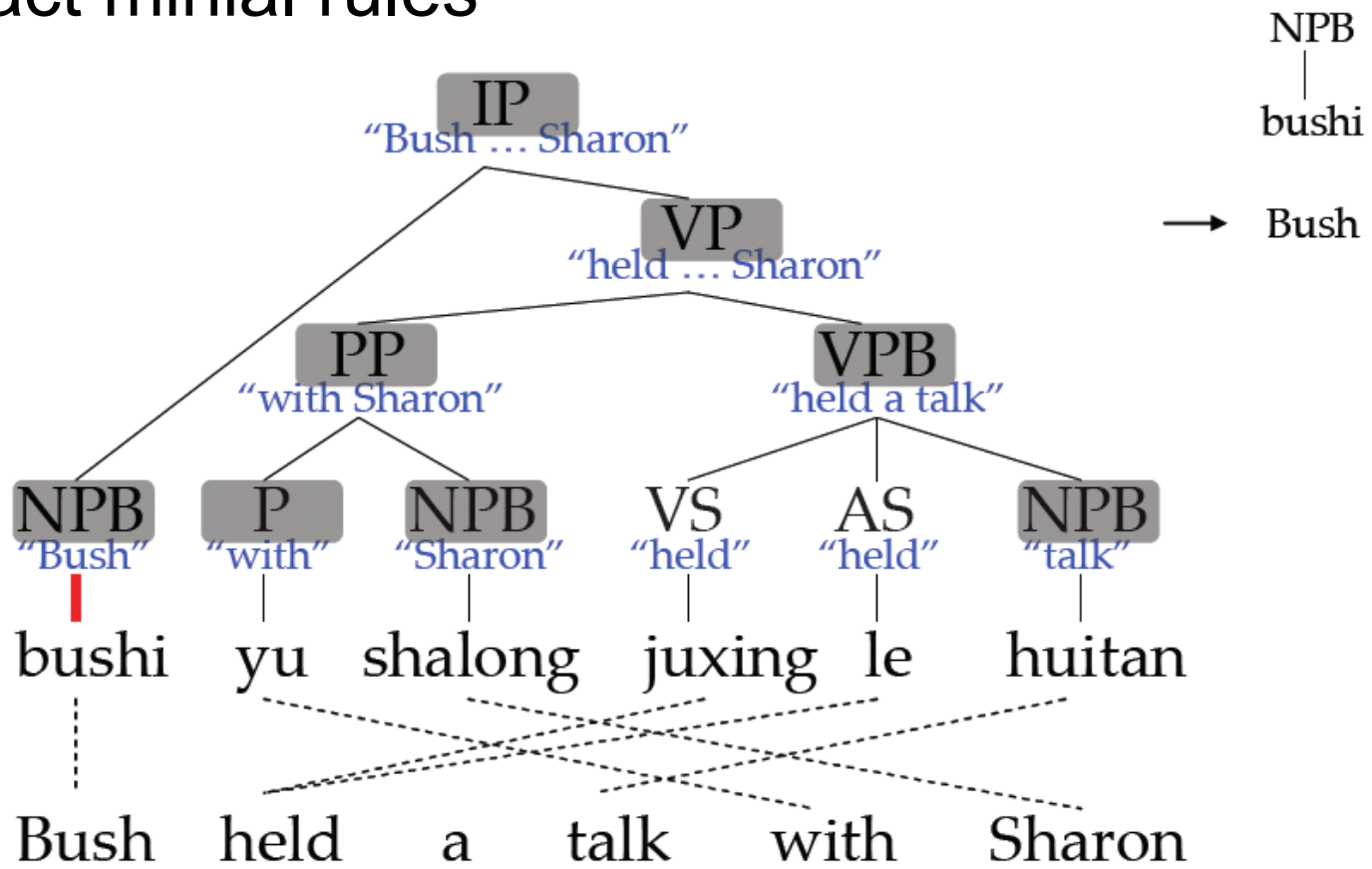
Tree-to-String Rule Extraction

- Find admissible nodes



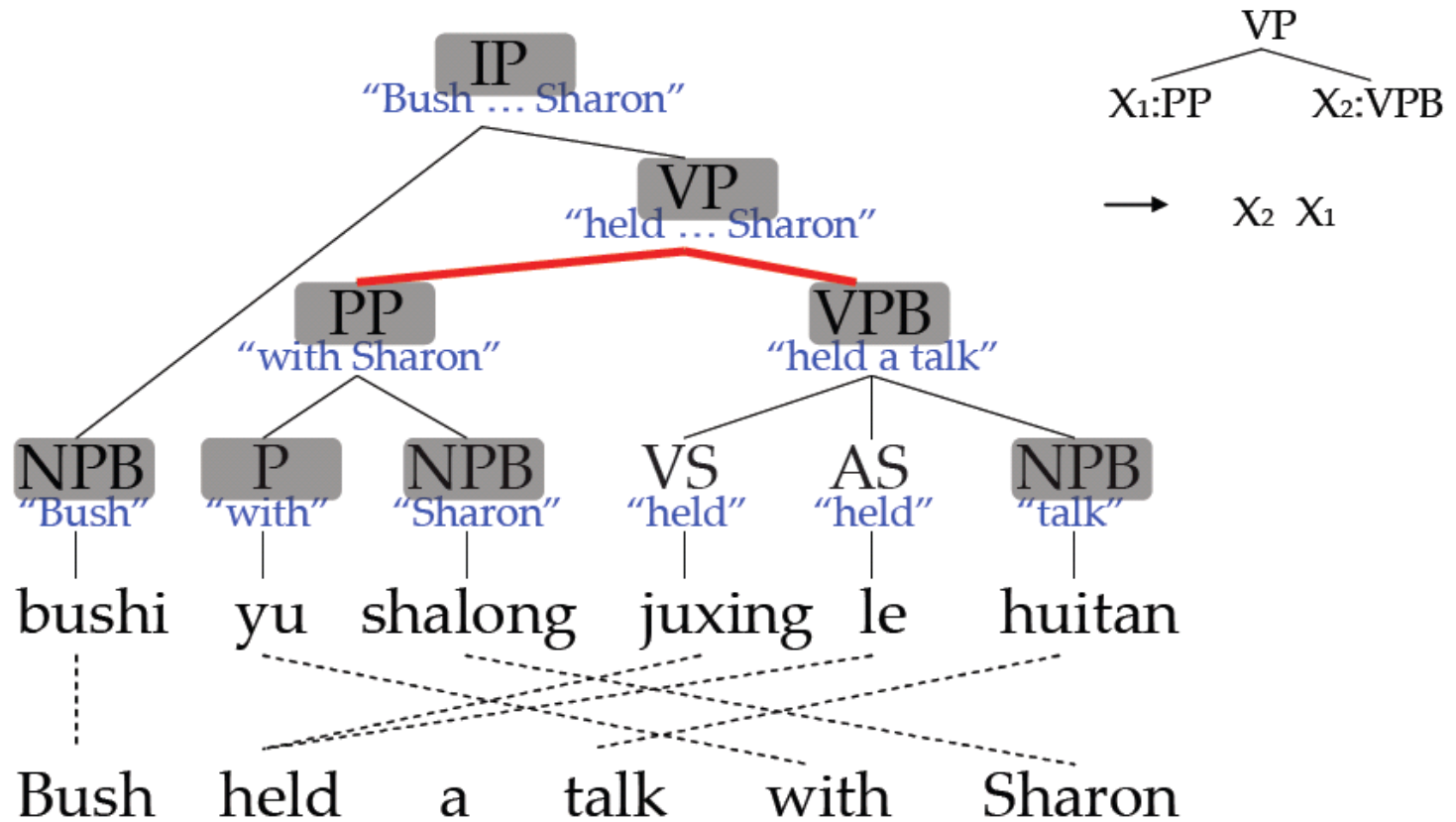
Tree-to-String Rule Extraction

- Extract minial rules



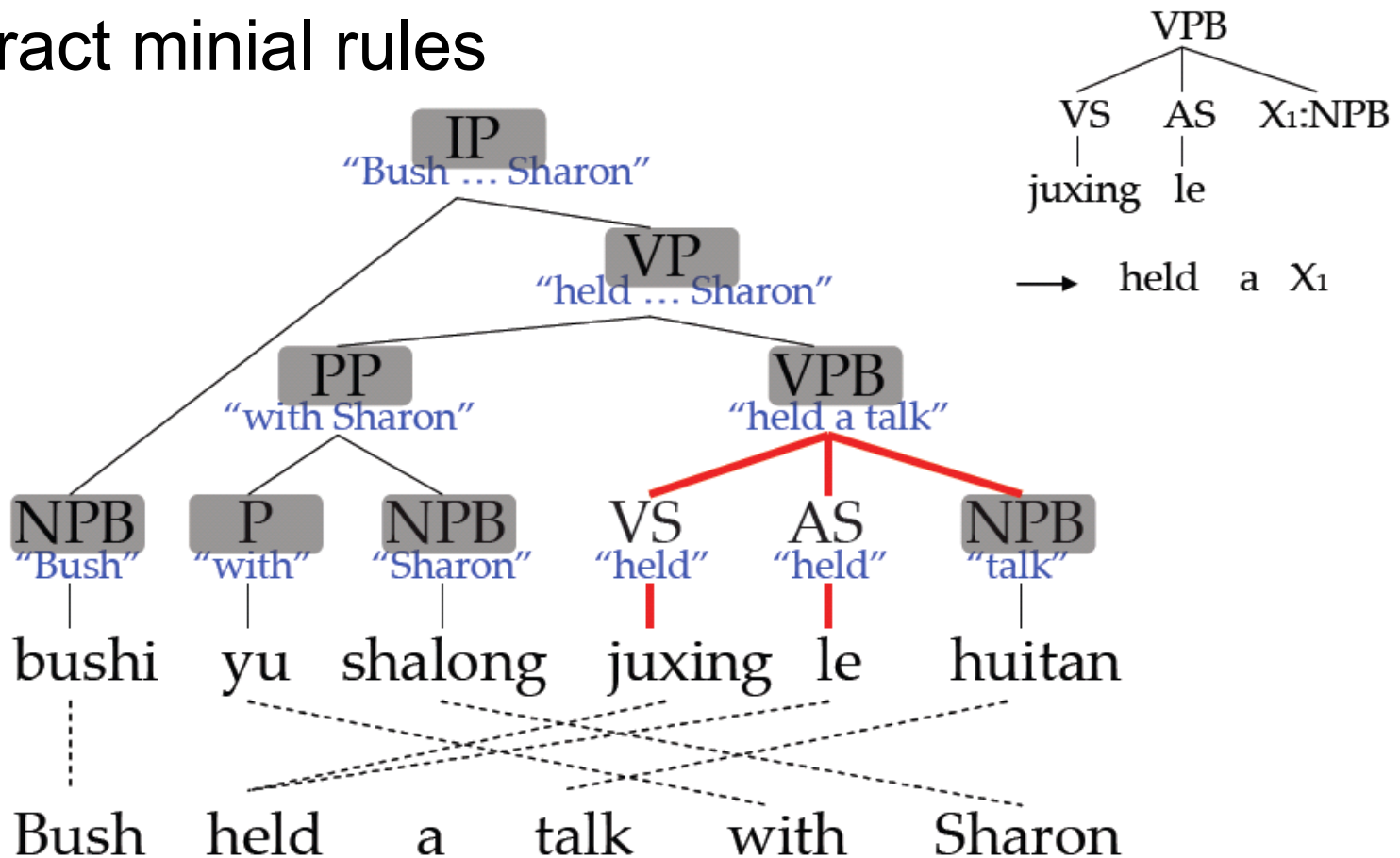
Tree-to-String Rule Extraction

- Extract minial rules



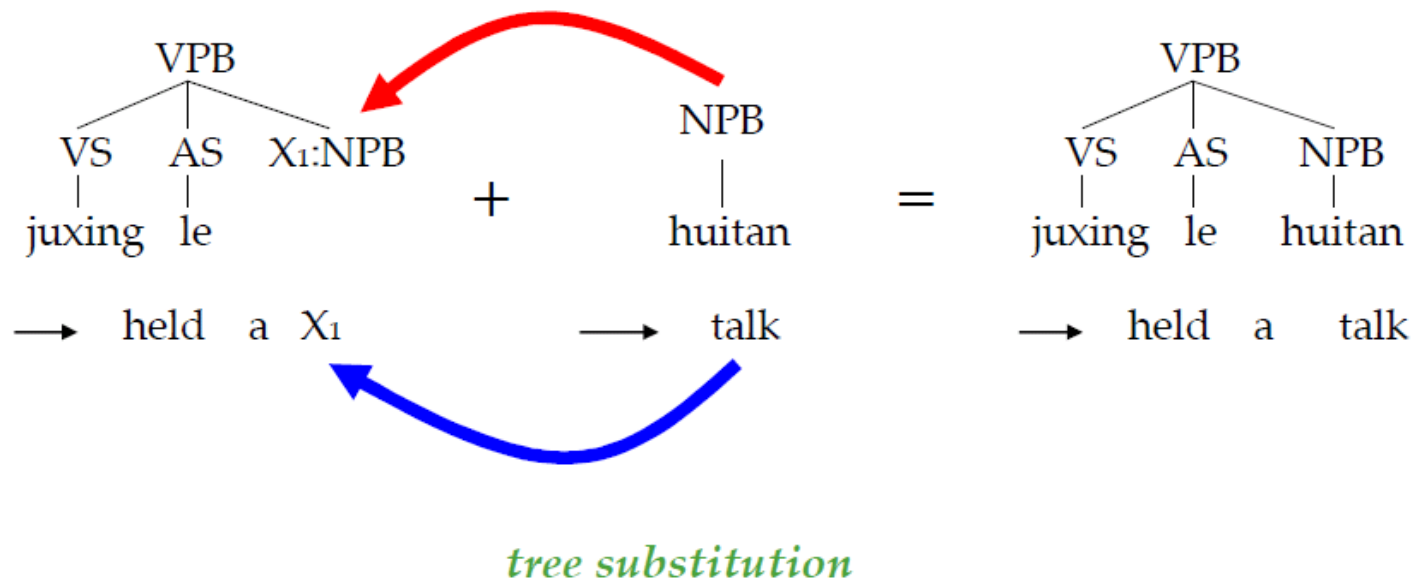
Tree-to-String Rule Extraction

- Extract minial rules



Tree-to-String Rule Extraction

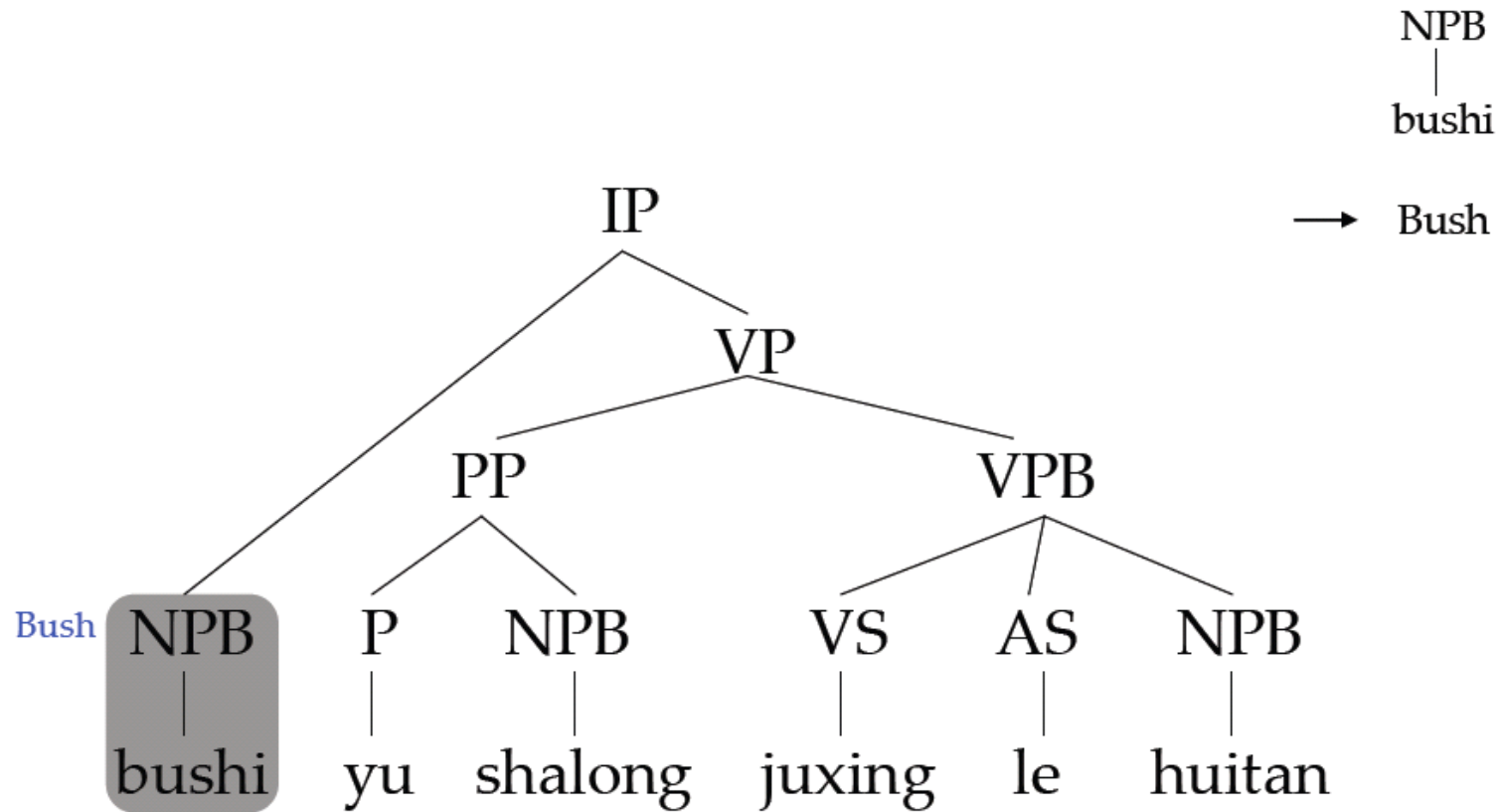
- Get composed rules



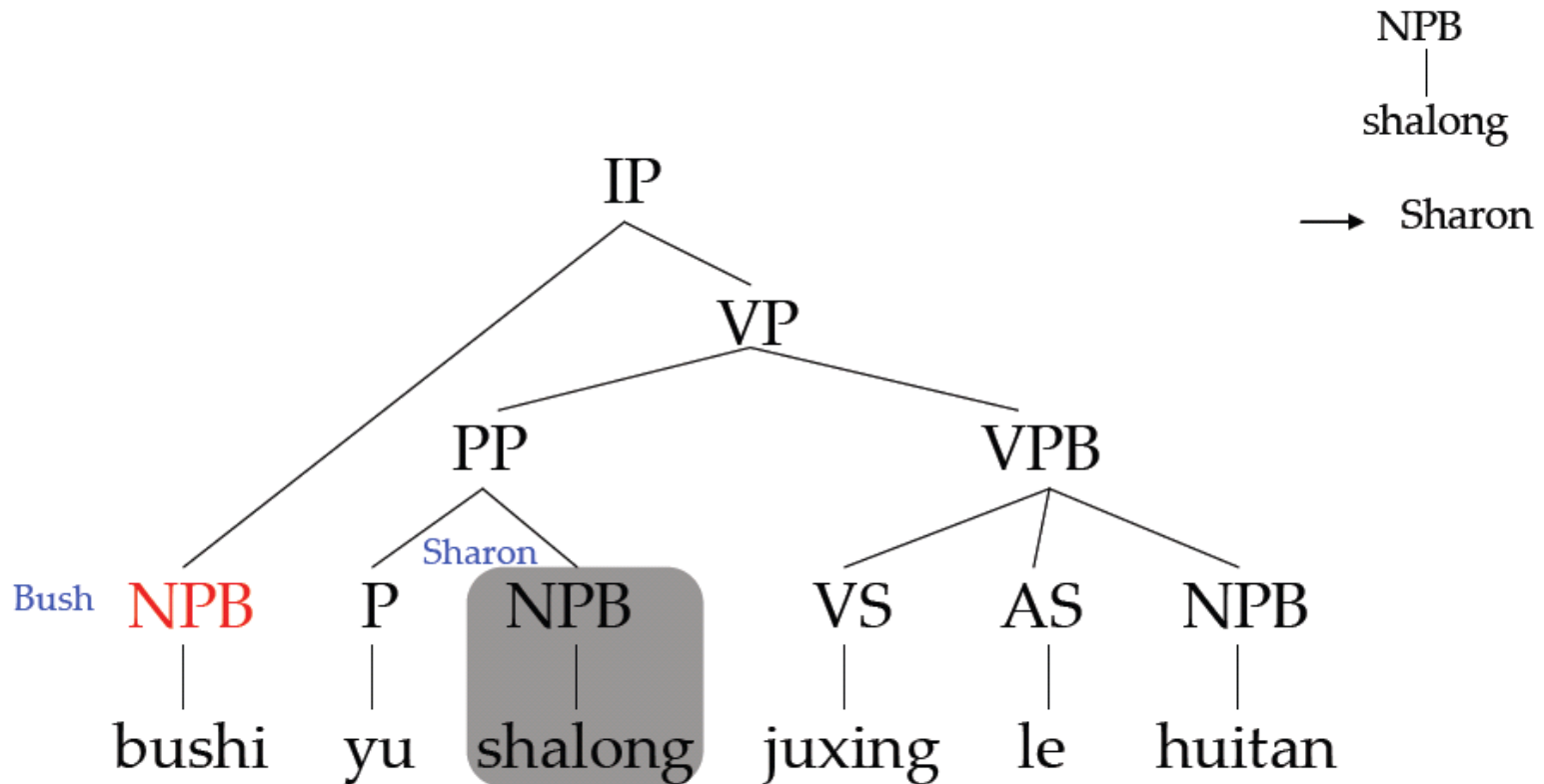
解码

- 自底向上
- 柱搜索（**Beam Search**）
- 对于每一棵子树，找到所有与其根节点匹配的 **TAT**，计算其候选译文（**Candidate**）
- 候选译文（**Candidate**）的数据结构：
 - **TAT** 序列
 - 部分翻译结果
 - 累积的特征值
 - 累积的概率值

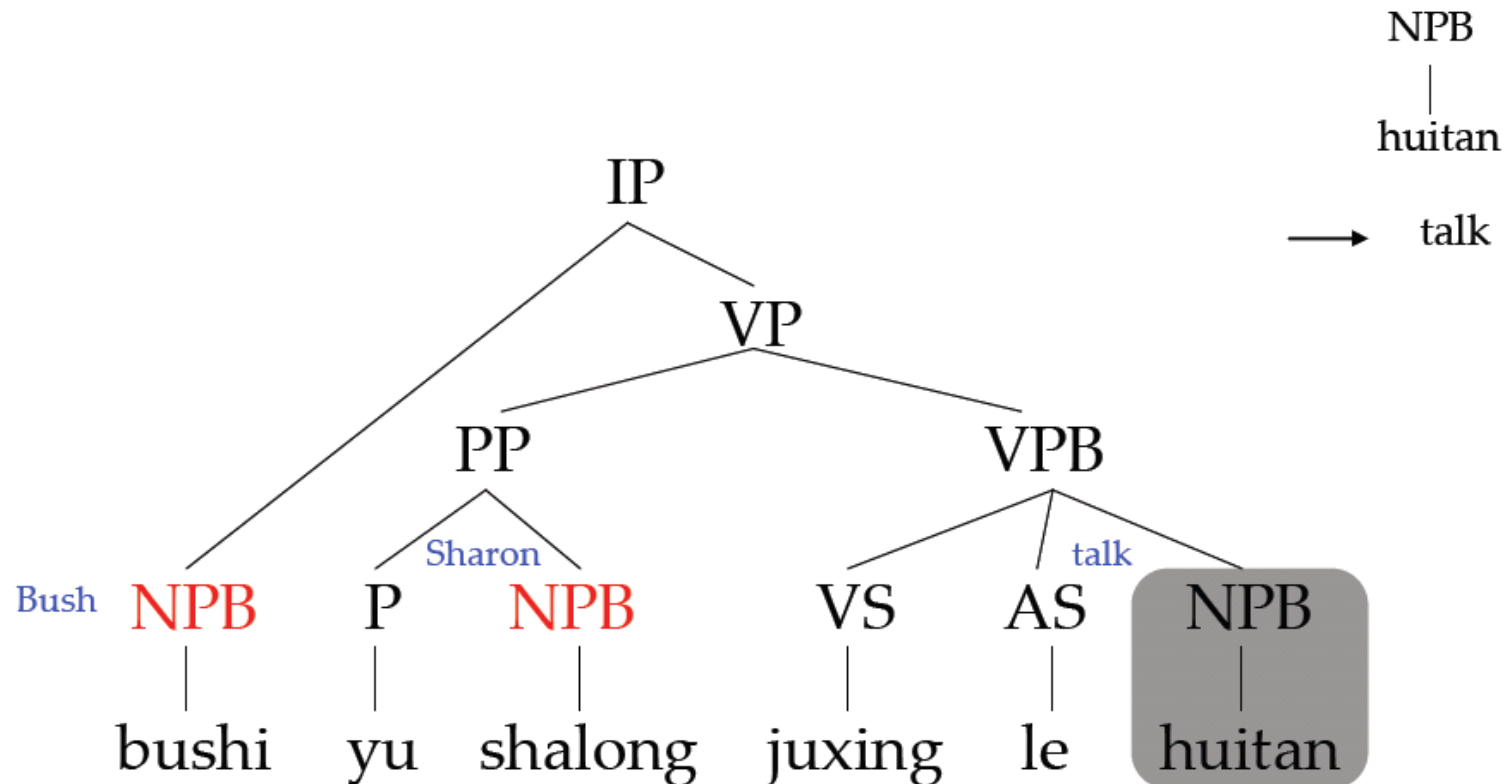
Tree-based Button-up Decoding



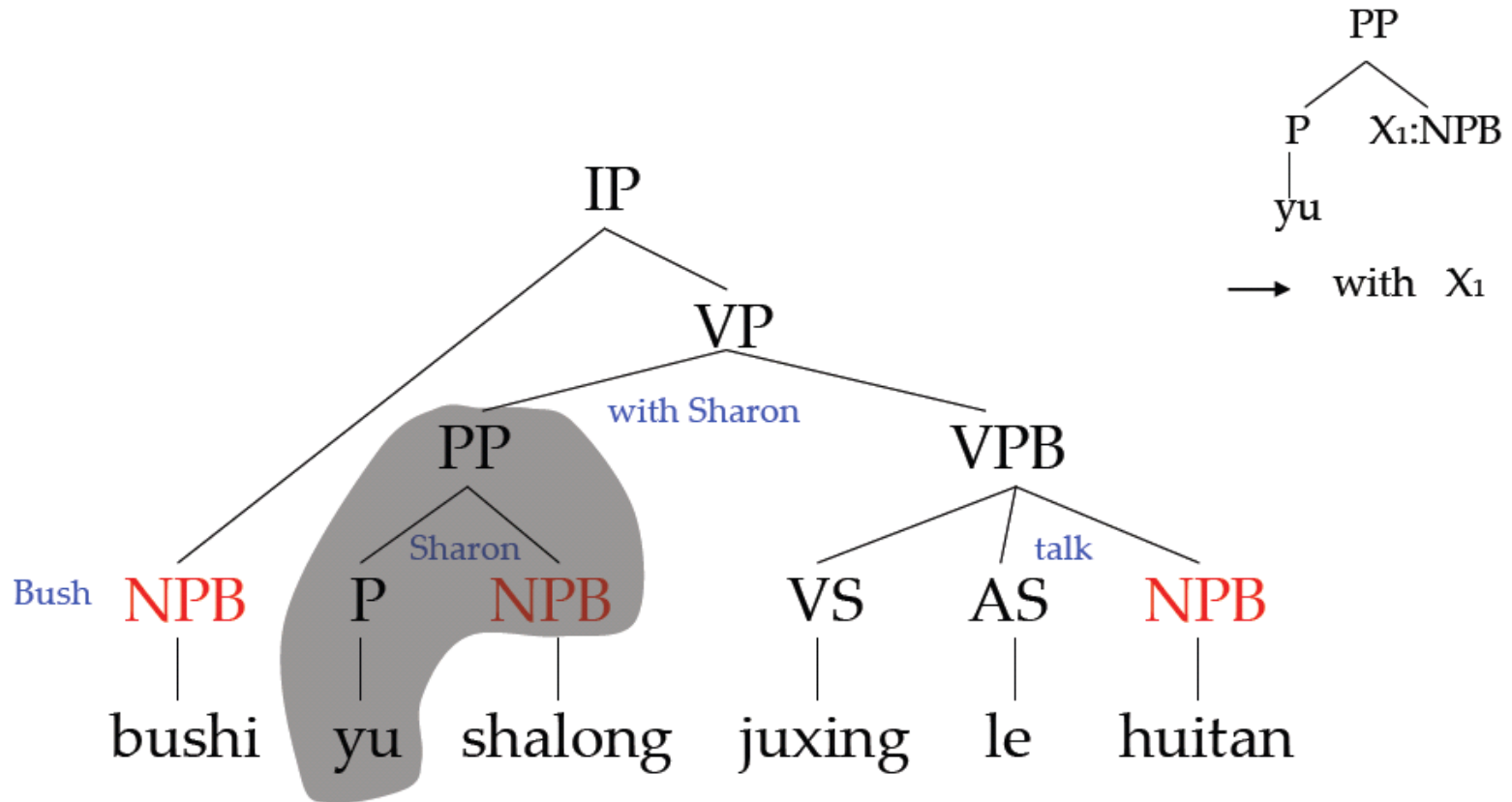
Tree-based Button-up Decoding



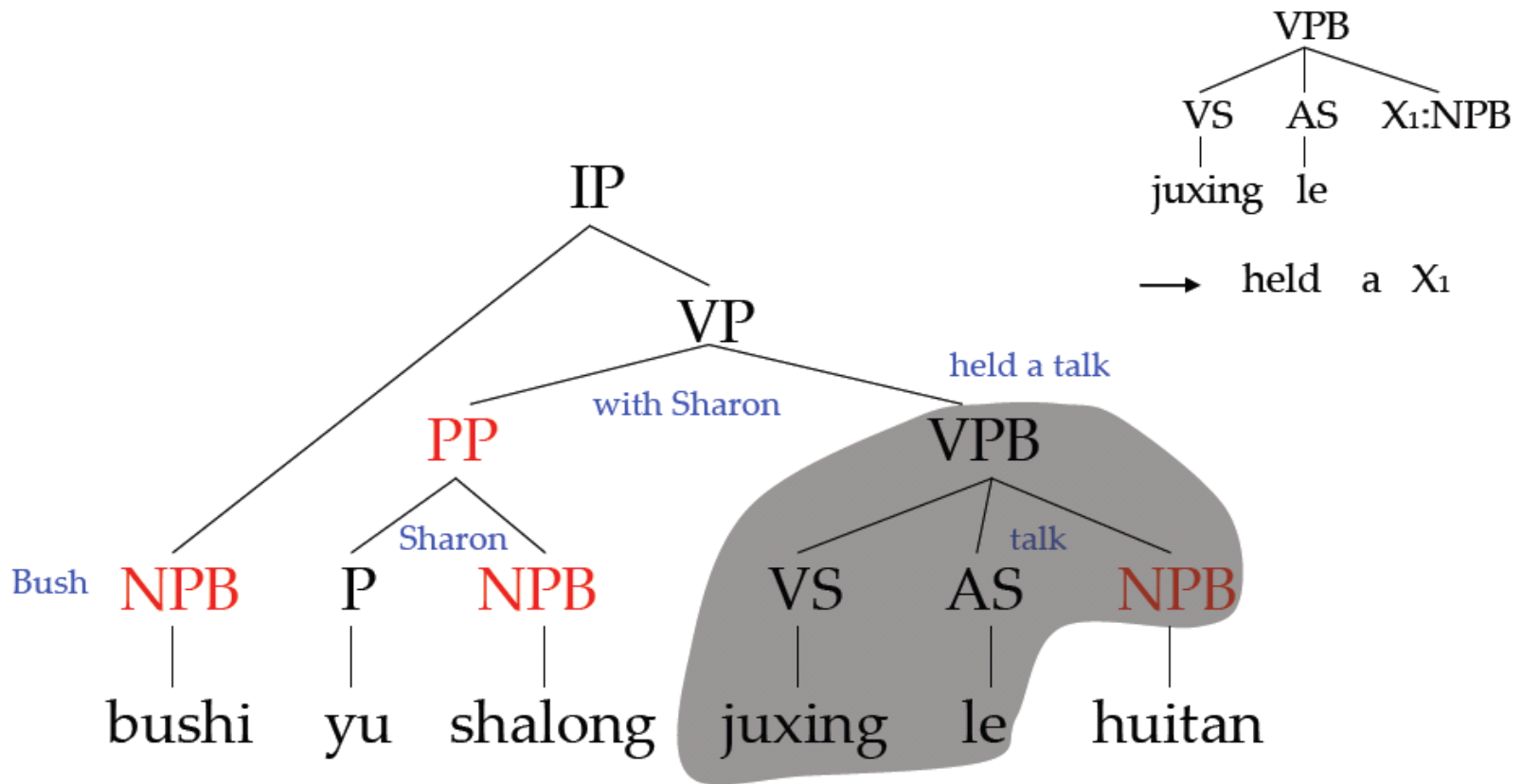
Tree-based Button-up Decoding



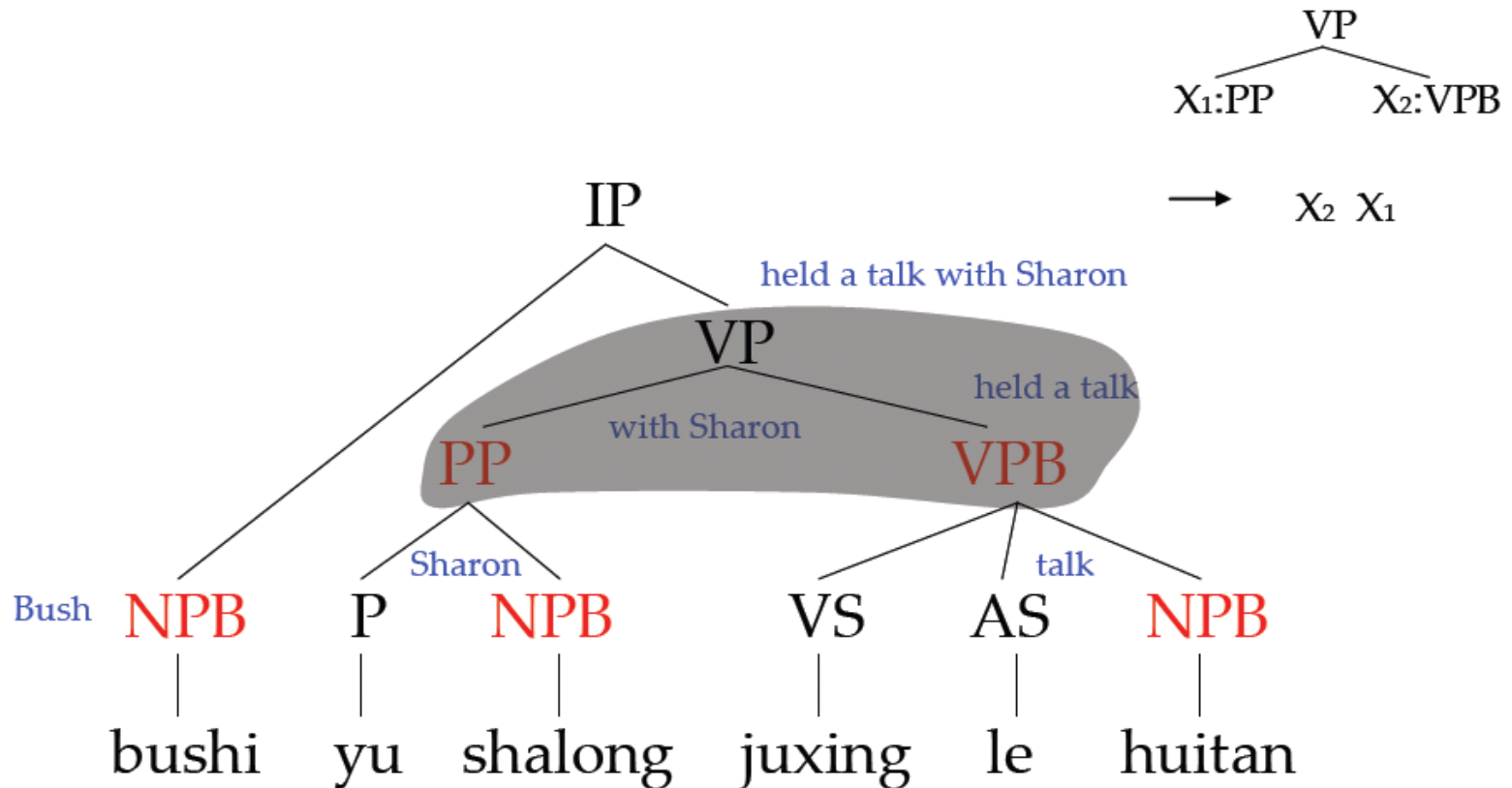
Tree-based Button-up Decoding



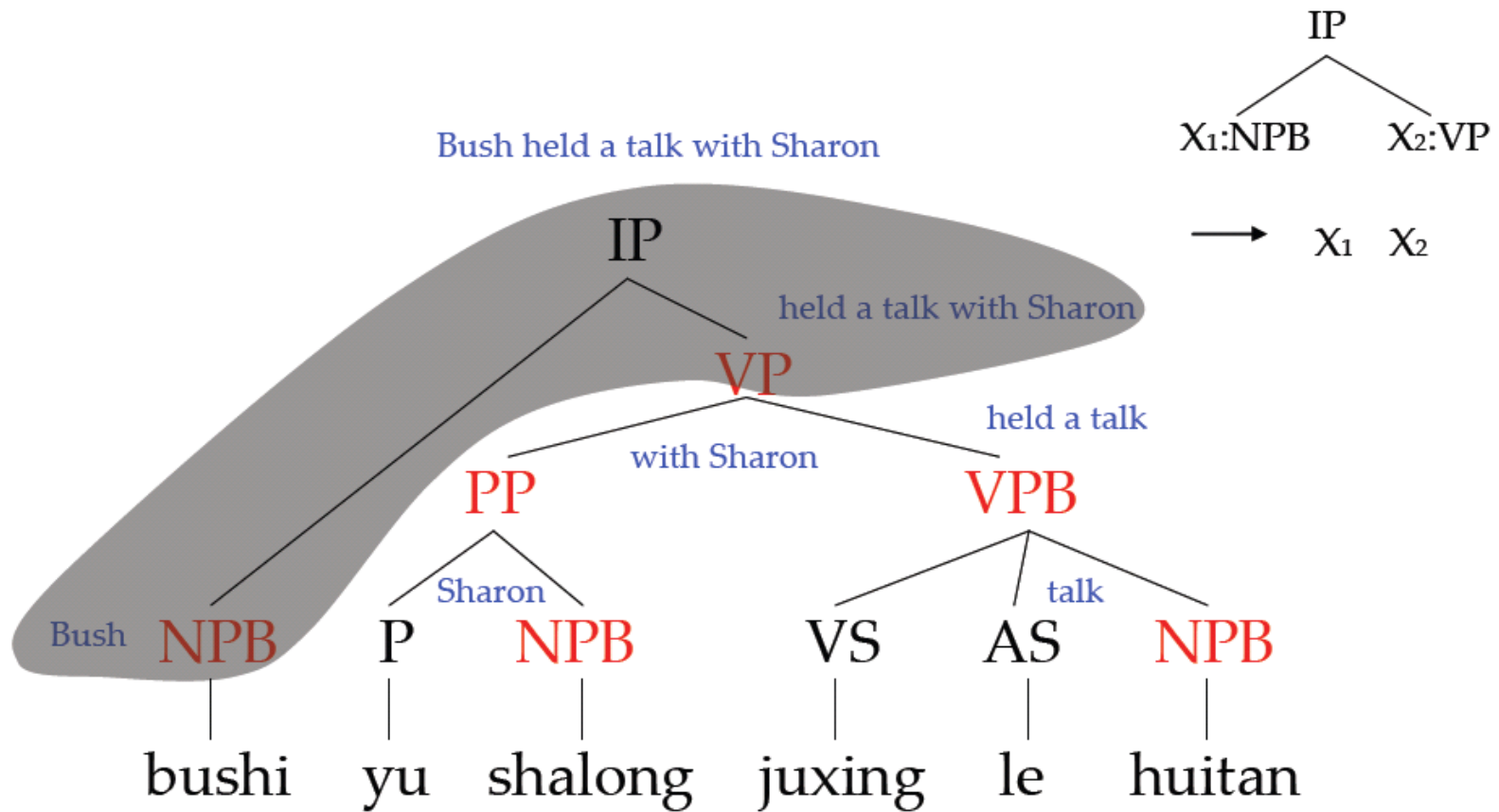
Tree-based Button-up Decoding



Tree-based Button-up Decoding

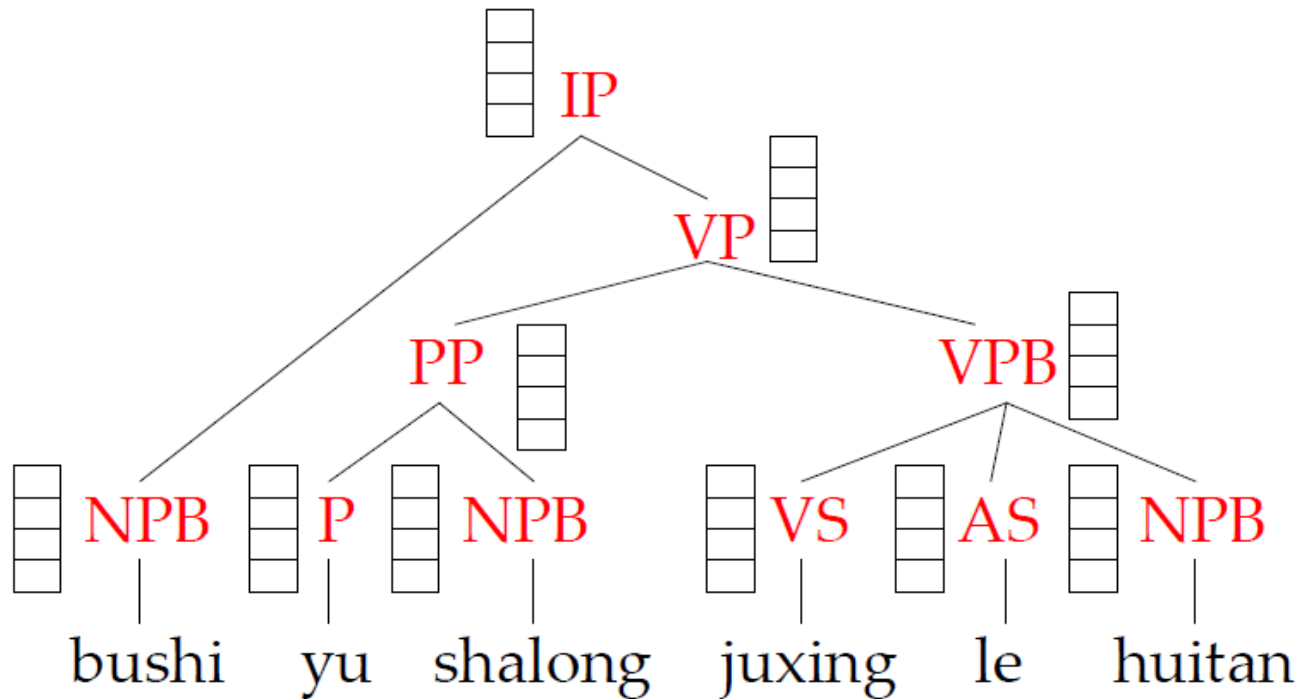


Tree-based Button-up Decoding



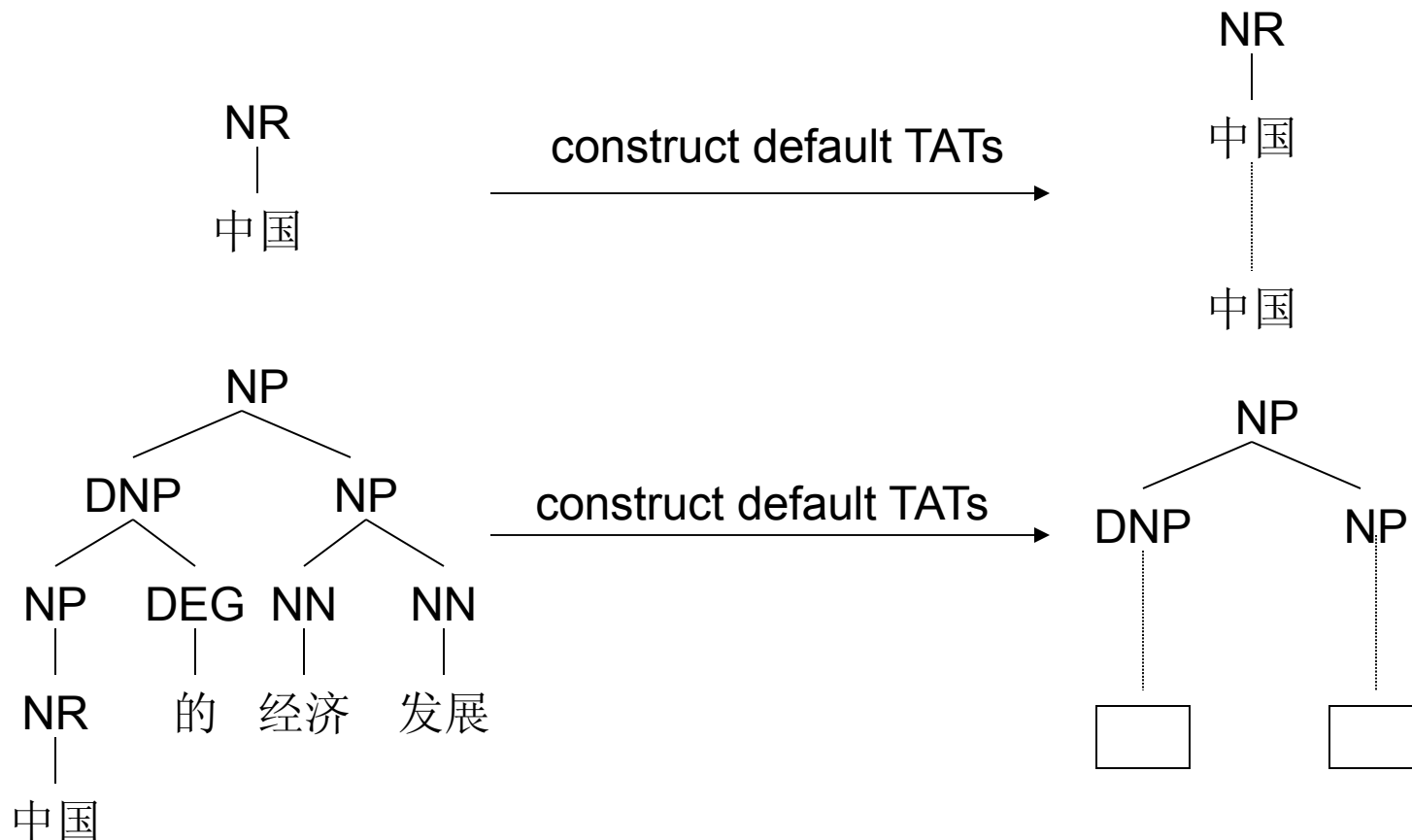
Tree-based Bottom-up Decoding

- Beam Search



解码：构造缺省 TAT

如果匹配不到合适的 TAT，就构造一个缺省的 TAT：



剪枝策略

- 模板表剪枝
 - tatTable_limit
 - tatTable_threshold
- 堆栈剪枝
 - stack_limit
 - stack_thrshold

候选译文归并 (Recombination)

The economic development of China is very rapid .

The economic develop of China is quite rapid .

The economic developing of Chinese is rapid .

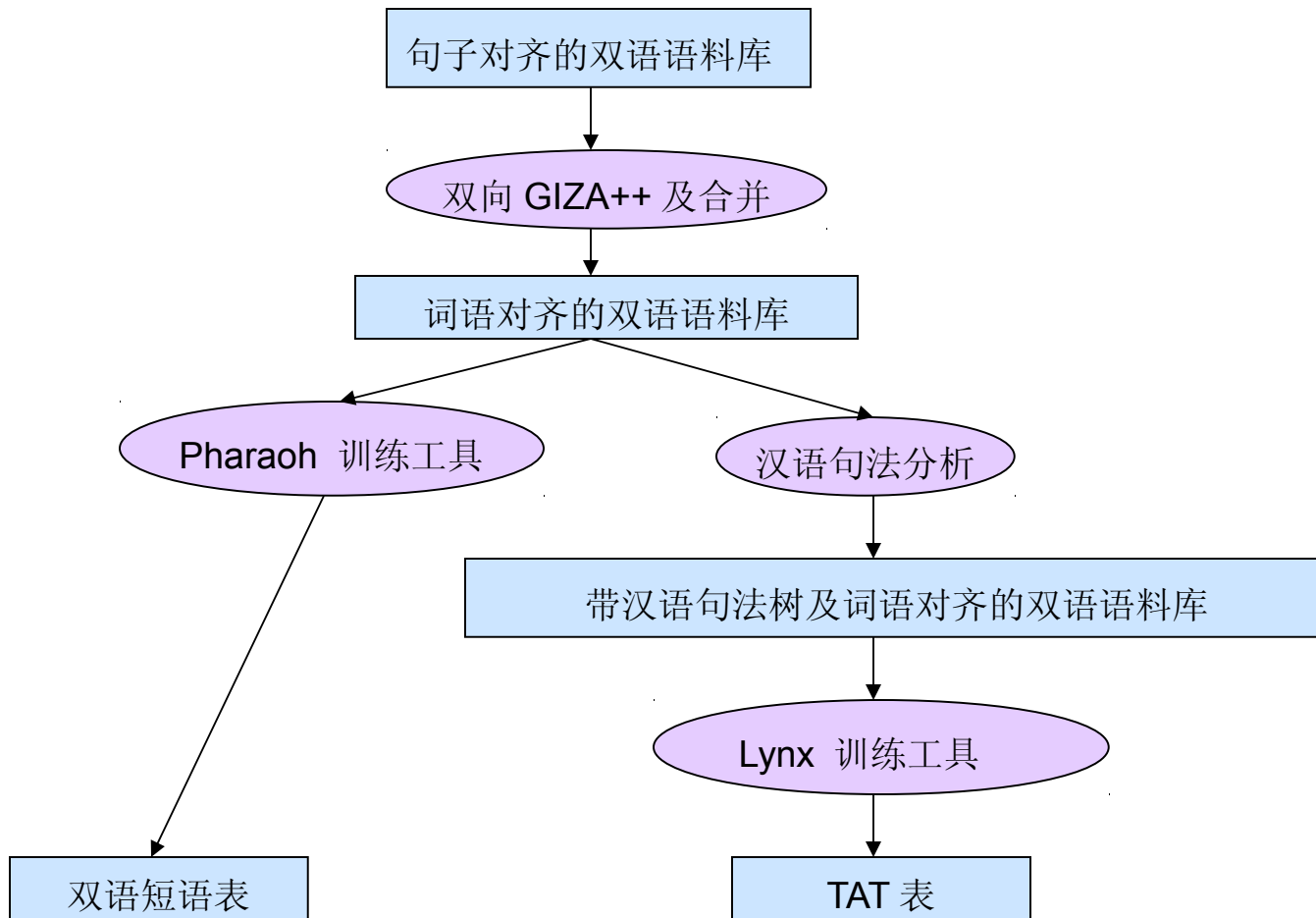
The economic development of Chinese are quite rapid .

考虑采用英文的三元语法模型，为了保证动态规划算法所要求的单调性，对同一个堆栈中，首尾 **Bigram** 完全相同的候选译文 (Candidate)，可以合并成一个结点。

实验

- Baseline: Pharaoh (Koehn et al., 2004)
- 实验系统: Lynx
- 训练语料: 31, 149 句子对
含843 K 汉语词和949 K 英语词
- 开发集: 2002 NIST 汉英测试数据的一部分
(571 of 878 sentences)
- 测试集: 2005 NIST 汉英测试数据
(1,082 sentences)

训练过程



实验环境

- 评测工具: mteval-v11b.pl
- 语言模型工具: SRI Language Modeling Toolkits (Stolcke, 2002)
- 显著性测试工具: Zhang et al., 2004
- 汉语句法分析: Xiong et al., 2005
- 最小错误率训练工具: optimizeV5IBMBLEU.m (Venugopal and Vogel, 2005)

实验结果

System	Features	BLEU4
Pharaoh	$d + \phi(e f)$	0.0573 ± 0.0033
	$d + \text{lm} + \phi(e f) + \text{wp}$	0.2019 ± 0.0083
	$d + \text{lm} + \phi(f e) + \text{lex}(f e) + \phi(e f) + \text{lex}(e f) + \text{pp} + \text{wp}$	0.2089 ± 0.0089
Lynx	h_1	0.1639 ± 0.0077
	$h_1 + h_6 + h_7$	0.2100 ± 0.0089
	$h_1 + h_2 + h_3 + h_4 + h_5 + h_6 + h_7$	0.2178 ± 0.0080

Comparison of Pharaoh and Lynx with different feature settings

Lynx achieves an absolute improvement of **0.9%** (4.3% relative) over Pharaoh in terms of BLEU score. This difference is statistically significant ($p < 0.01$).

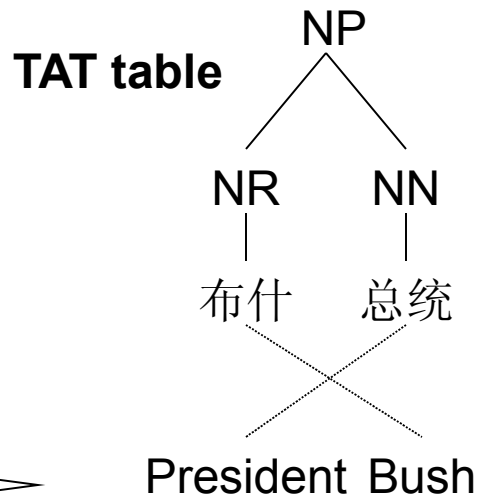
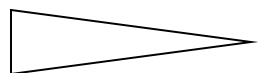
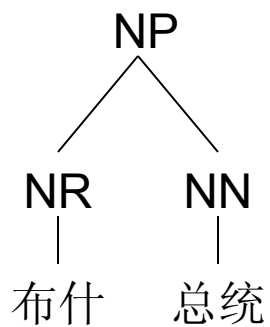
与短语的兼容性

- **TAT** 模型与短语模型是不兼容的
 - 句法短语可以表示为 **TAT**
 - 非句法短语无法表示为 **TAT**
- 实验表明，非句法短语对于提高系统性能有重要作用
- 即使对于句法短语，由于句法分析不可靠（对于同一个短语的分析有时正确有时错误），也会造成 **TAT** 概率估计上的不准确
- 设想：利用双语短语（**BP**）可以改进 **TAT** 模型的性能

利用句法短语修正 TAT 的概率

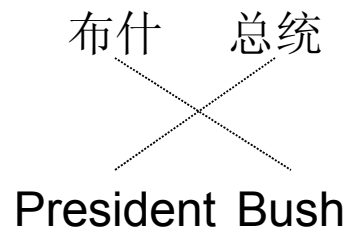
- 理由：
 - 句法分析是不可靠的，对于同一个短语，可能有时分析正确，有时分析错误，这样会导致 TAT 概率估计上的不准确
- 做法：
 - 把句法短语（SBP）的四个概率与相应的 TAT 的四个概率进行比较，用其中较大者取代 TAT 原来的概率

例子



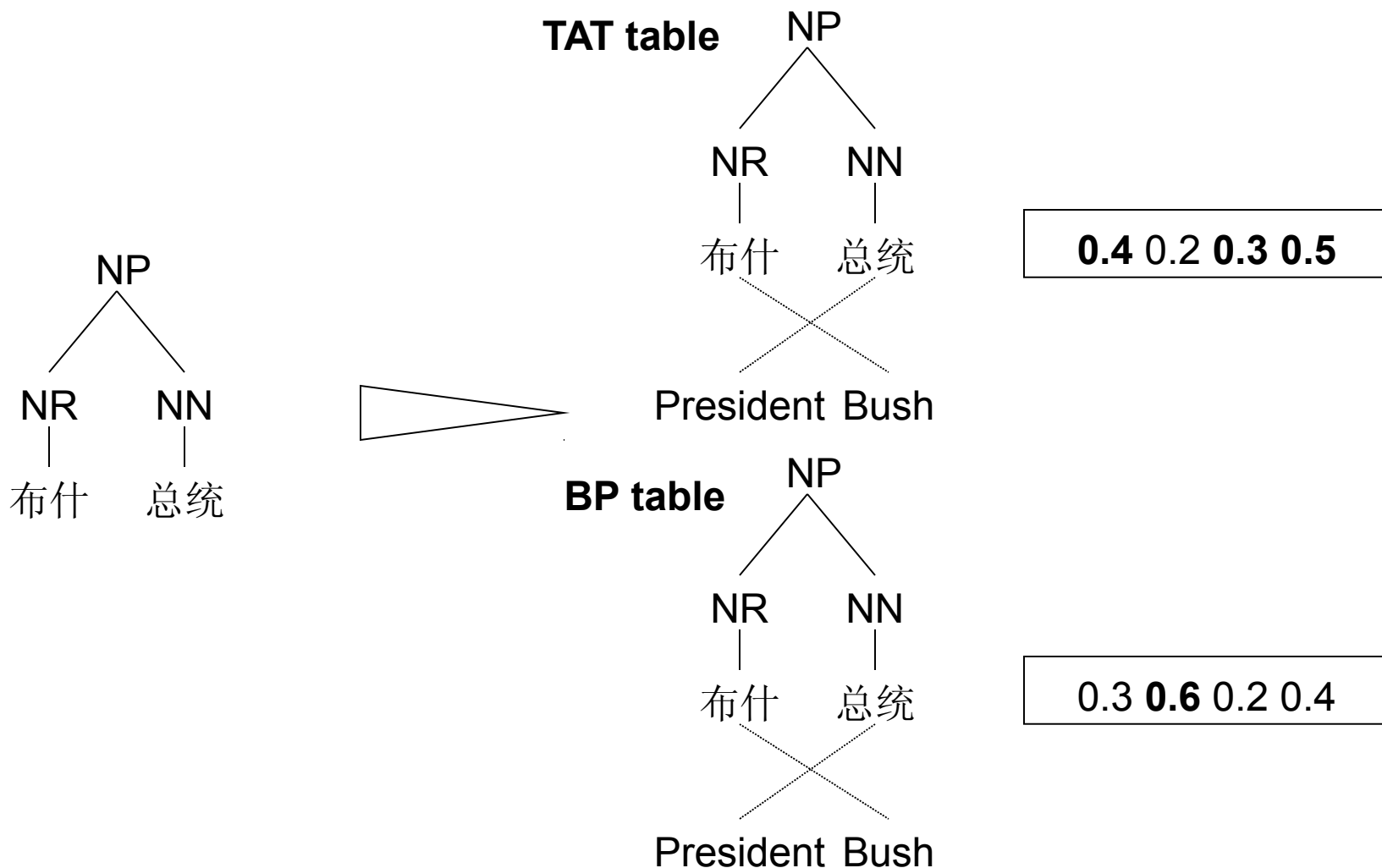
0.4 0.2 0.3 0.5

BP table

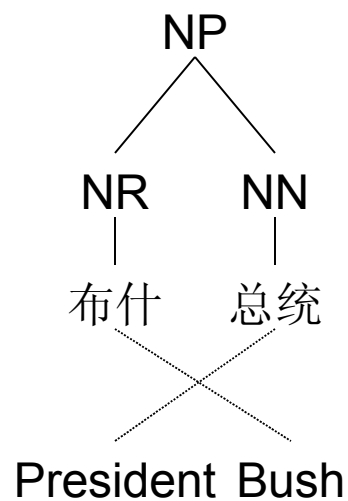
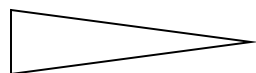
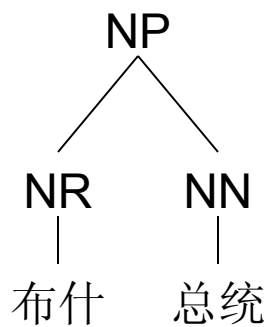


0.3 0.6 0.2 0.4

例子



例子



0.4 0.6 0.3 0.5

采用句法短语修正 TAT 概率的效果

	BLEU4
tat	0.2178 \pm 0.0080
tat + bp	0.2240 \pm 0.0083

Effect of Using Bilingual Phrases for Lynx

Using bilingual phrases brings an absolute improvement of **0.6%** in terms of BLEU score

利用非句法双语短语改进译文流利度

Problem with Lynx:

国际足联将严惩足球场上的欺骗行为

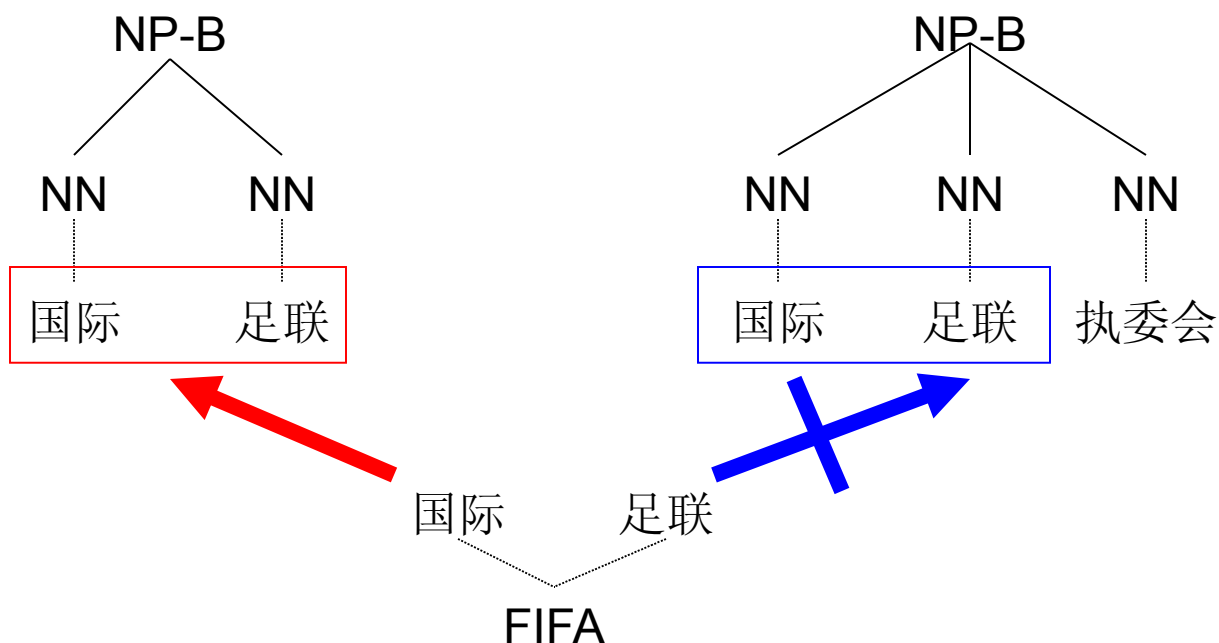
FIFA will severely punish cheat behaviour on the football field

国际足联执委会还宣布了一些改革措施。

international 足联 Executive Committee also announces that some reform measures.

How could this happen?

两棵句法树

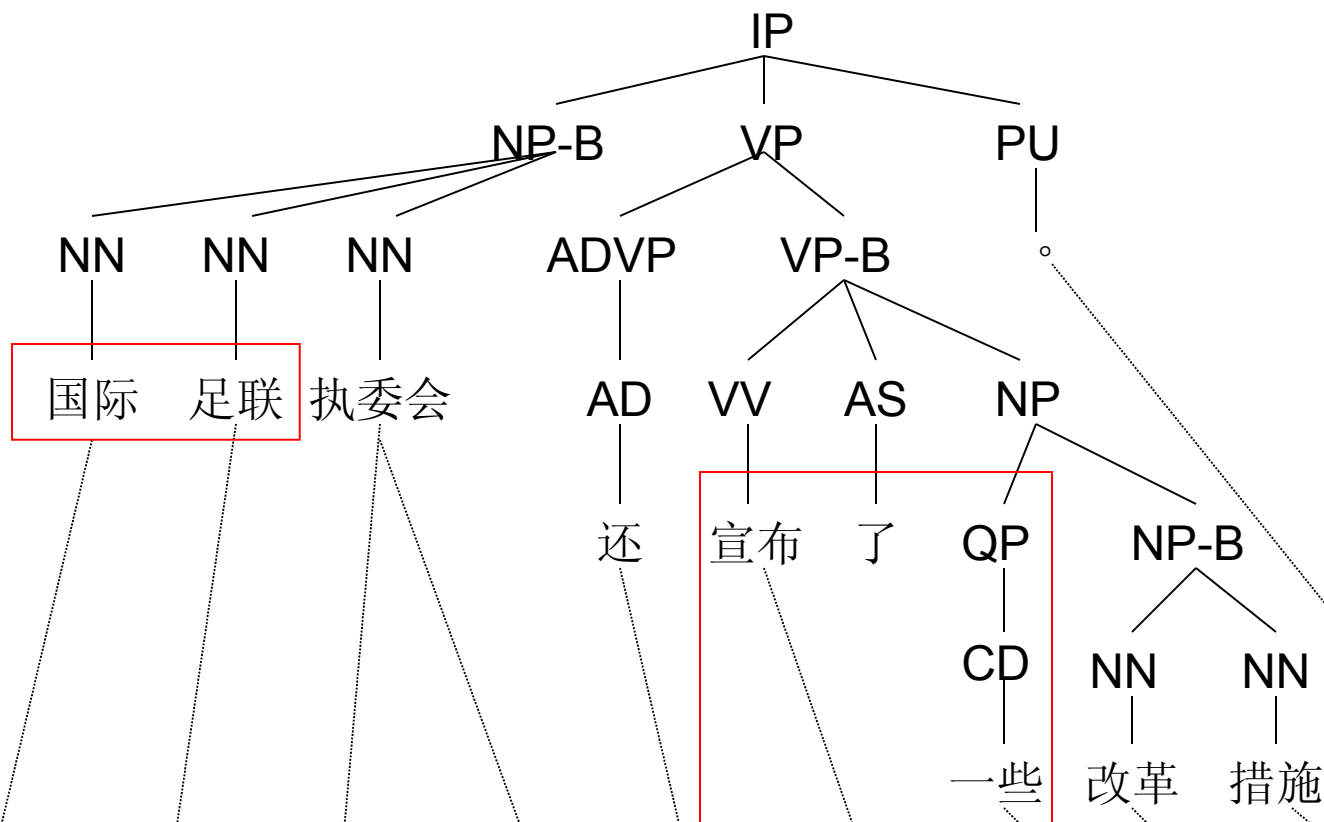


the strength of BPs is restricted!

解决方法

- 搜索结束后，在翻译结果上，根据词语对齐，将译文中的短语替换成流利度更高的短语（根据语言模型分值）
- 如果有多个候选译文，每个译文都可以进行上述替换，这时可以通过计算调整后的总的翻译概率（同时考虑语言模型、翻译模型和其他特征），选择分数最高的候选译文

例子

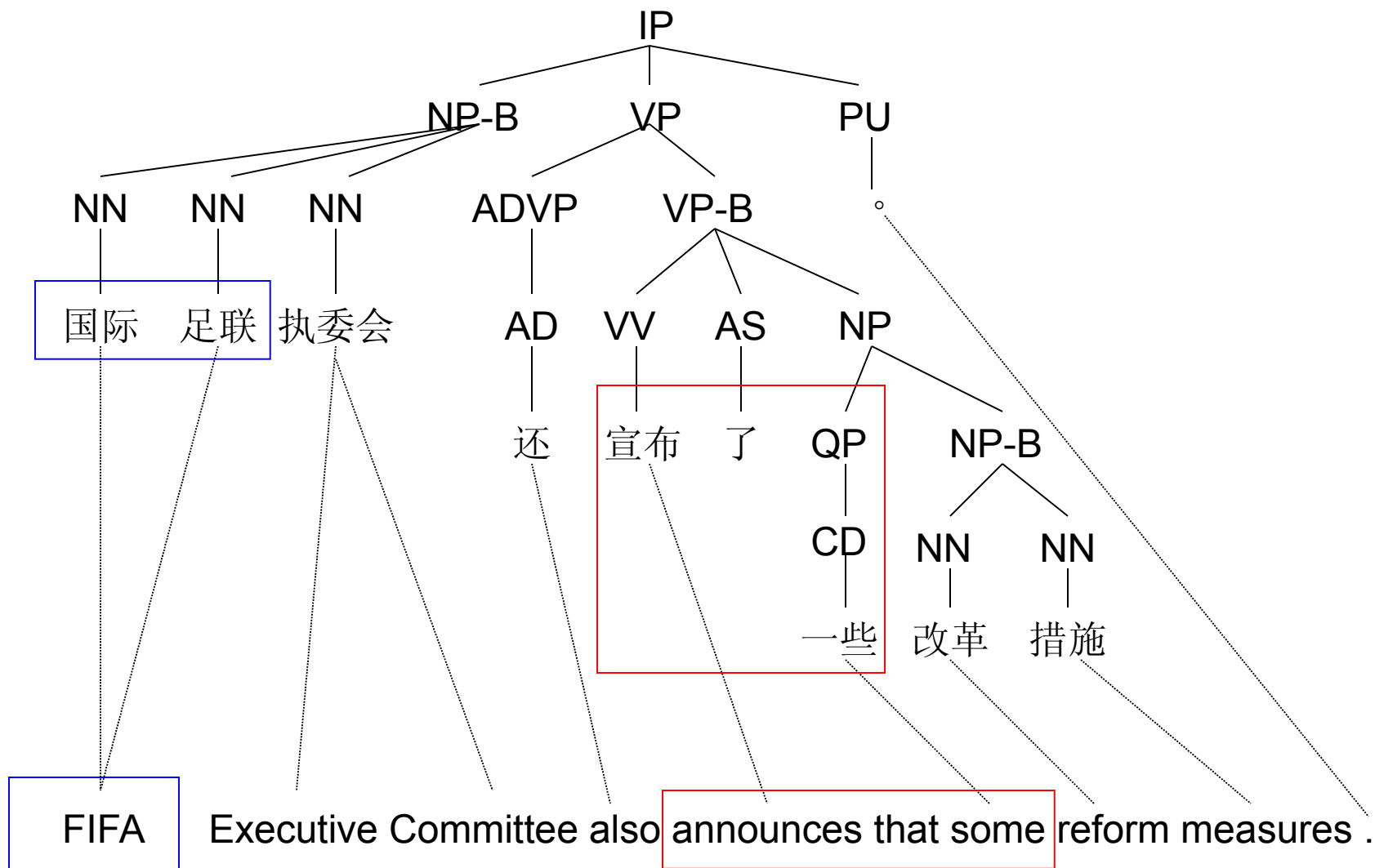


international 足联

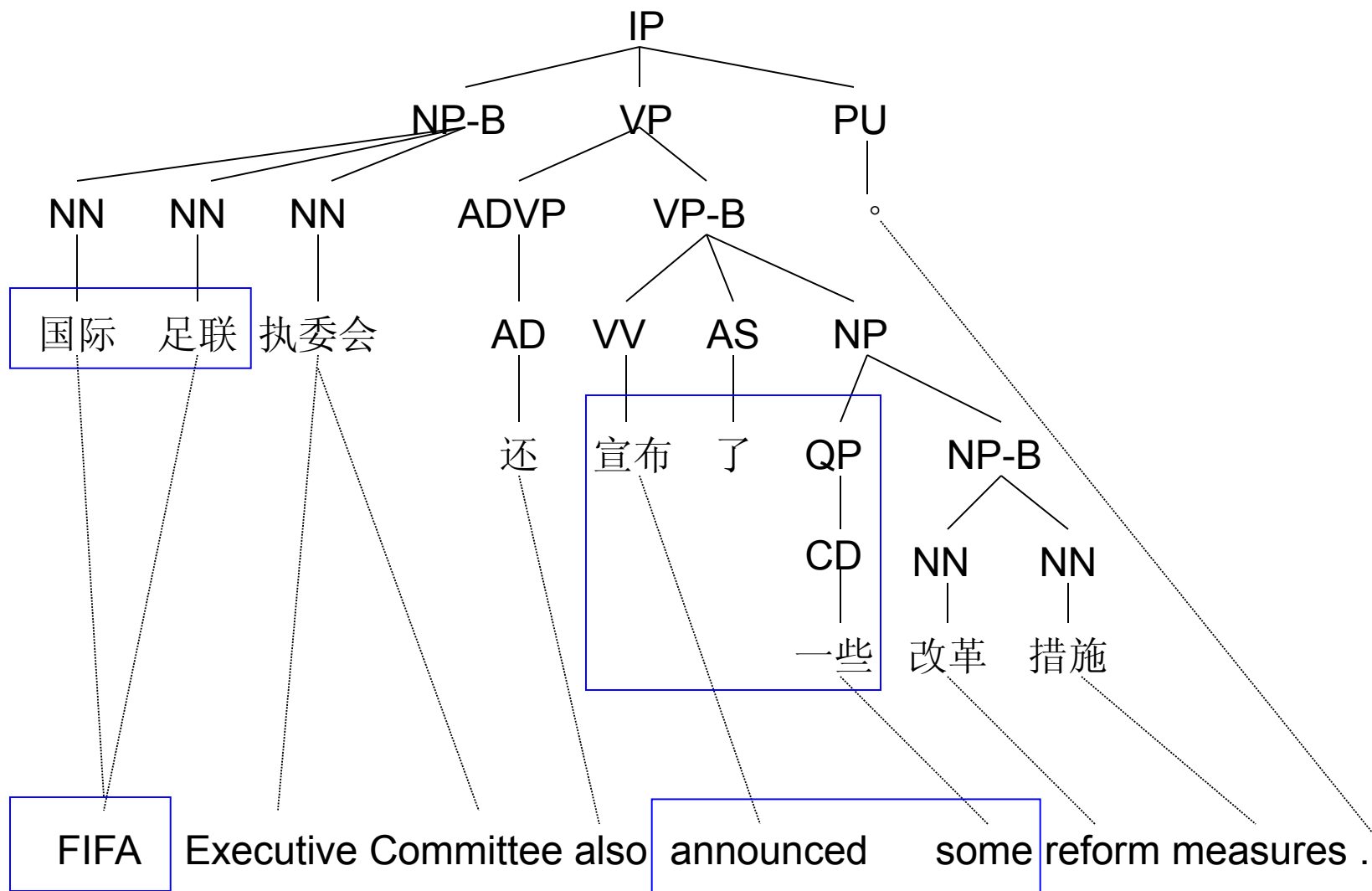
Executive Committee also

announces that some reform measures .

例子



例子



加大数据规模

- Bilingual corpus (train BPs and TATs)
 - 2.6M sentence pairs (68.1M Chinese words and 73.8M English words)
 - Use all the data to obtain BPs and a portion of 800K pairs to obtain TATs
- Monolingual corpora (train LM)
 - English side of the bilingual corpus (73.8M words)
 - Xinhua portion of Gigaword corpus (181M words)

加大数据规模后的实验结果

Results of Lynx on test set with various settings.

Training Data (pairs)		Language Model		Improve Fluency	BLEU4
TAT	BP	Data (words)	Order		
31K	-	949K	one 3-gram	No	0.2178
31K	31K	949K	one 3-gram	No	0.2240
31K	800K	73M	one 3-gram	No	0.2431
800K	2.6M	73M	one 3-gram	No	0.2692
800K	2.6M	73M 181M	two 3-gram	No	0.2934
800K	2.6M	73M 181M	two 4-gram	No	0.3047
800K	2.6M	73M 181M	two 4-gram	Yes	0.3184

小结

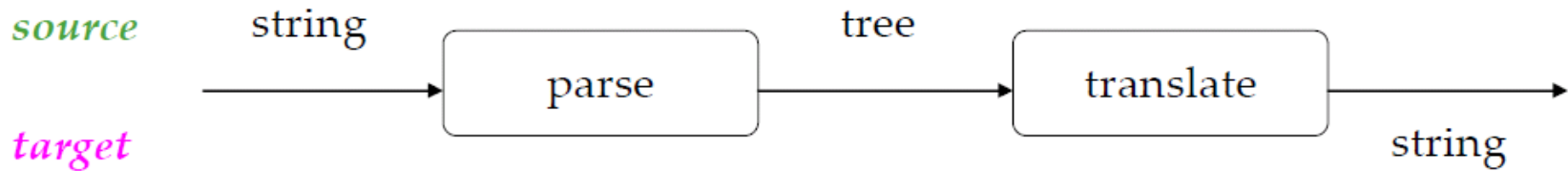
- 基于树到串对齐模板的翻译模型
 - 一种树到串模型
 - 在源语言句法分析和词语对齐的双语语料库上抽取双语对齐模板（**TAT**），构建翻译模型
 - 解码时先进行源语言句法分析，然后自底向上依次对树的每个结点构造候选译文
- 模型简洁直观，可以较好地利用句法信息进行重排序
- 在给定句法分析结果的情况下，解码极快
- 非句法短语兼容性不好
- 受句法分析性能影响，性能不高

基于森林的方法

Forest-based Translation

- Haitao Mi, Liang Huang and Qun Liu. Forest-Based Translation. In Proceedings of ACL 2008 Columbus, OH
- Haitao Mi and Liang Huang. Forest-based Translation Rule Extraction. In Proceedings of EMNLP 2008 ,Honolulu, Hawaii.
Nominated for the best-paper award

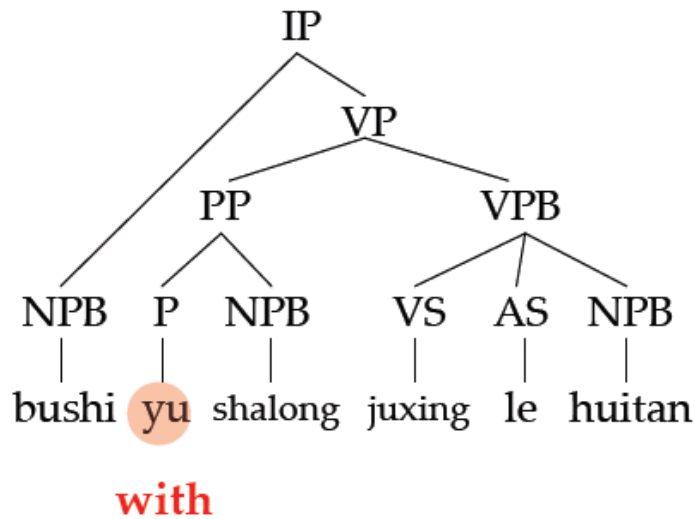
Parsing Mistake Propagation



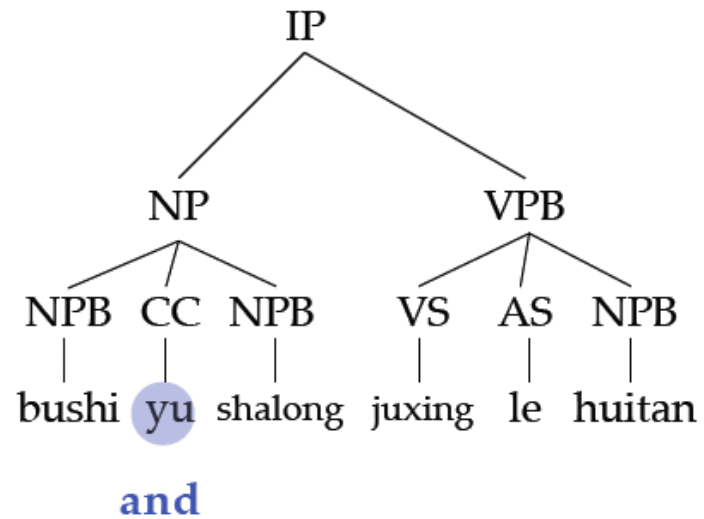
parsing mistakes potentially introduce translation mistakes!

Syntactic Ambiguity

It is important to choose a correct tree for producing a good translation!

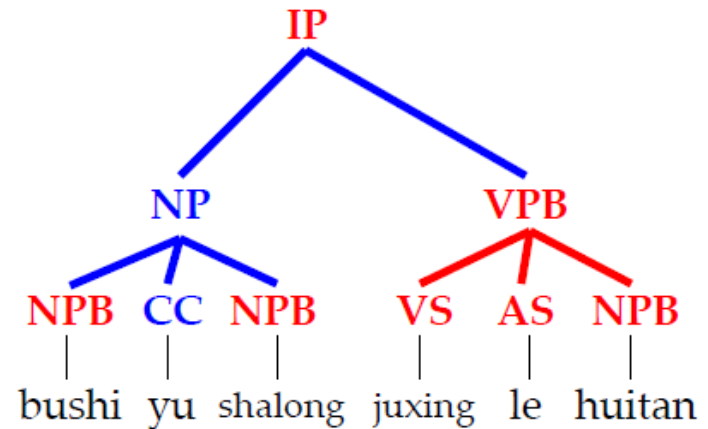
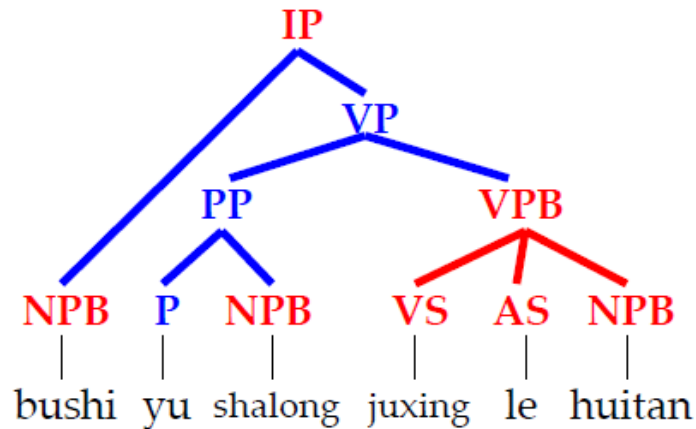


“Bush held a talk **with** Sharon”



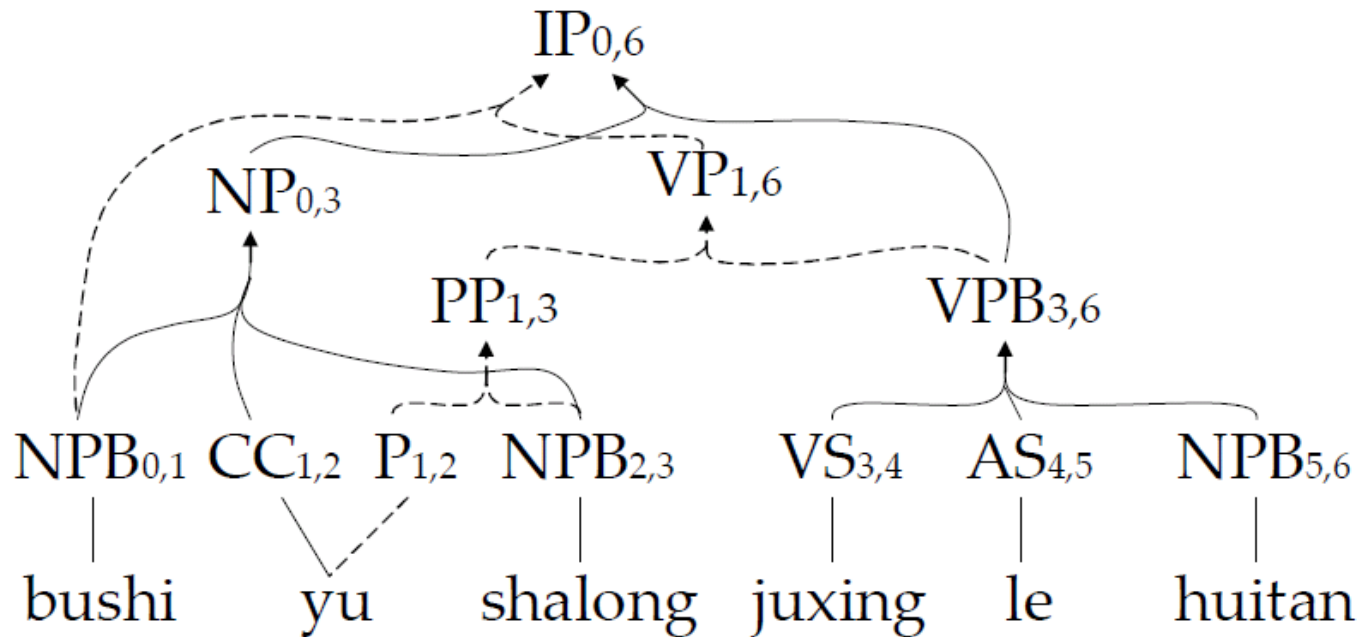
“Bush **and** Sharon held a talk”

1-best \rightarrow n-best trees?

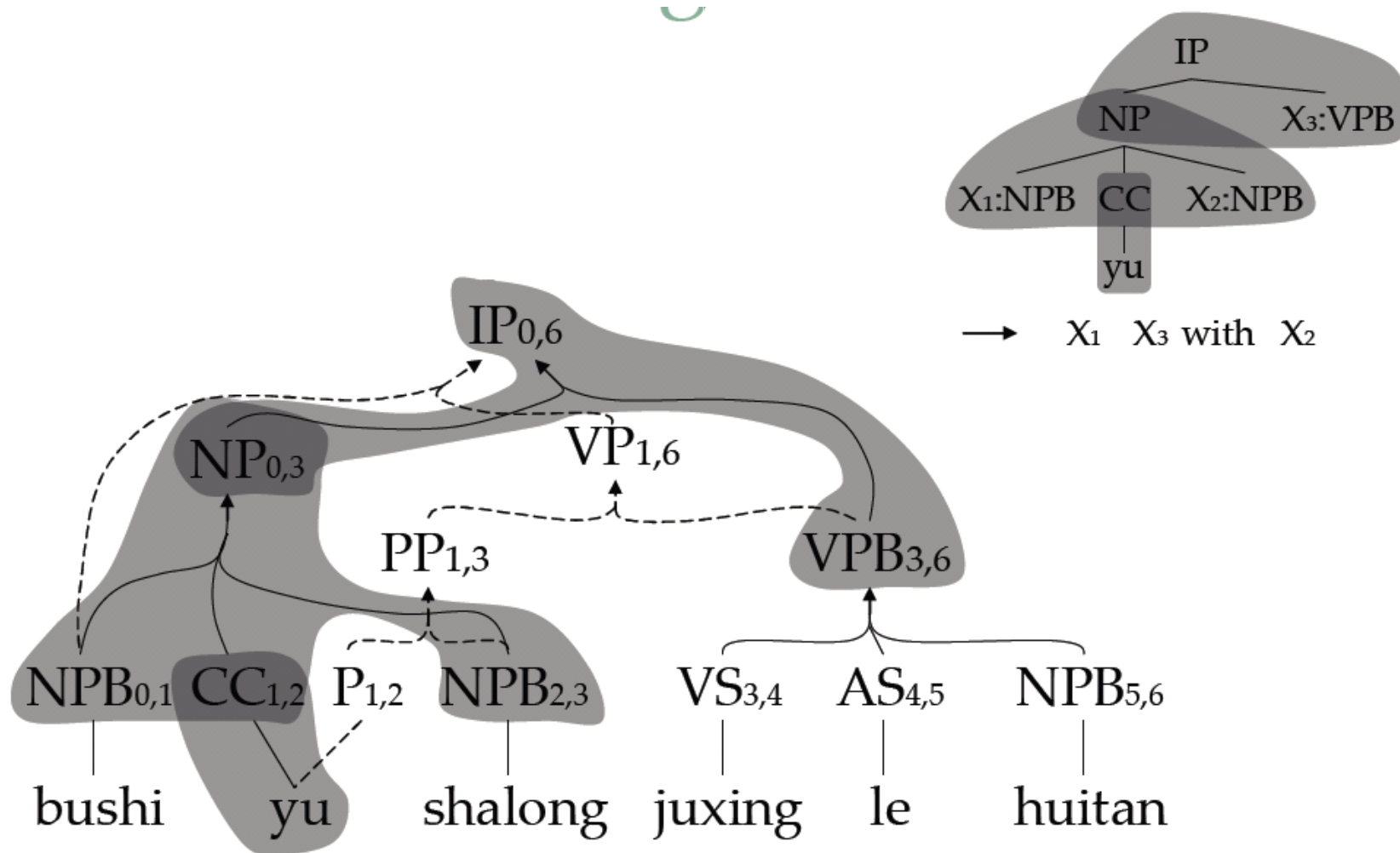


Very few variations among the *n*-best trees!

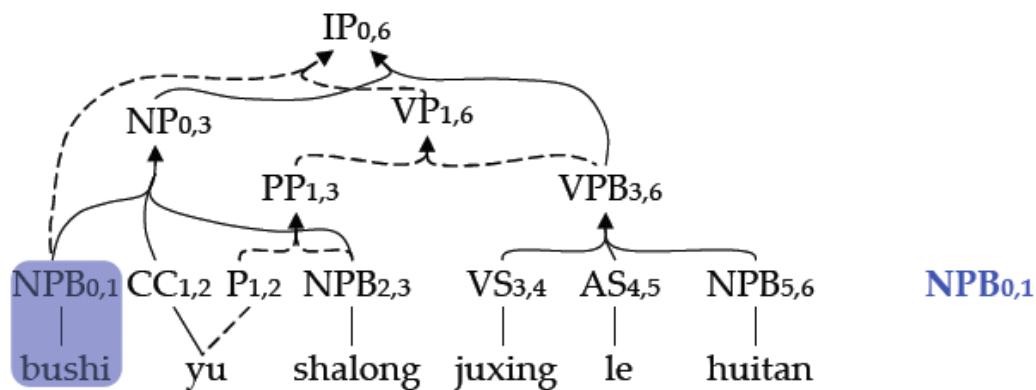
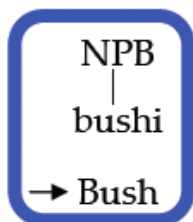
Packed Forest



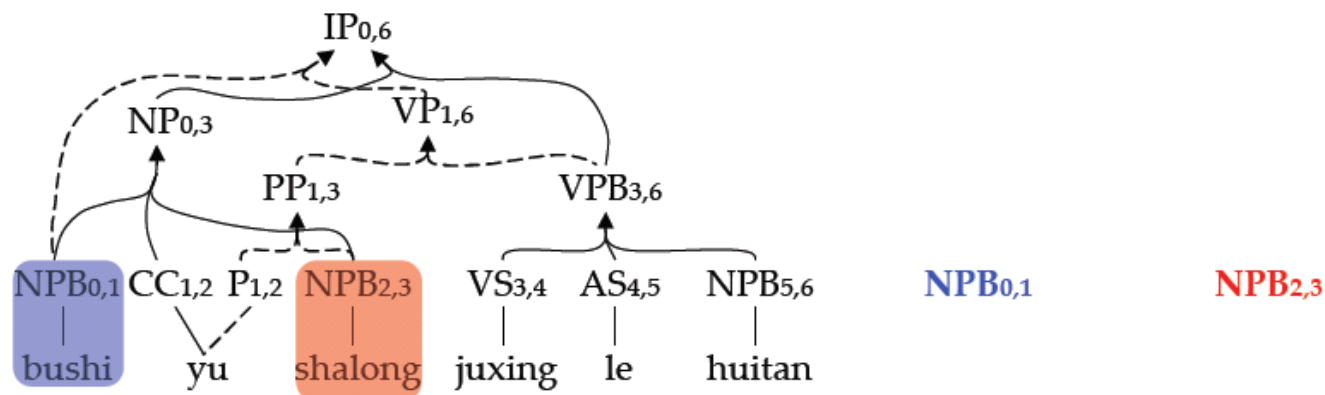
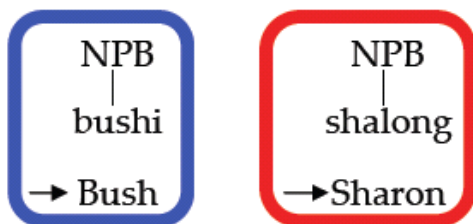
Patten Matching on Forest



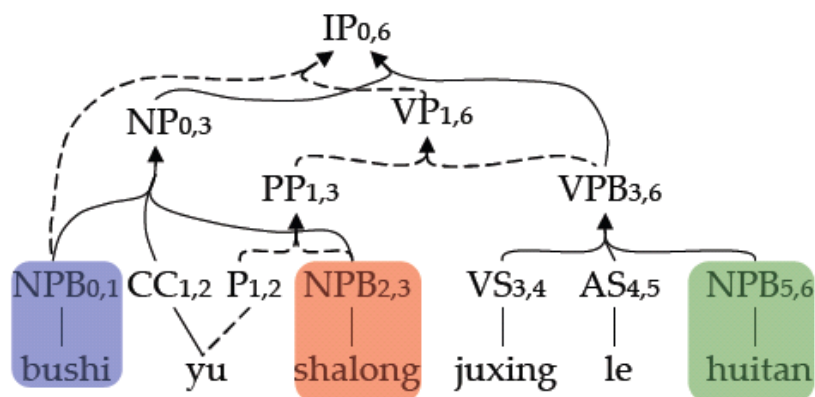
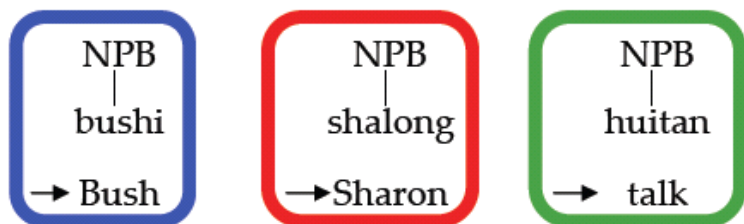
Translation Forest



Translation Forest



Translation Forest

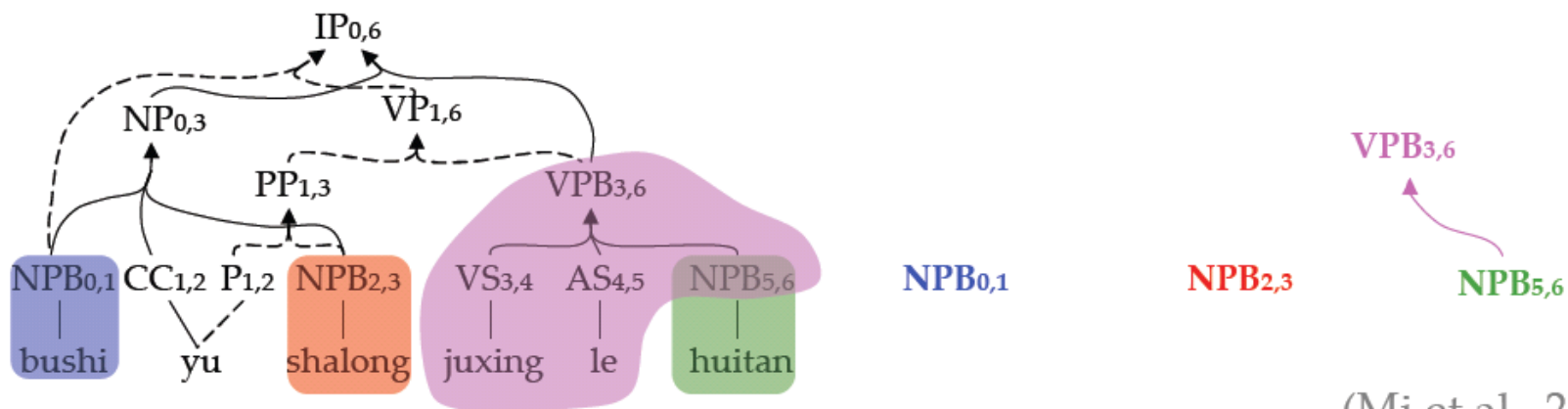
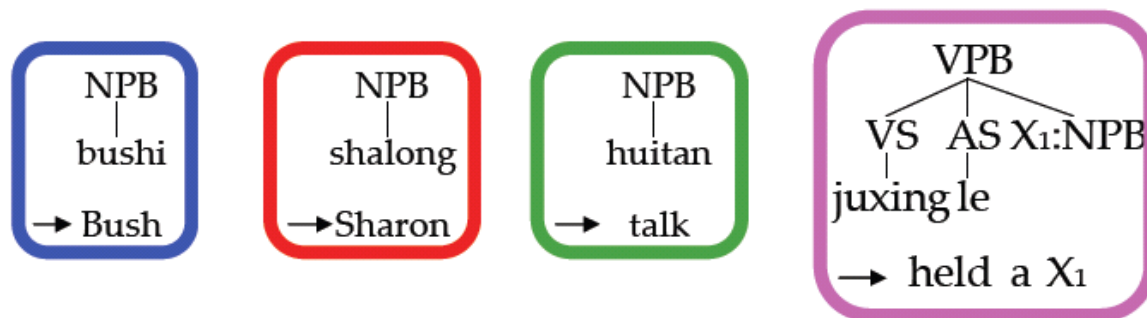


$NPB_{0,1}$

$NPB_{2,3}$

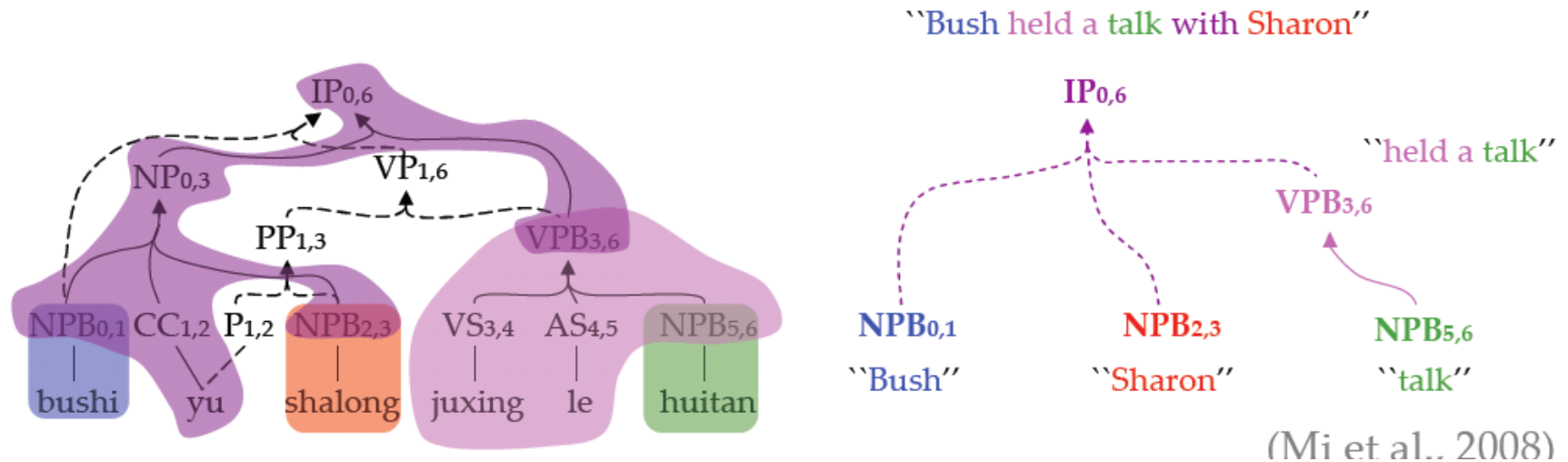
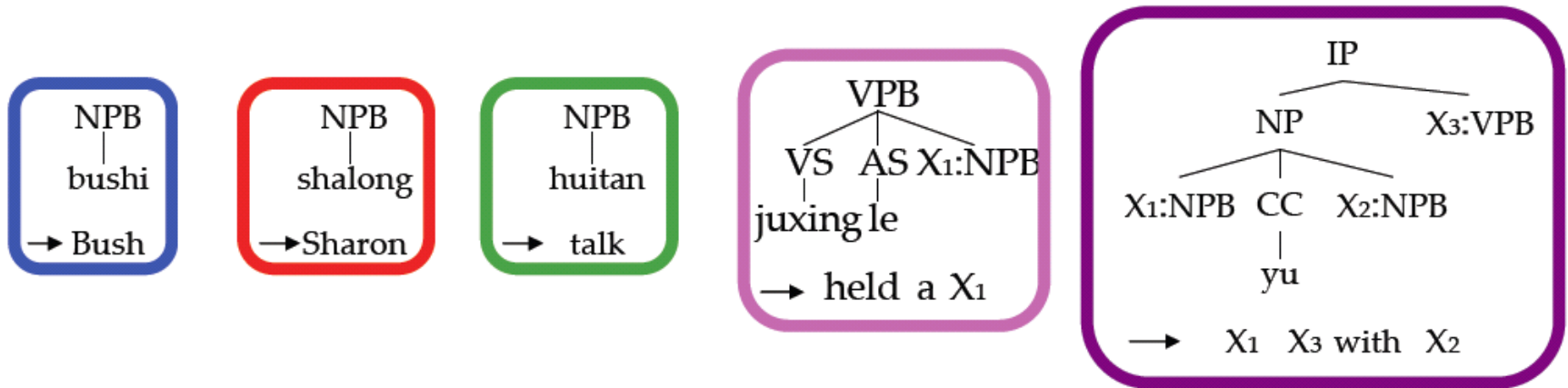
$NPB_{5,6}$

Translation Forest

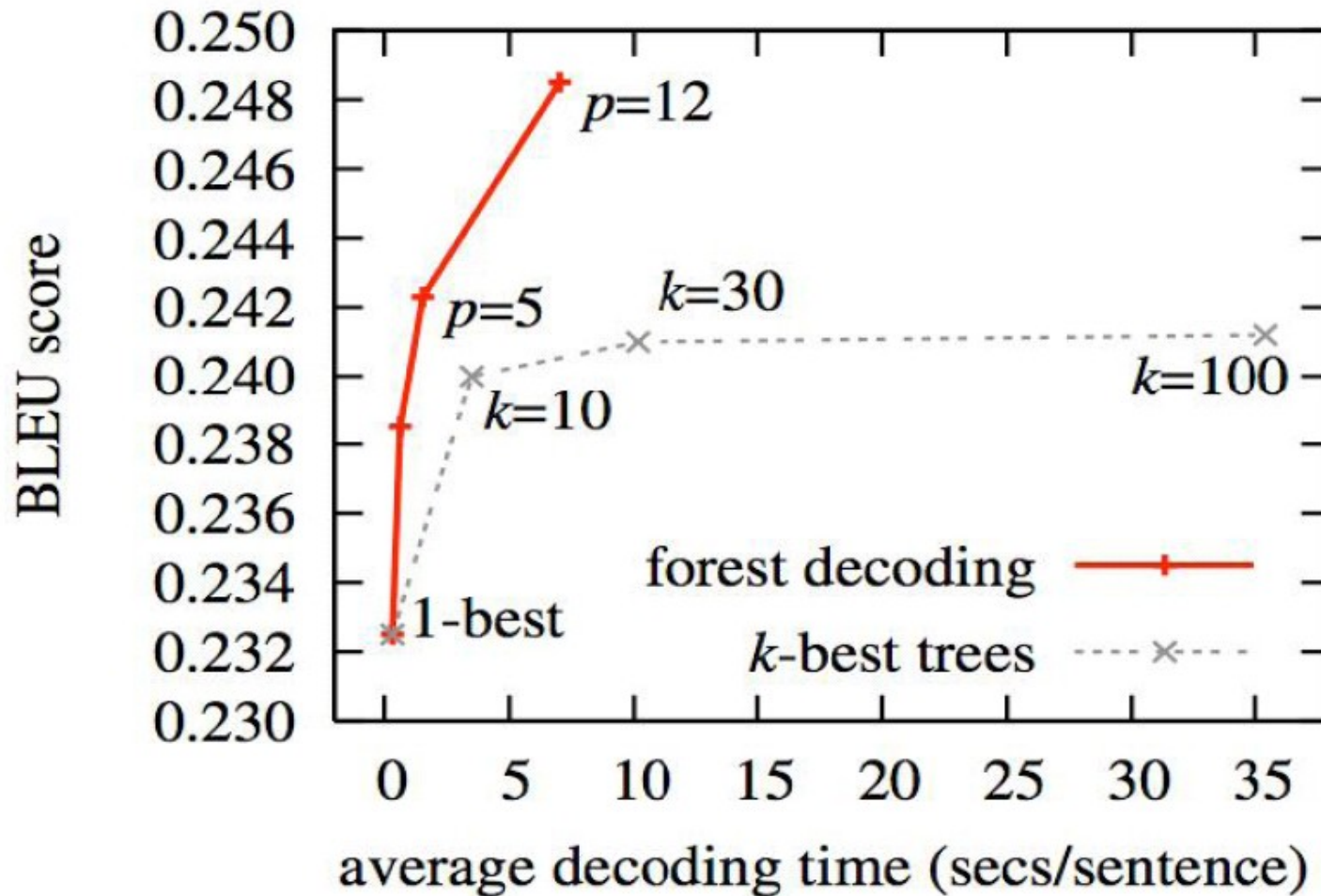


(Mi et al., 2002)

Translation Forest

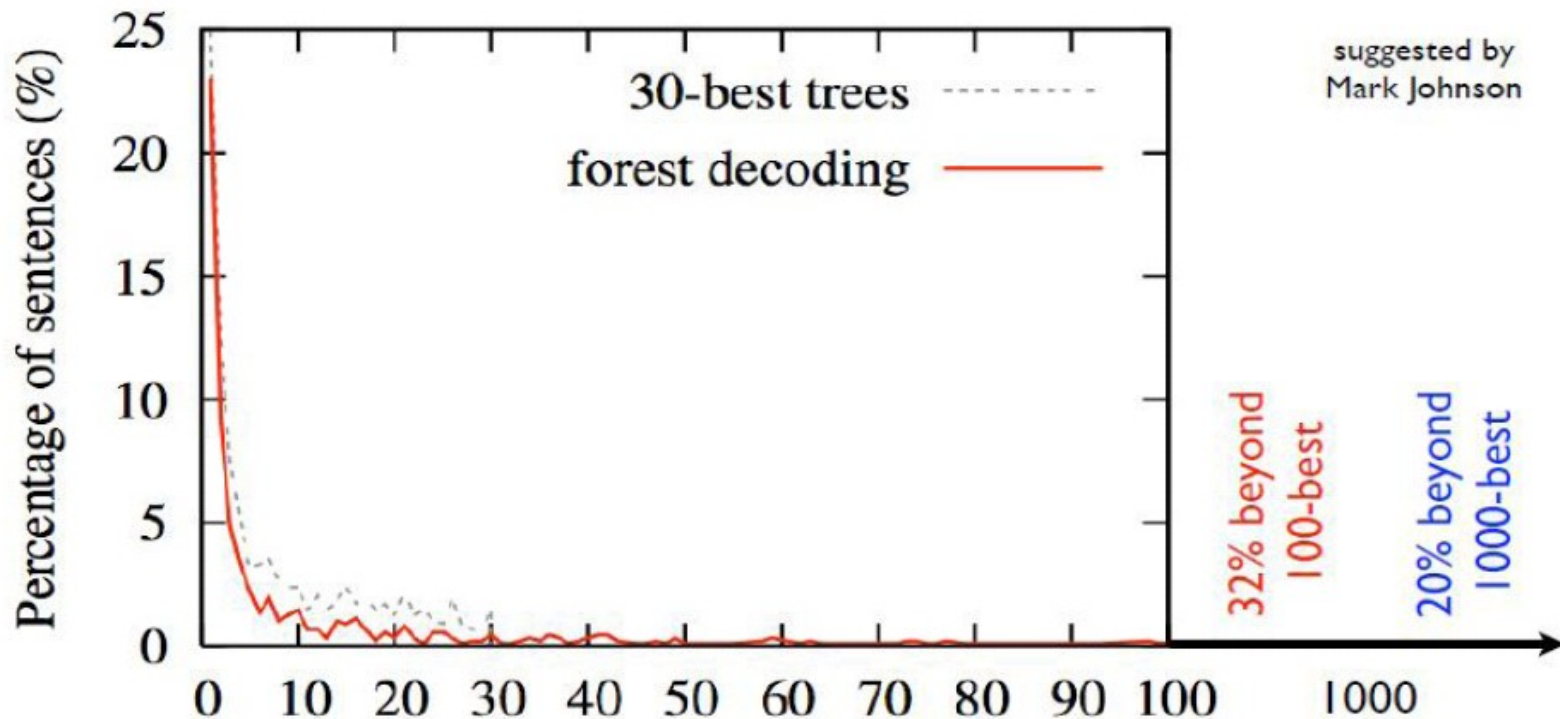


N-best Trees vs. Forest



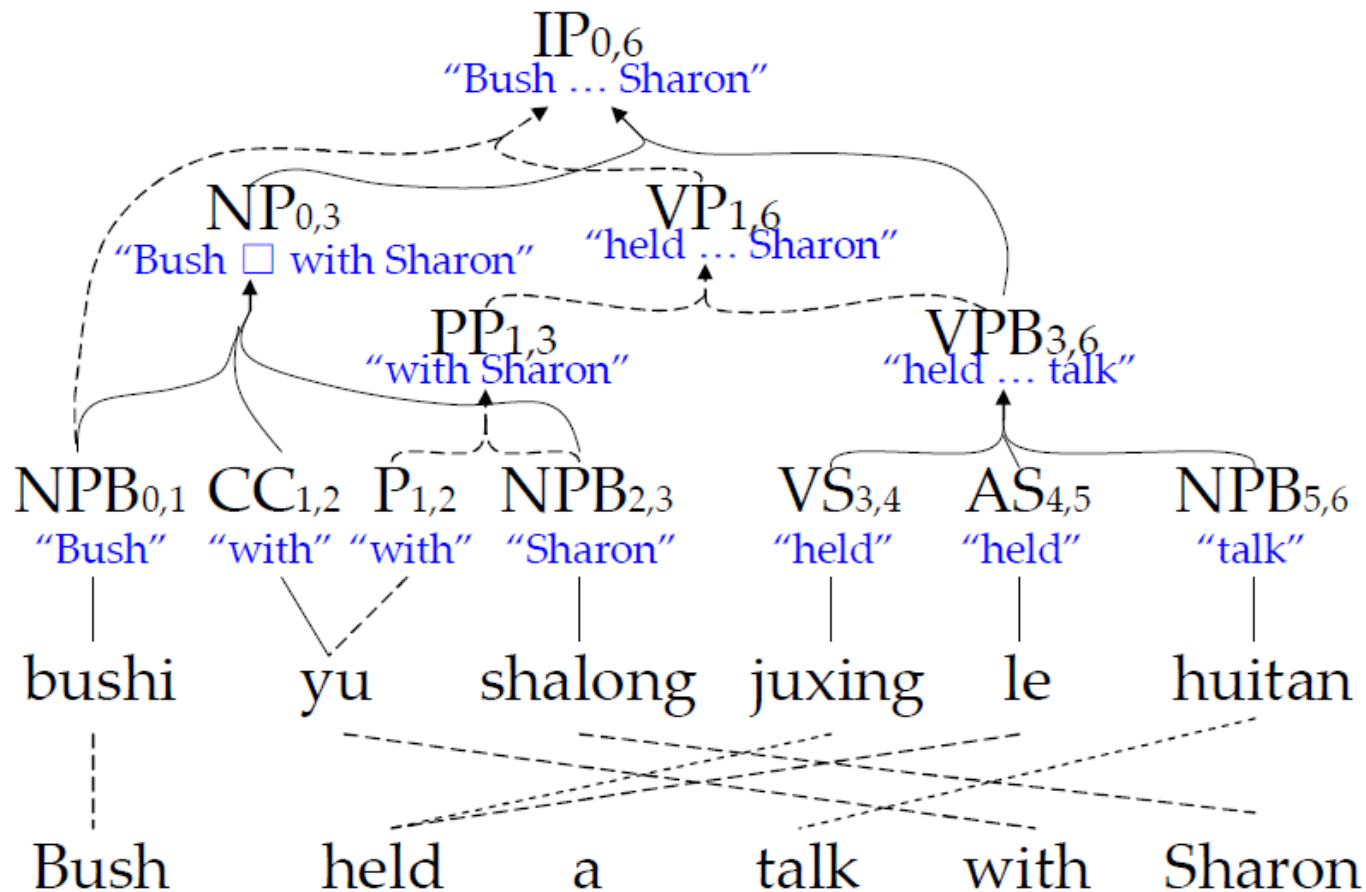
Forest as Virtual ∞ -best List

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



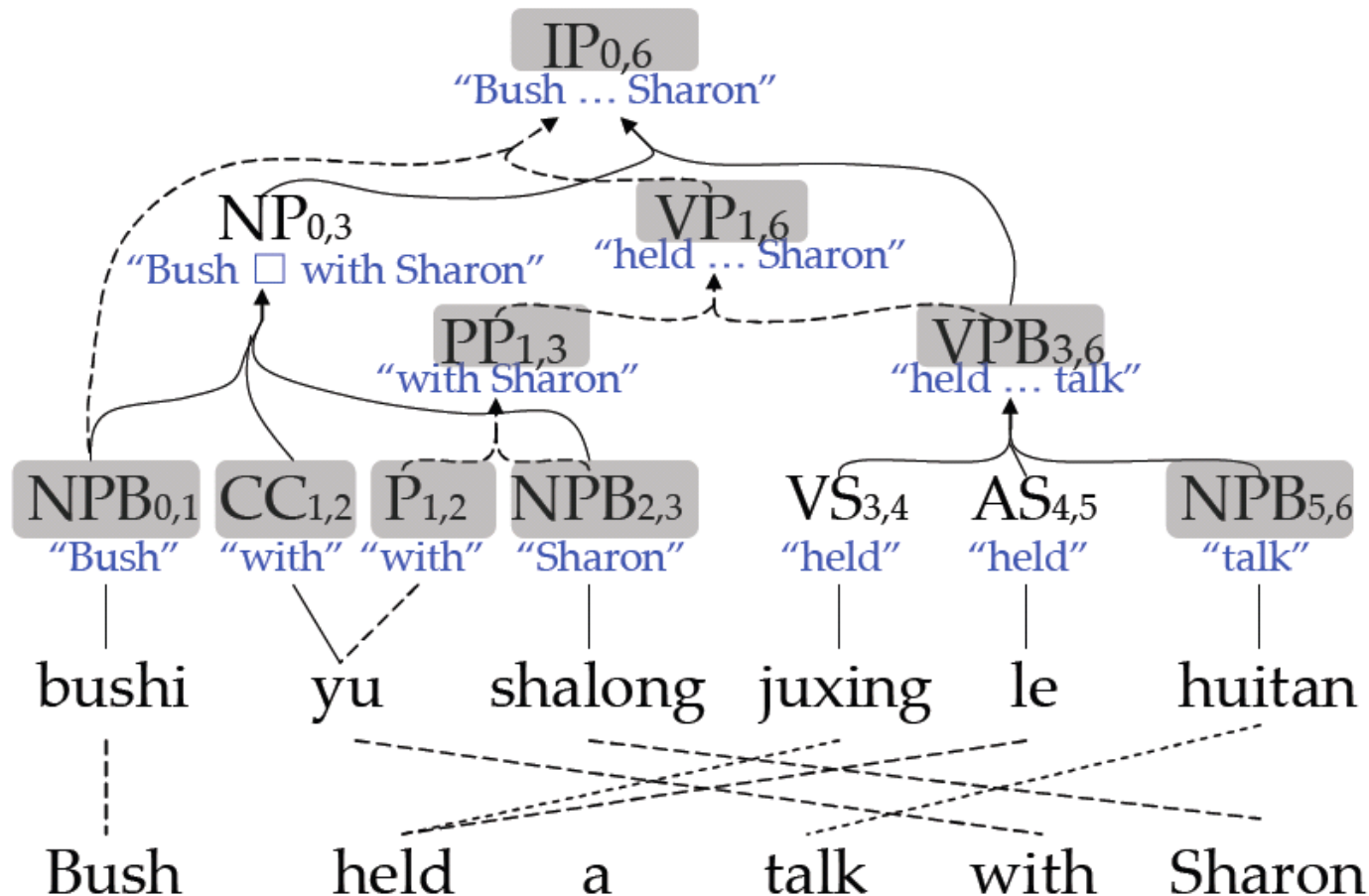
Forest-based Rule Extraction

- Compute target spans



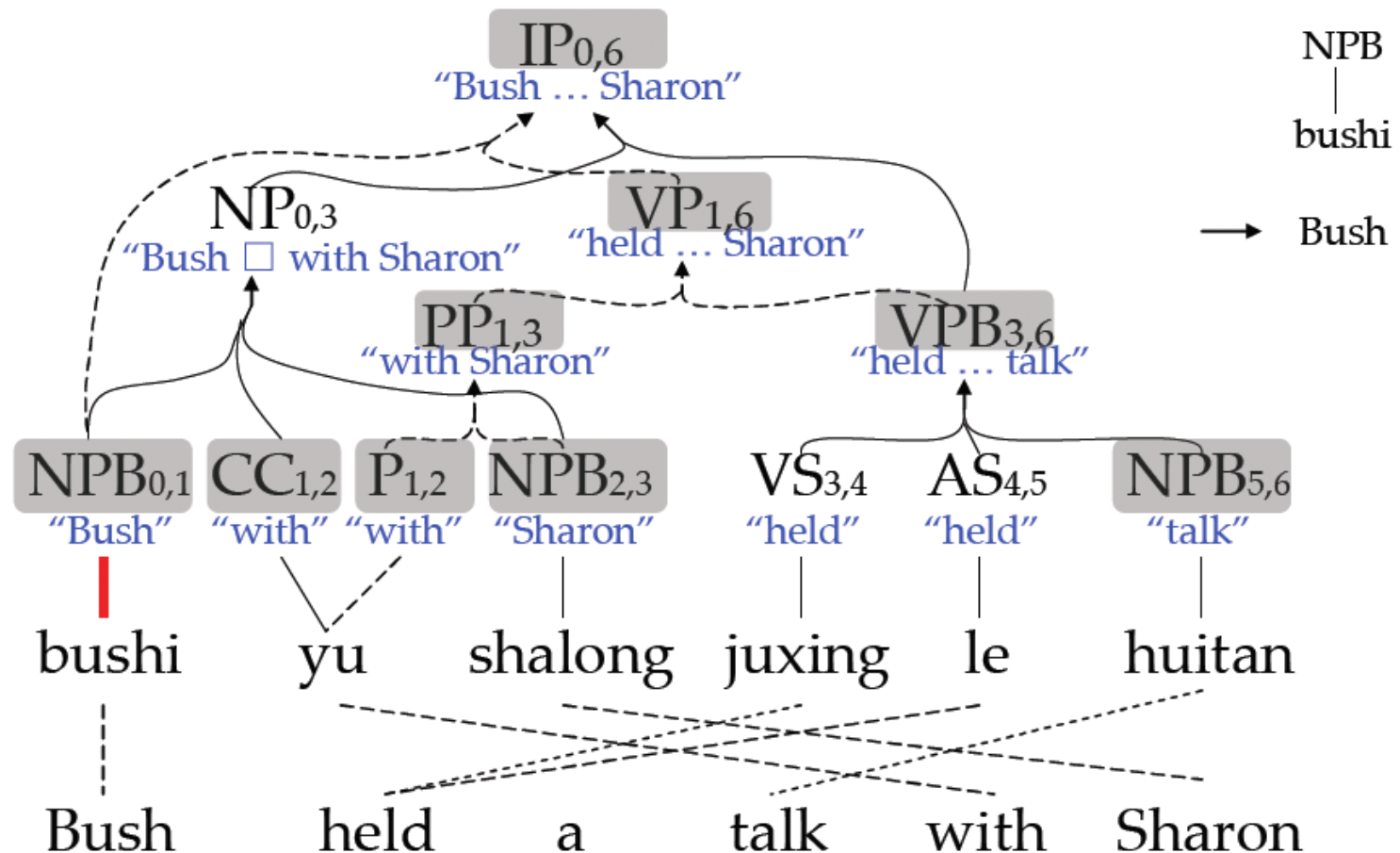
Forest-based Rule Extraction

- Compute admissible nodes



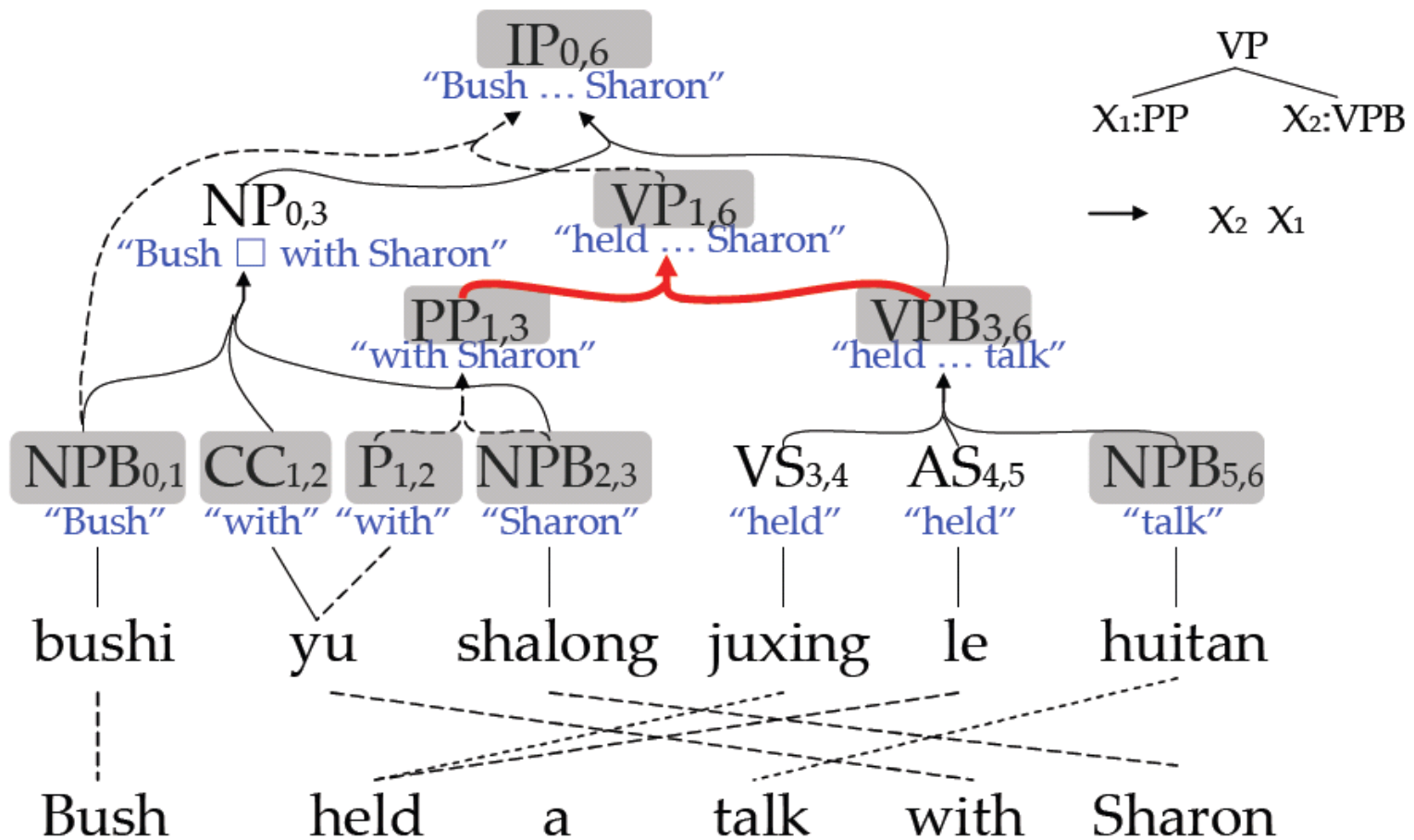
Forest-based Rule Extraction

- Extract Minimal Rules



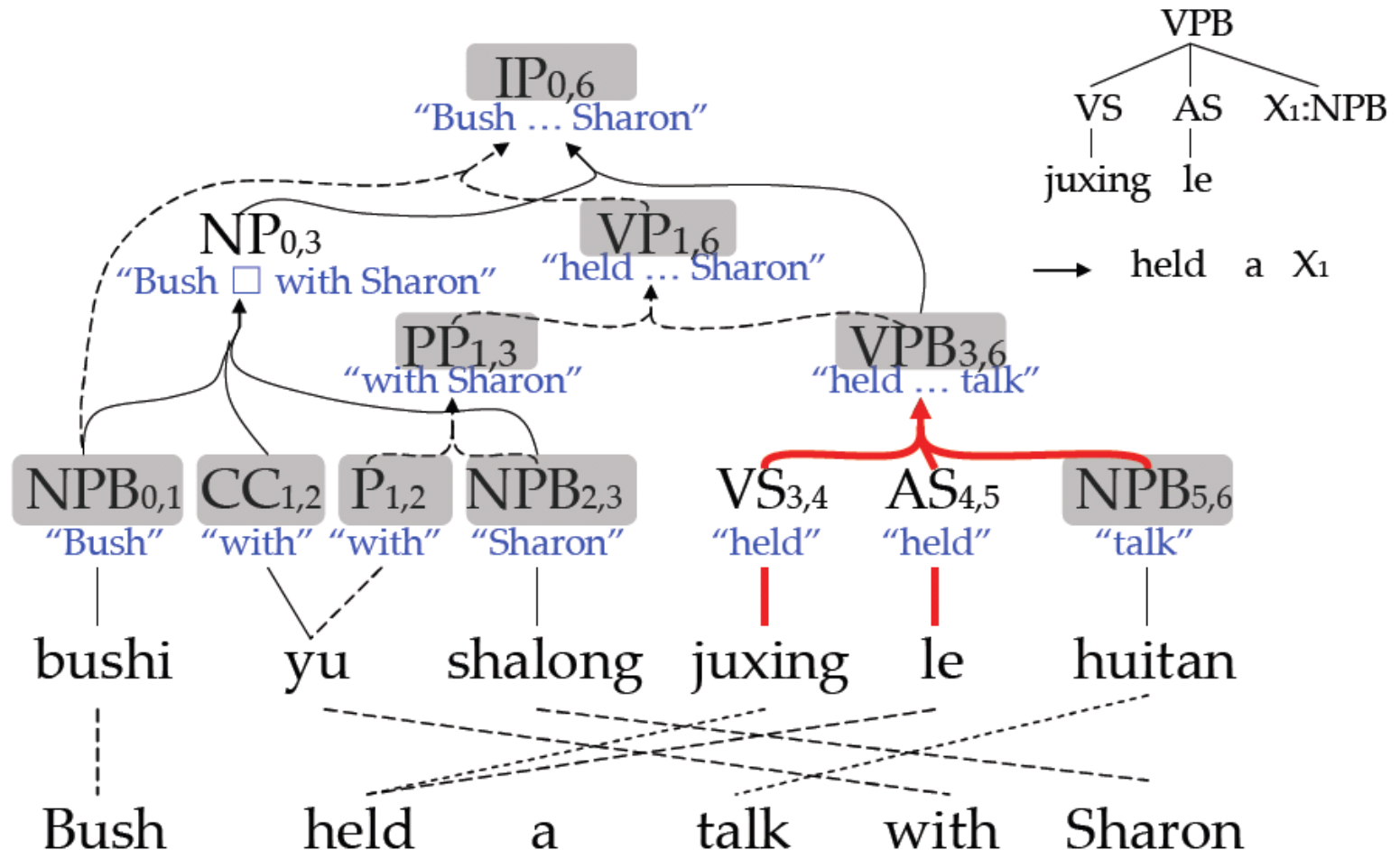
Forest-based Rule Extraction

- Extract Minimal Rules



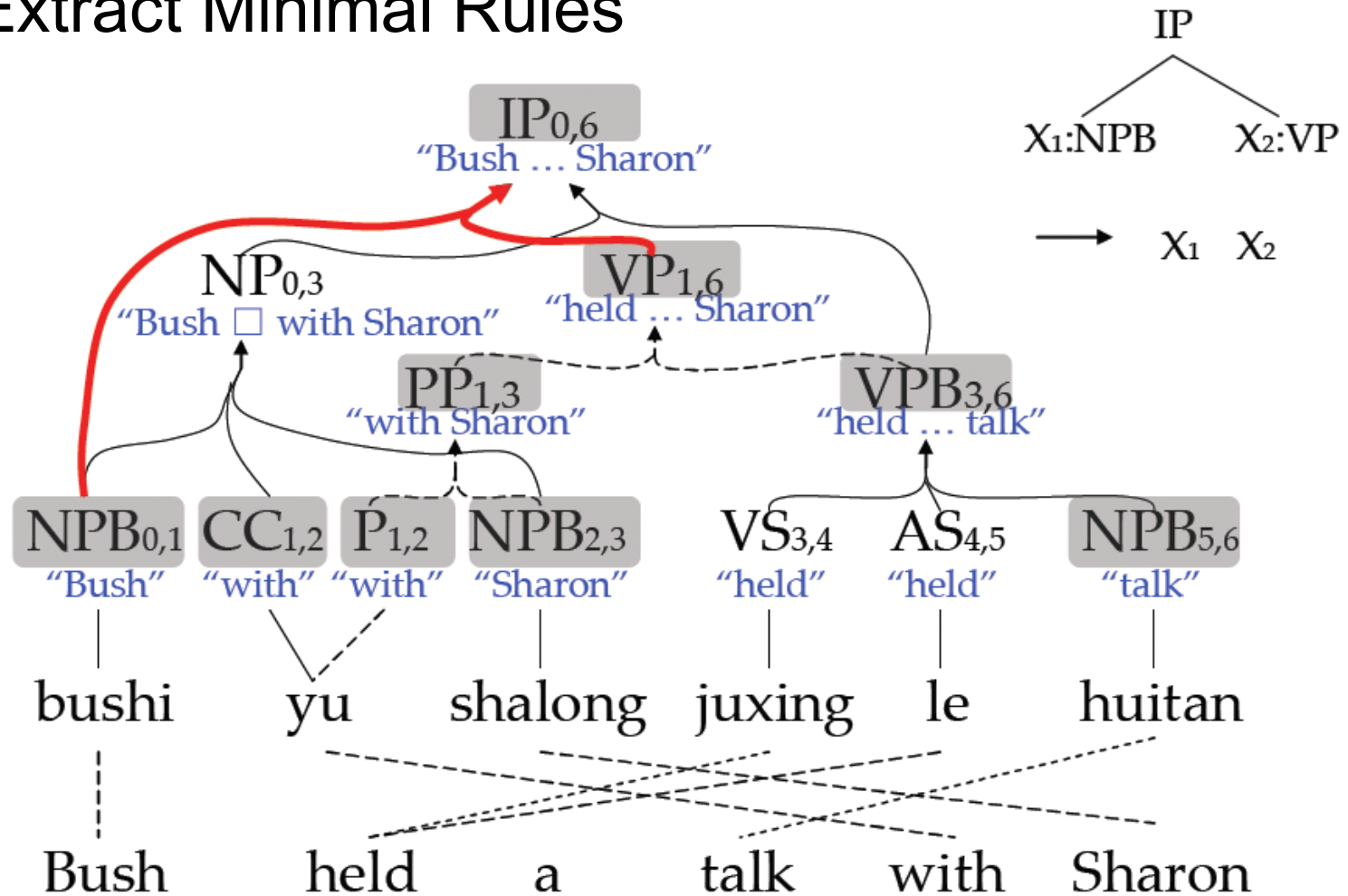
Forest-based Rule Extraction

- Extract Minimal Rules



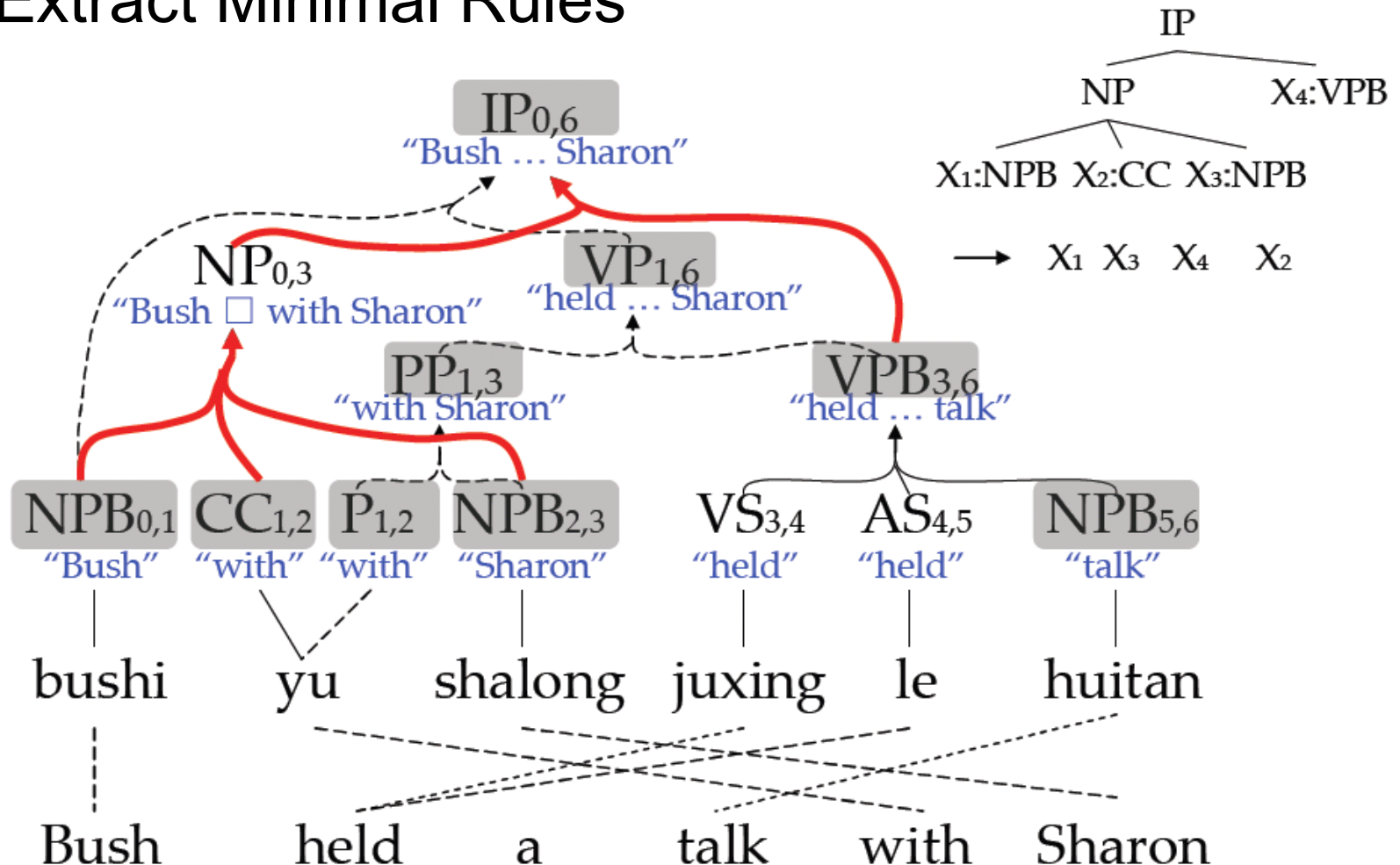
Forest-based Rule Extraction

- Extract Minimal Rules



Forest-based Rule Extraction

- Extract Minimal Rules



Rule Probabilities and Rule counts

$$P(r \mid lhs(r)) = \frac{c(r)}{\sum_{r': lhs(r')=lhs(r)} c(r')}$$

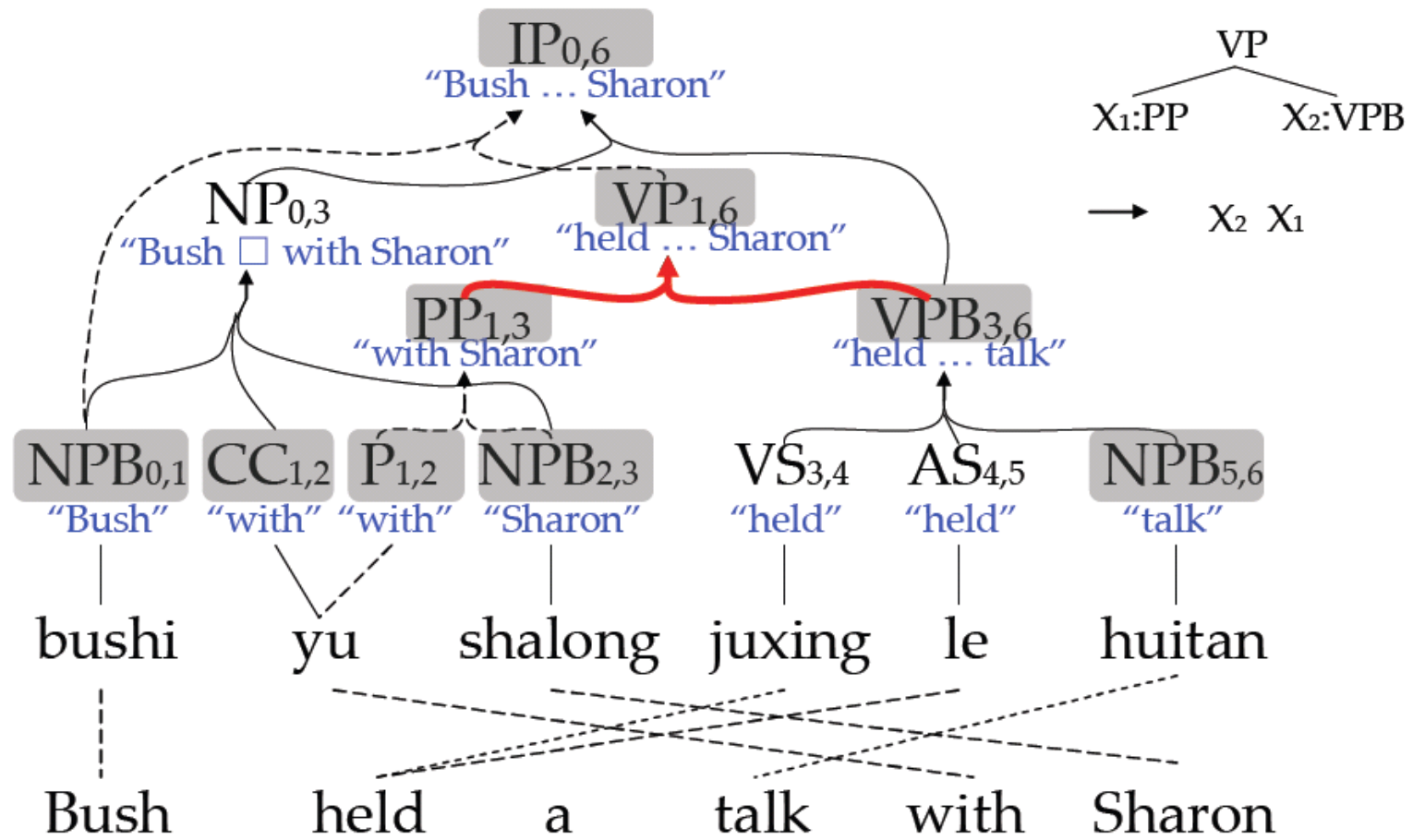
How often does a rule occur in training examples?

$$P(r \mid rhs(r)) = \frac{c(r)}{\sum_{r': rhs(r')=rhs(r)} c(r')}$$

$$P(r \mid root(lhs(r))) = \frac{c(r)}{\sum_{r': root(lhs(r'))=root(lhs(r))} c(r')}$$

Fractional Count

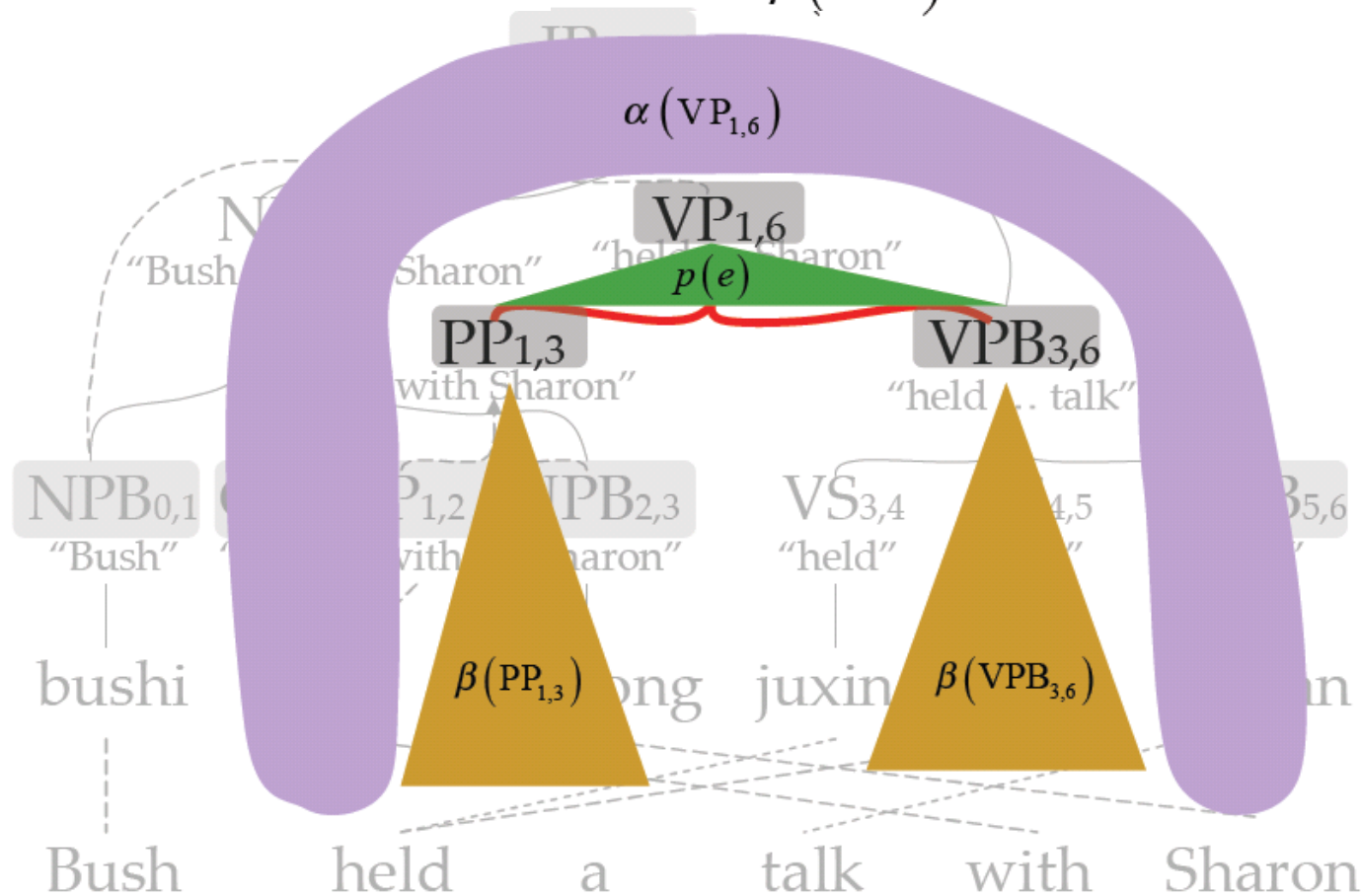
Q: What 's the count of this rule on this training example?



Fractional Count

$$\alpha\beta(\{e\}) = \alpha(\text{VP}_{1,6}) \times p(e) \times \beta(\text{PP}_{1,3}) \times \beta(\text{VPB}_{3,6})$$

$$c(r) = \frac{\alpha\beta(lhs(r))}{\alpha\beta(TOP)}$$



Results on forest training and decoding

rule extraction

decoding

	1-best tree	forest
1-best tree	0.2560	0.2674
forest	0.2679	0.2816

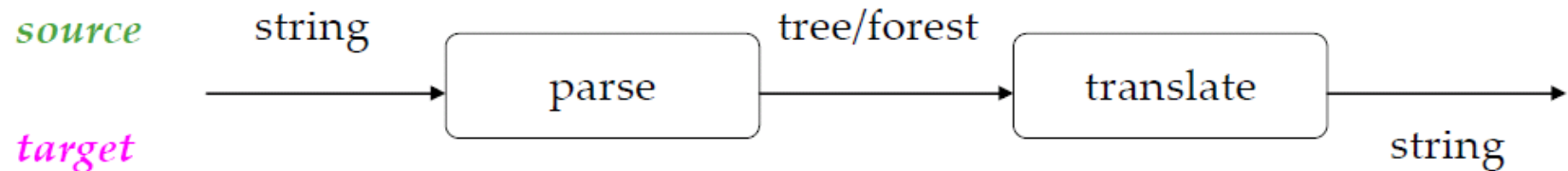
小结

基于串的方法

Joint Parsing and Translation

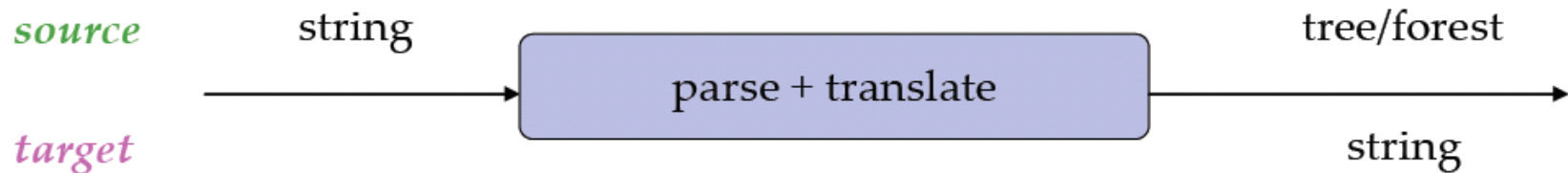
- Yang Liu and Qun Liu. 2010. Joint Parsing and Translation. In Proceedings of COLING 2010, pages 707-715, Beijing, China, August.

Seperate Parsing and Translation



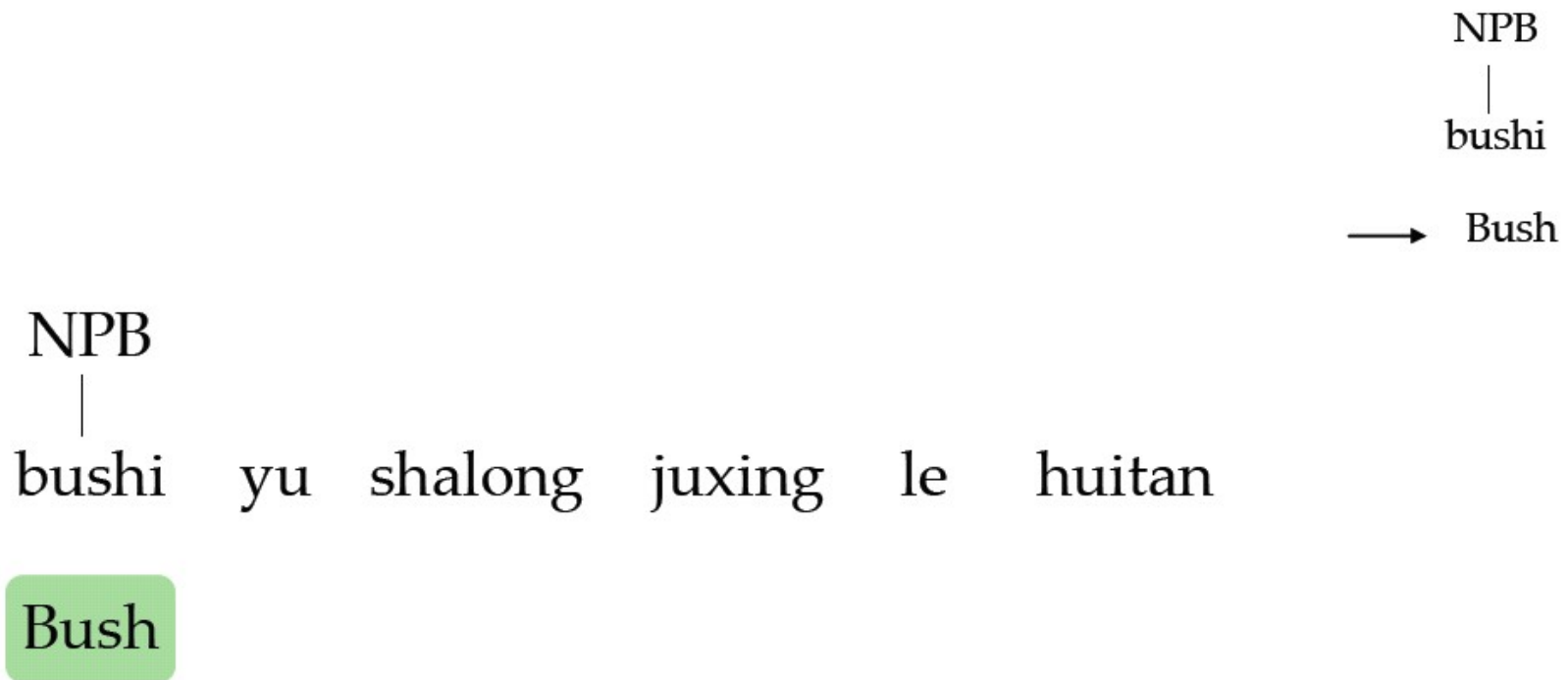
- ☺ Separate grammar for parsing and translation
- ☺ decoding is fast!

Joint Parsing and Translation

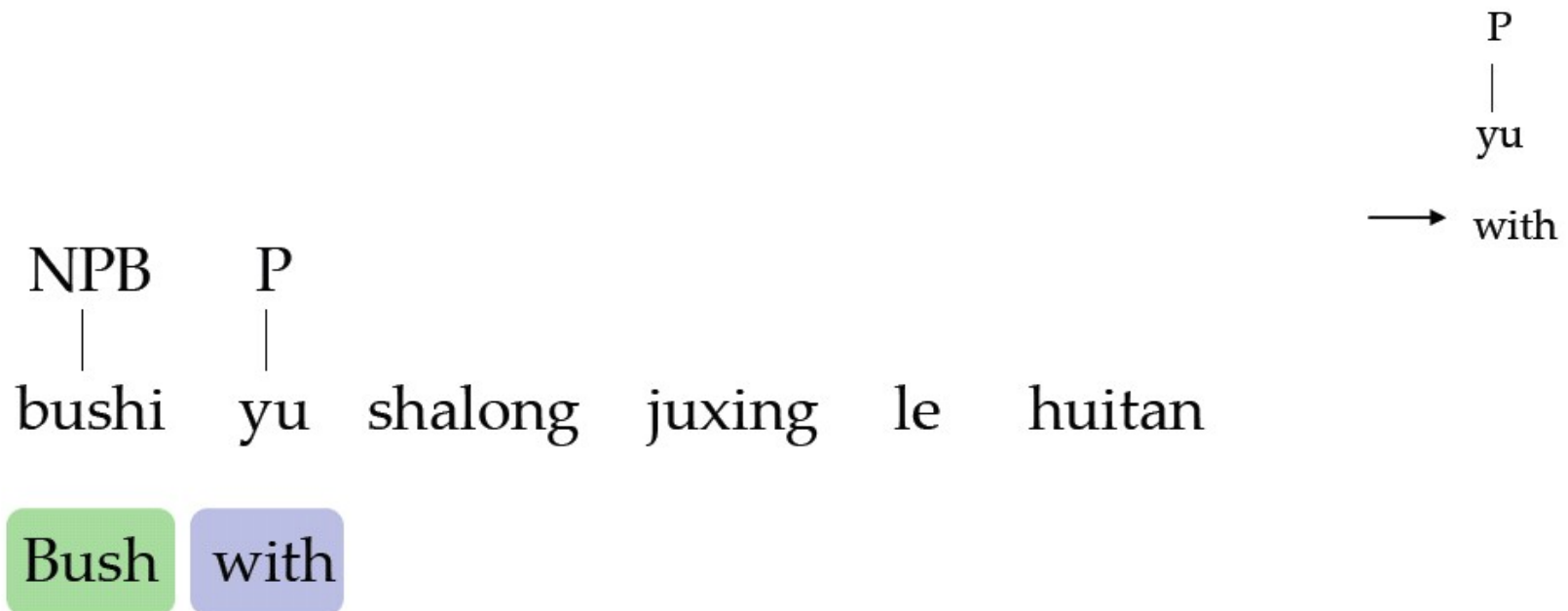


- Its search space is larger than tree/forest
- It is a translator as well as a parser
- Parsing interacts with translation

Joint Parsing and Translation



Joint Parsing and Translation



Joint Parsing and Translation

NPB
|
shalong
→ Sharon

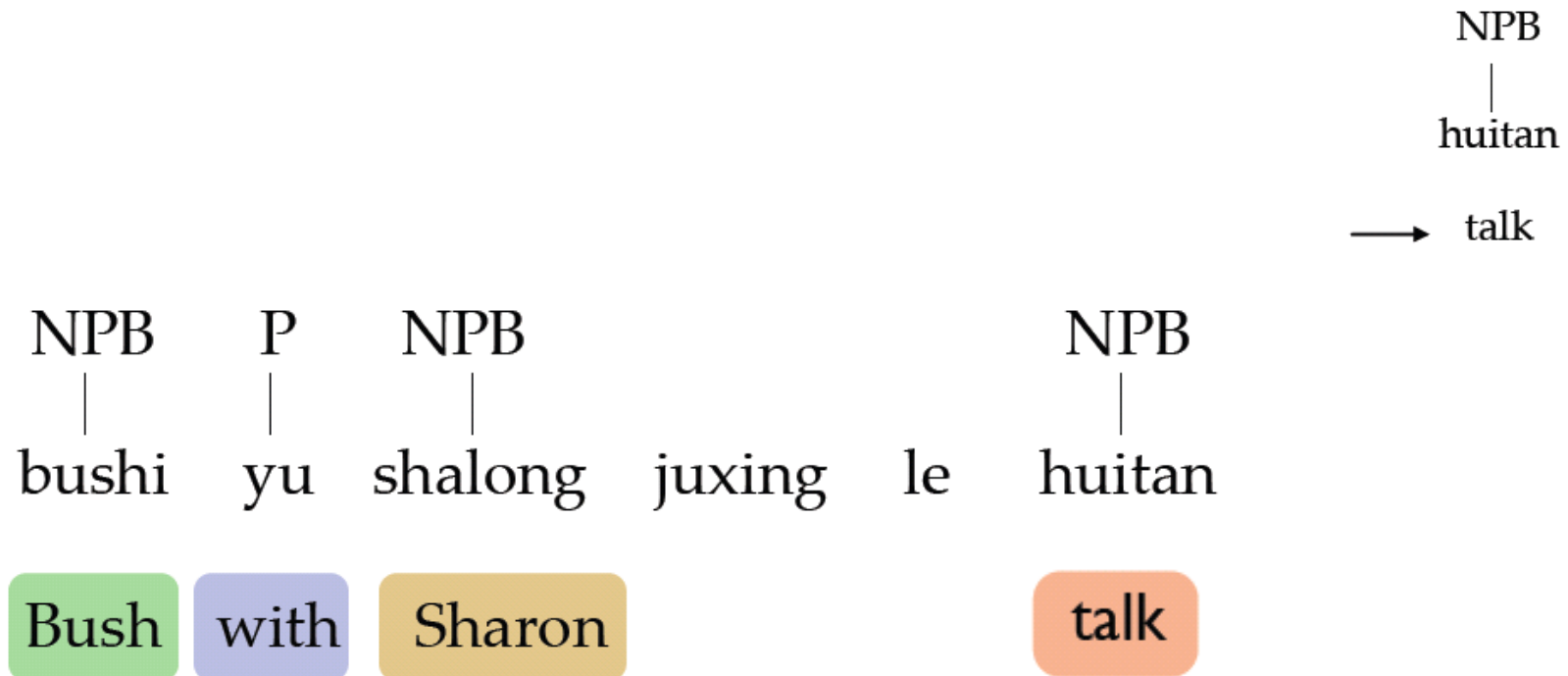
NPB P NPB
| | |
bushi yu shalong juxing le huitan

Bush

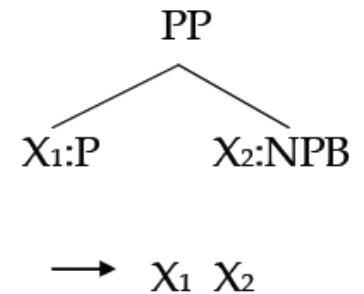
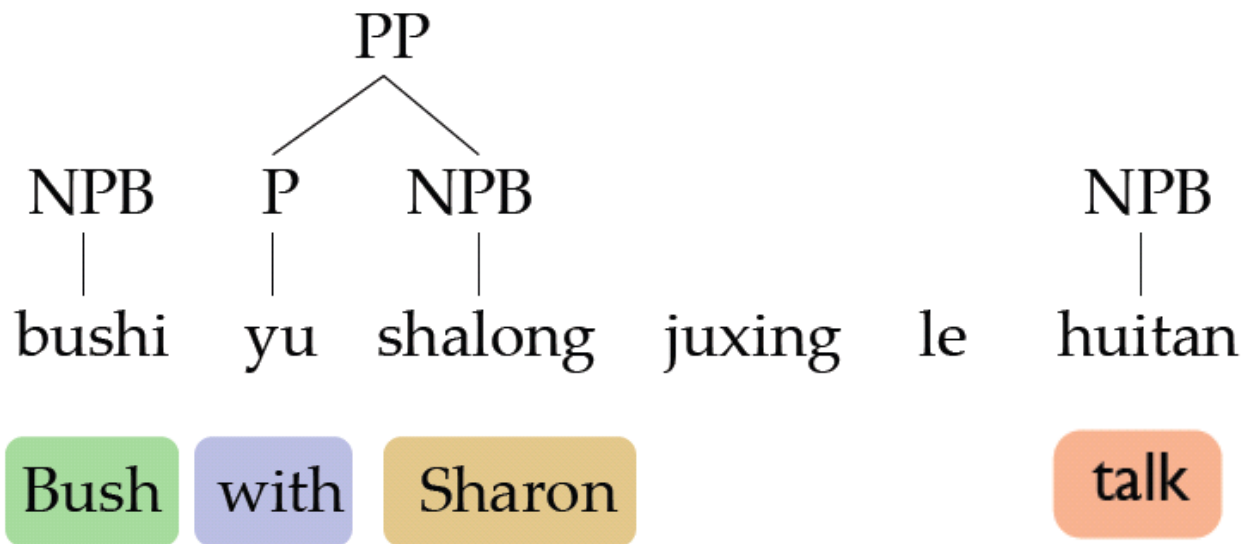
with

Sharon

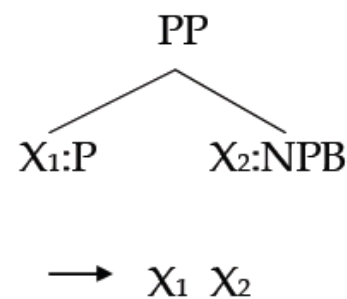
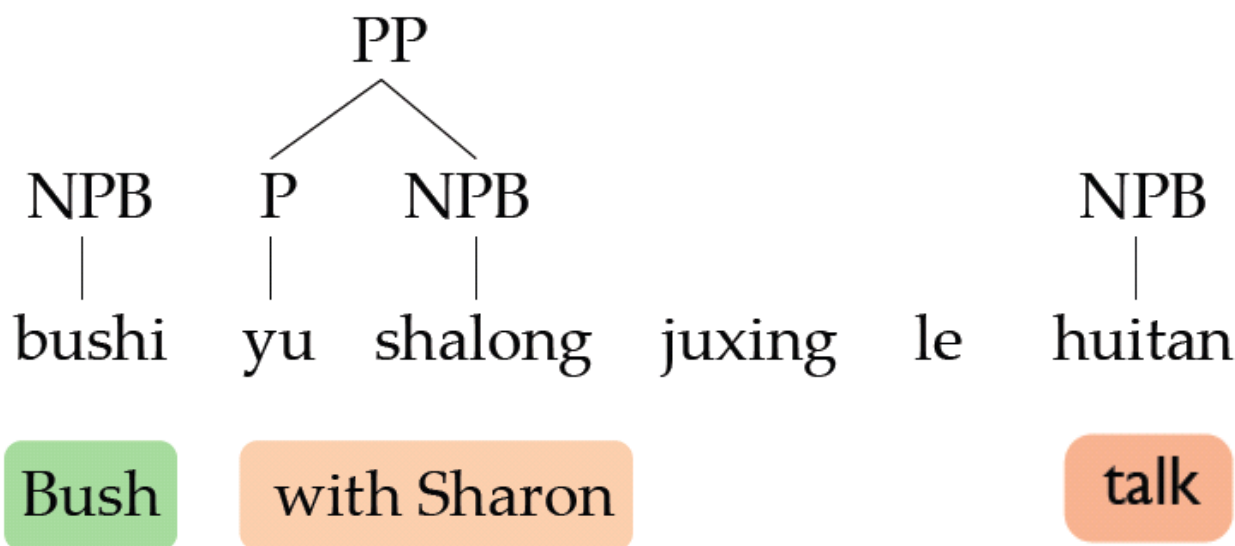
Joint Parsing and Translation



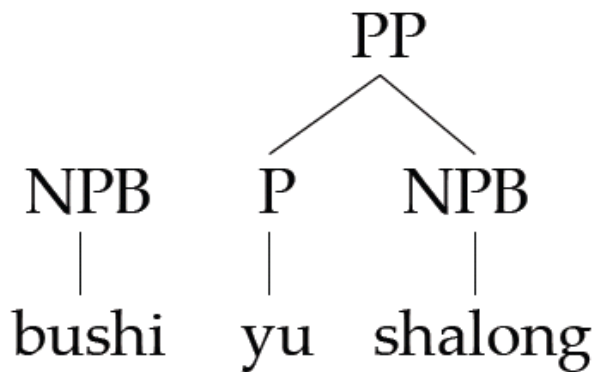
Joint Parsing and Translation



Joint Parsing and Translation

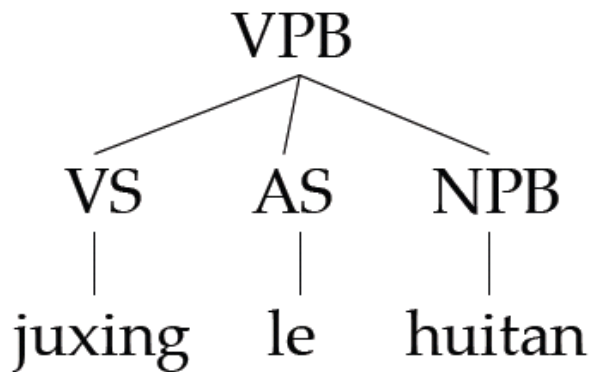


Joint Parsing and Translation

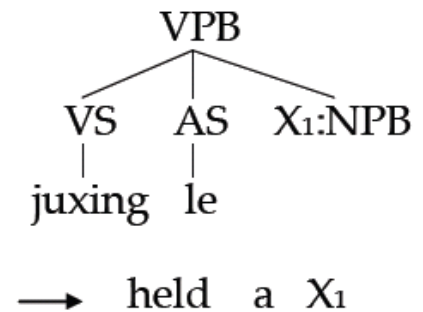


Bush

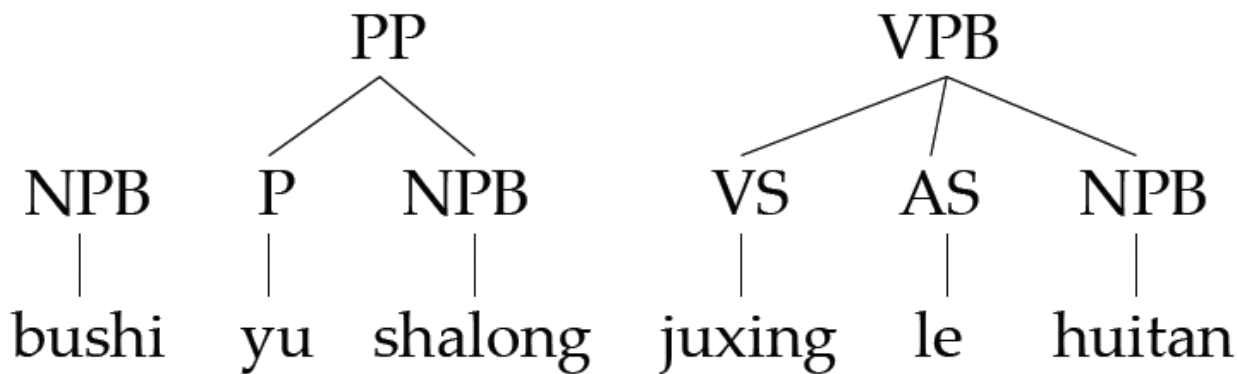
with Sharon



talk



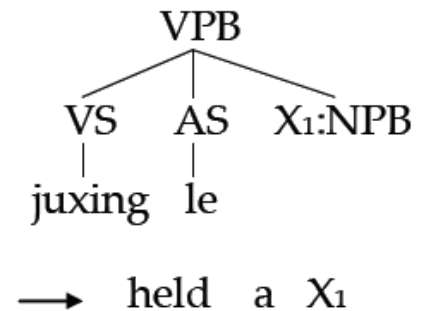
Joint Parsing and Translation



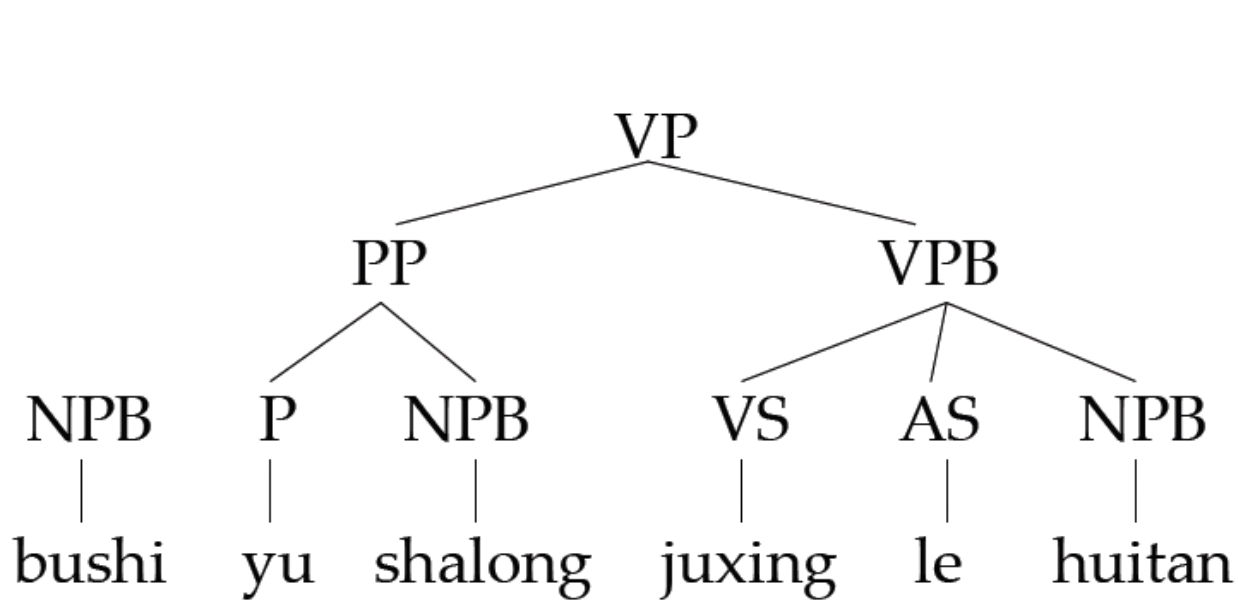
Bush

with Sharon

held a talk



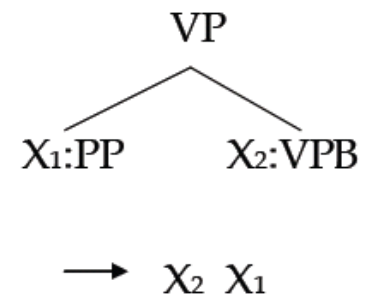
Joint Parsing and Translation



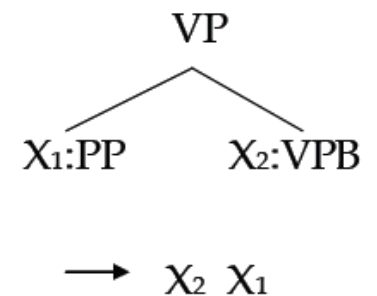
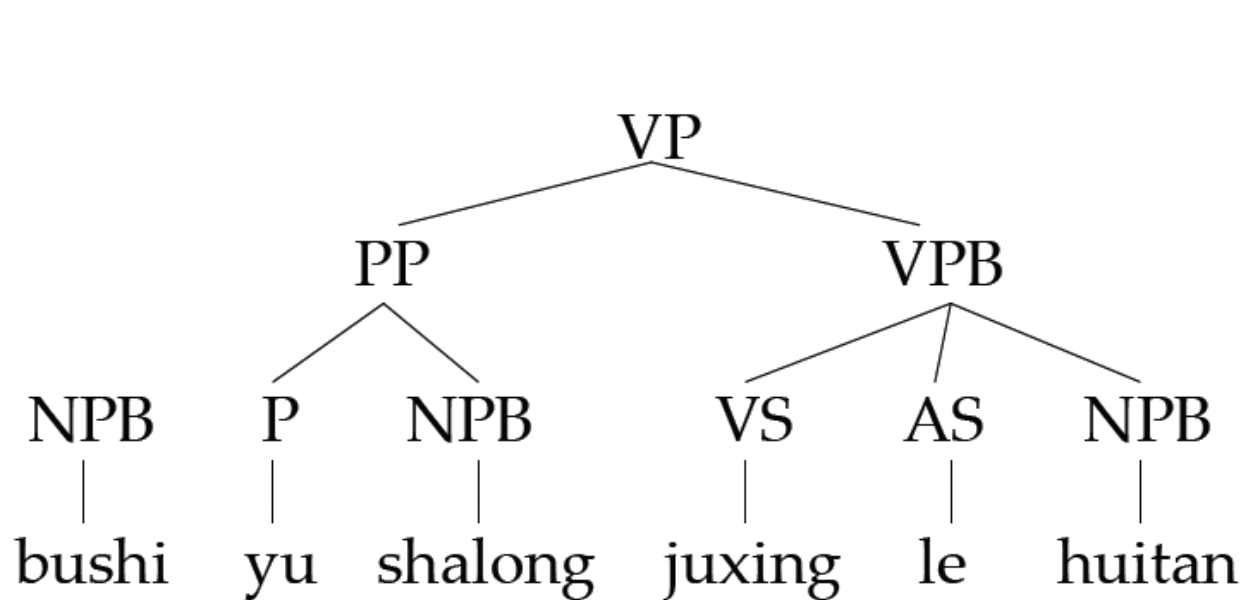
Bush

with Sharon

held a talk



Joint Parsing and Translation



Bush

held

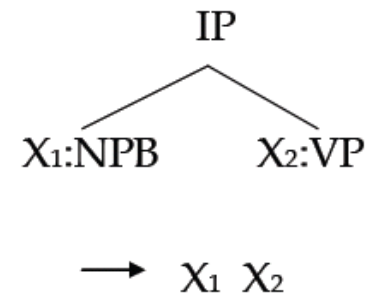
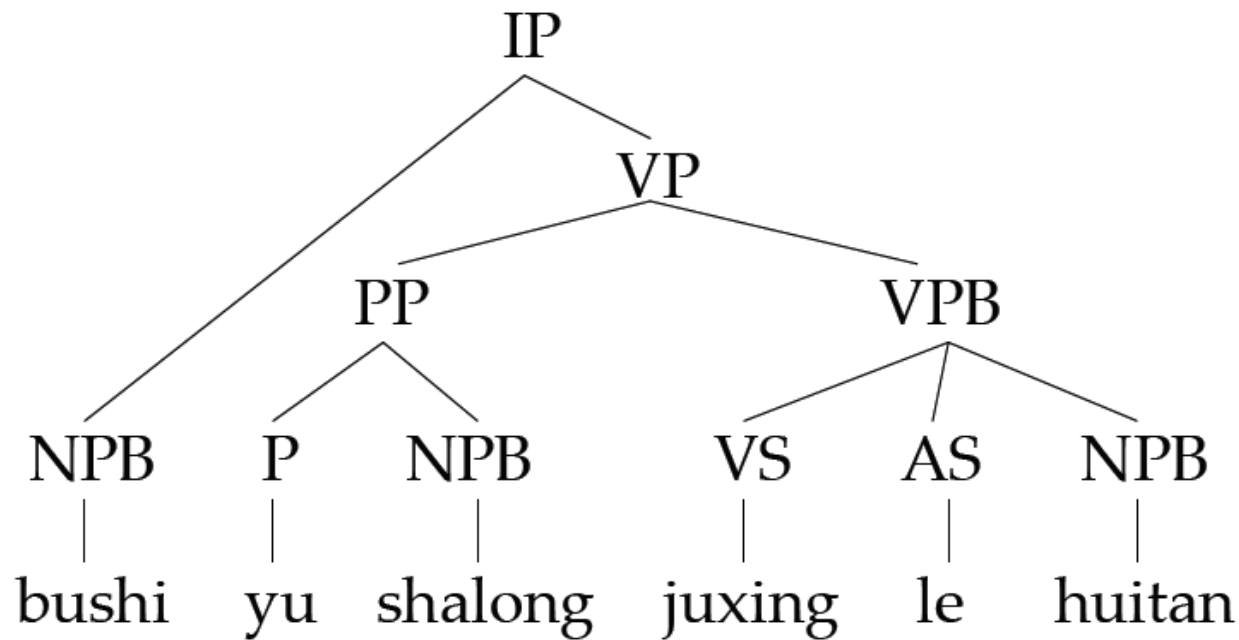
a

talk

with

Sharon

Joint Parsing and Translation



Bush

held

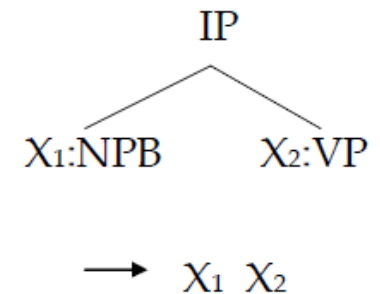
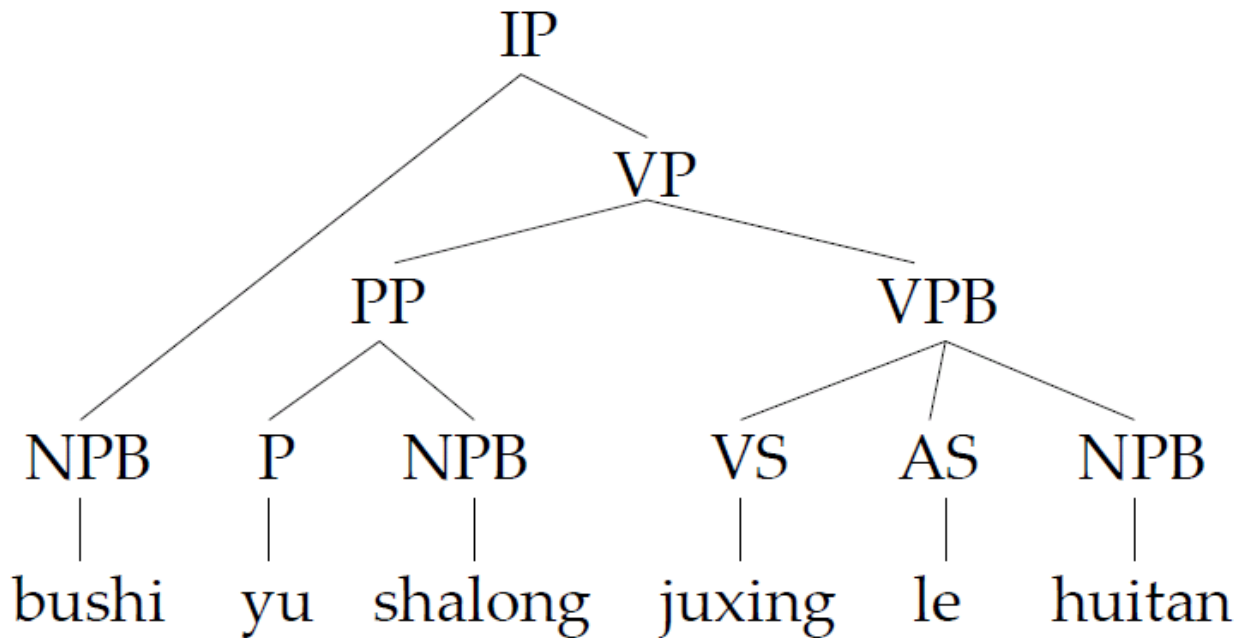
a

talk

with

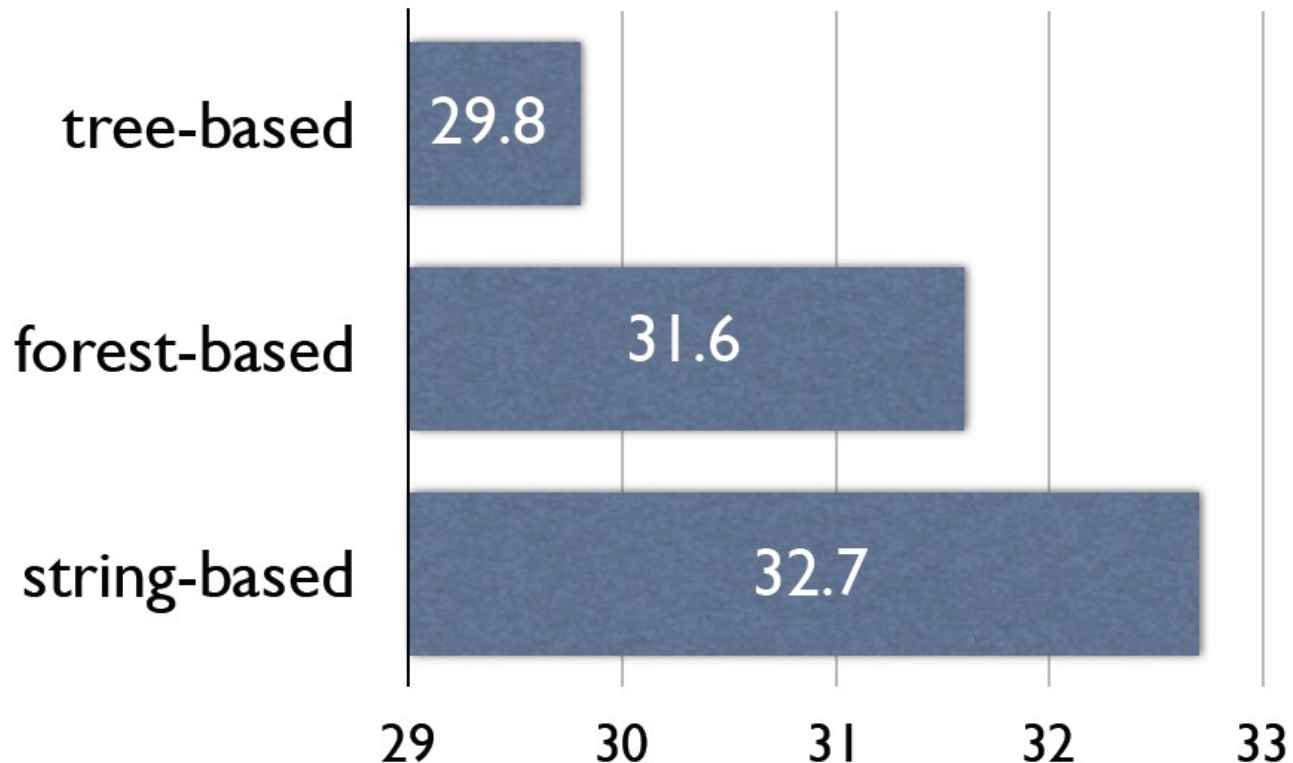
Sharon

Joint Parsing and Translation



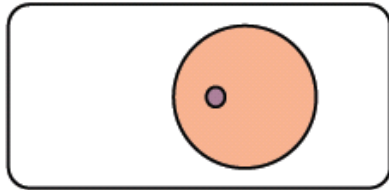
Bush held a talk with Sharon

Evaluation

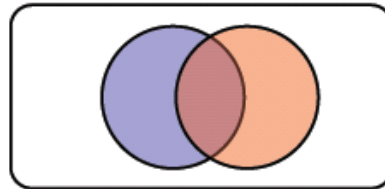


String-based Translation = Joint Parsing and Translation

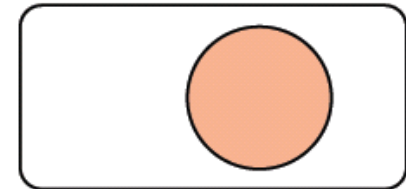
Search Space Comparison



tree-based



forest-based



string-based

String-based Translation = Joint Parsing and Translation

小结

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

串到树翻译模型

- 串到树翻译模型指这样一类翻译模型：
 - 在源语言端进行不句法分析
 - 在目标语言端进行句法分析
 - 从目标语言端句法分析和词语对齐的语料库中抽取翻译规则并构造翻译模型
- 目前，串到树翻译模型的典型工作是美国南加州大学信息科学研究所（**USC/ISI**）从2001年到2005年的系列工作

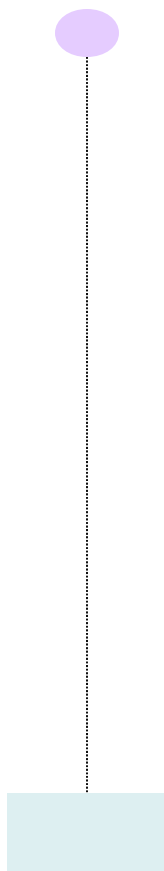
ISI 的工作

- Ulrich Germann, ACL2001 (Best Paper Award)
- Kenji Yamada, ACL2001, ACL2002
- Yaser Al-Onaizan, ACL2002
- Michel Galley, NAACL-HLT 2004
- Jonathan Graehl, NAACL-HLT 2004
- Kevin Knight, CICLing 2005
- Michel Galley, COLING/ACL 2006
- Daniel Marcu, COLING/ACL 2006
- Hao Zhang, NAACL-HLT 2006
- Liang Huang, AMTA 2006

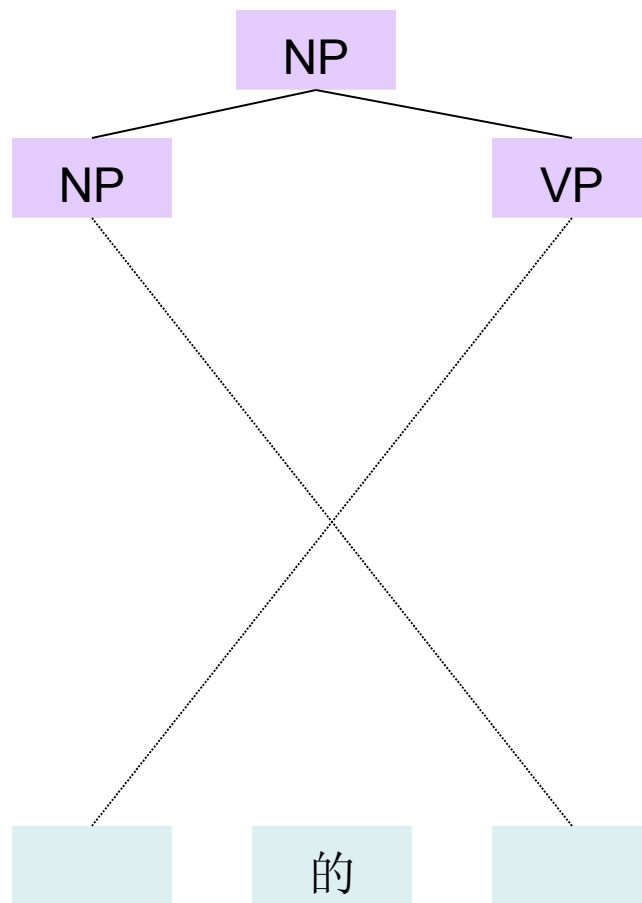
ISI 的工作

- Scalable Inference and Training of Context-Rich Syntactic Models
 - Michel Galley
 - COLING/ACL 2006
- SPMT: Statistical Machine Translation with Syntactified Target Language Phrases
 - Daniel Marcu
 - EMNLP 2006
- Synchronous Binarization for Machine Translation
 - Hao Zhang
 - NAACL-HLT 2006

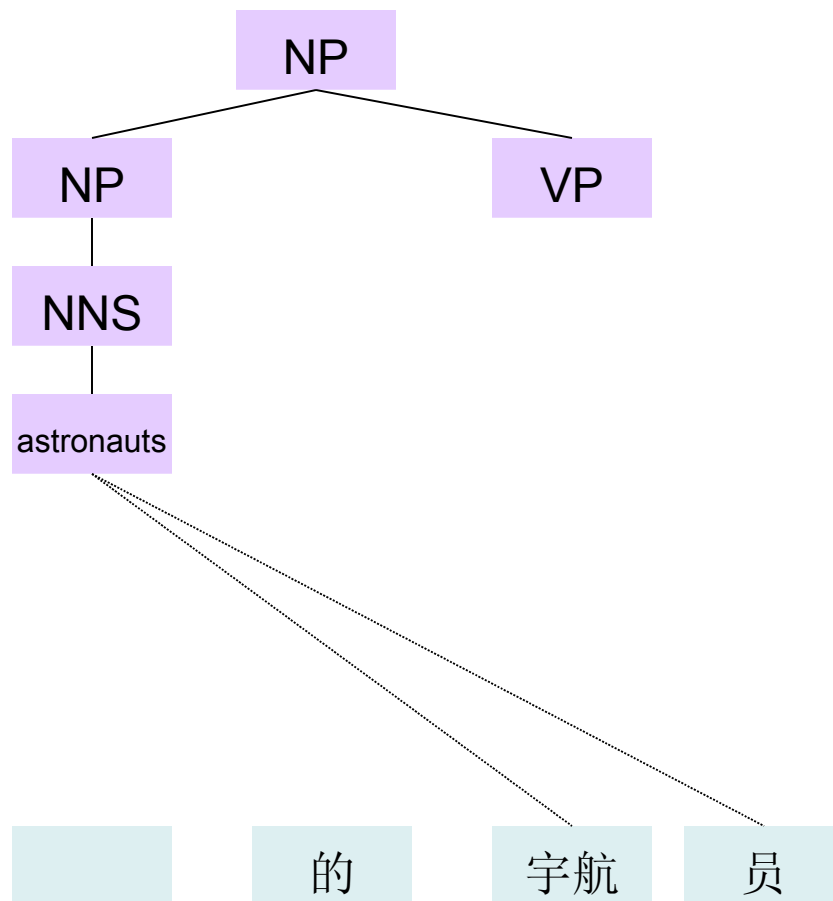
同步生成树、串和对齐



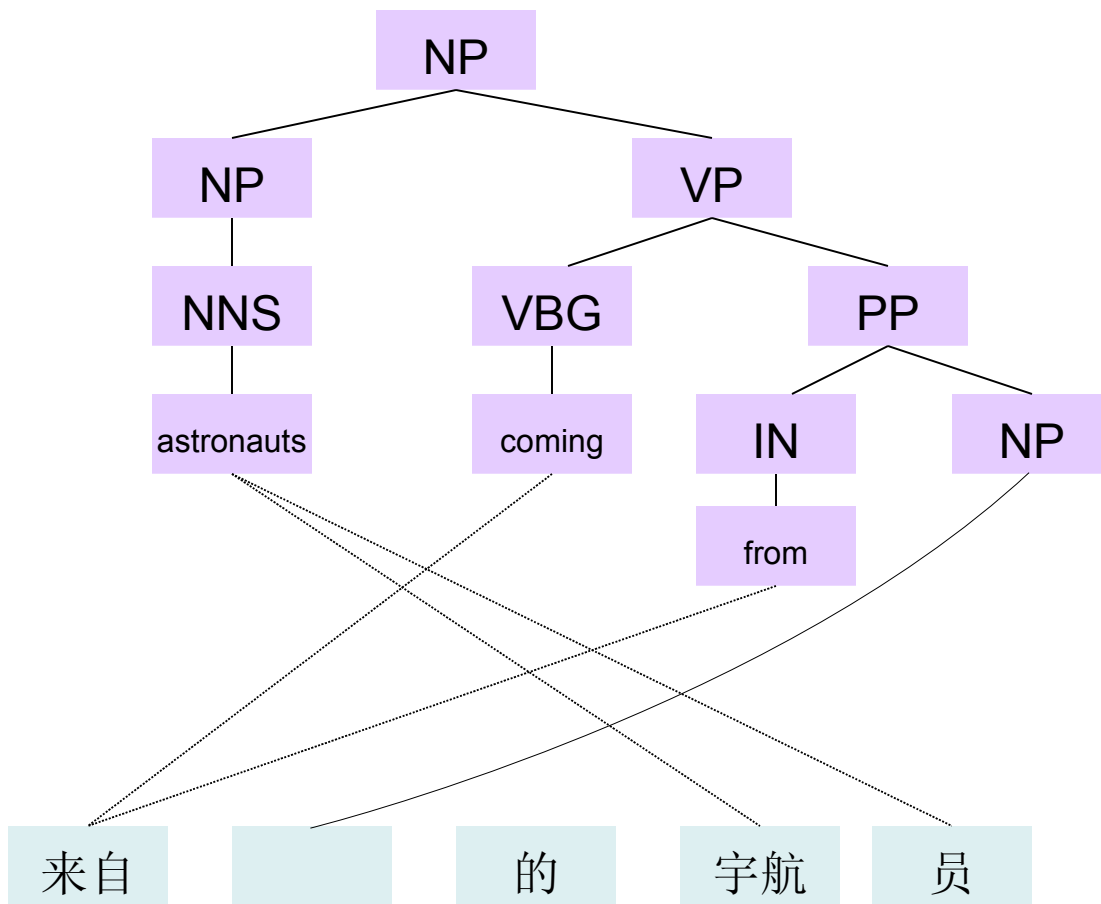
同步生成树、串和对齐



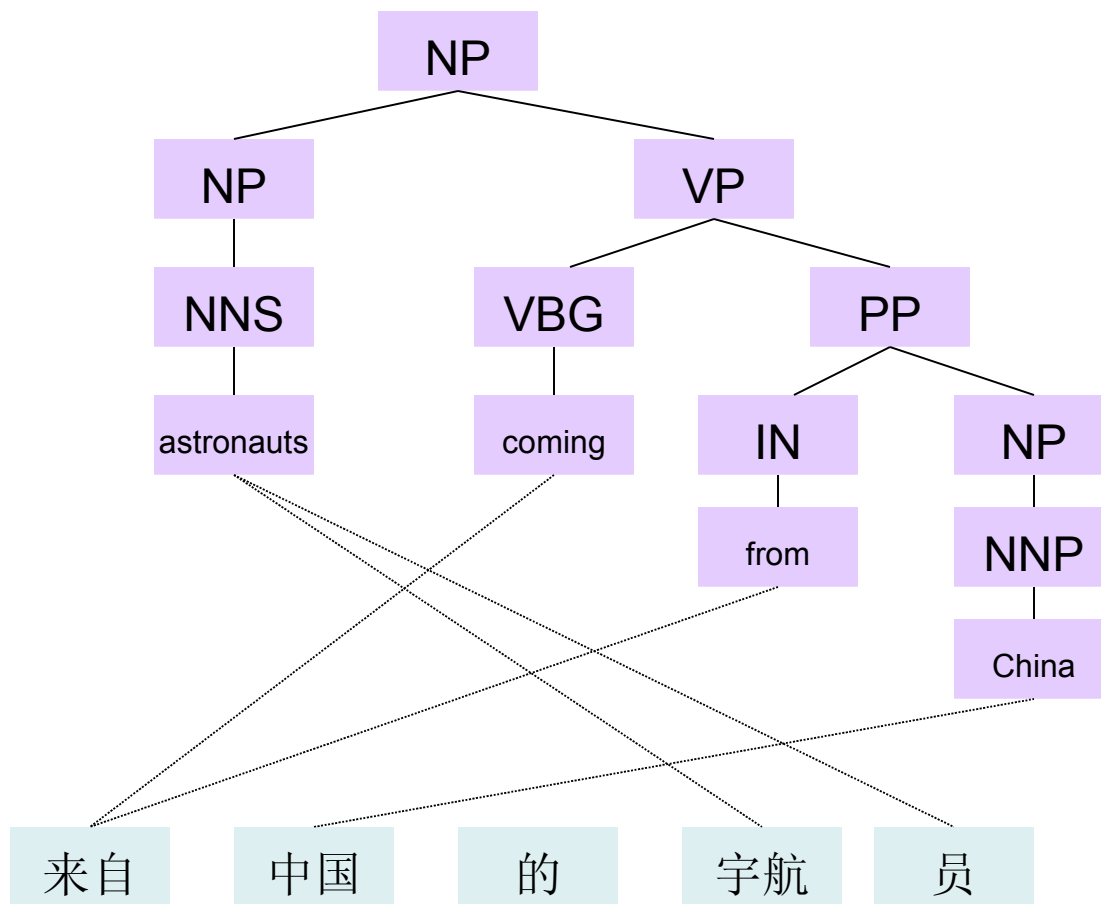
同步生成树、串和对齐



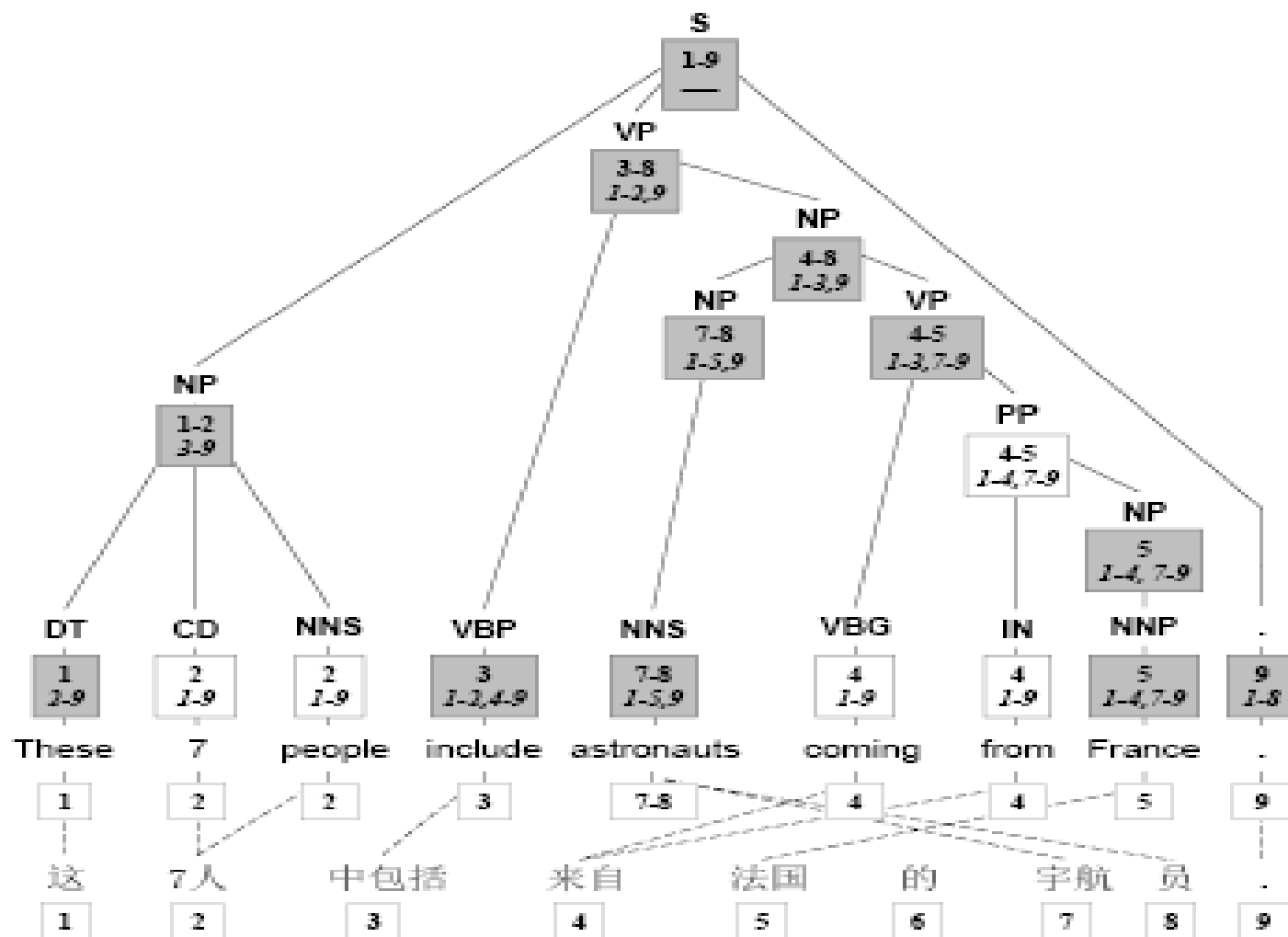
同步生成树、串和对齐



同步生成树、串和对齐

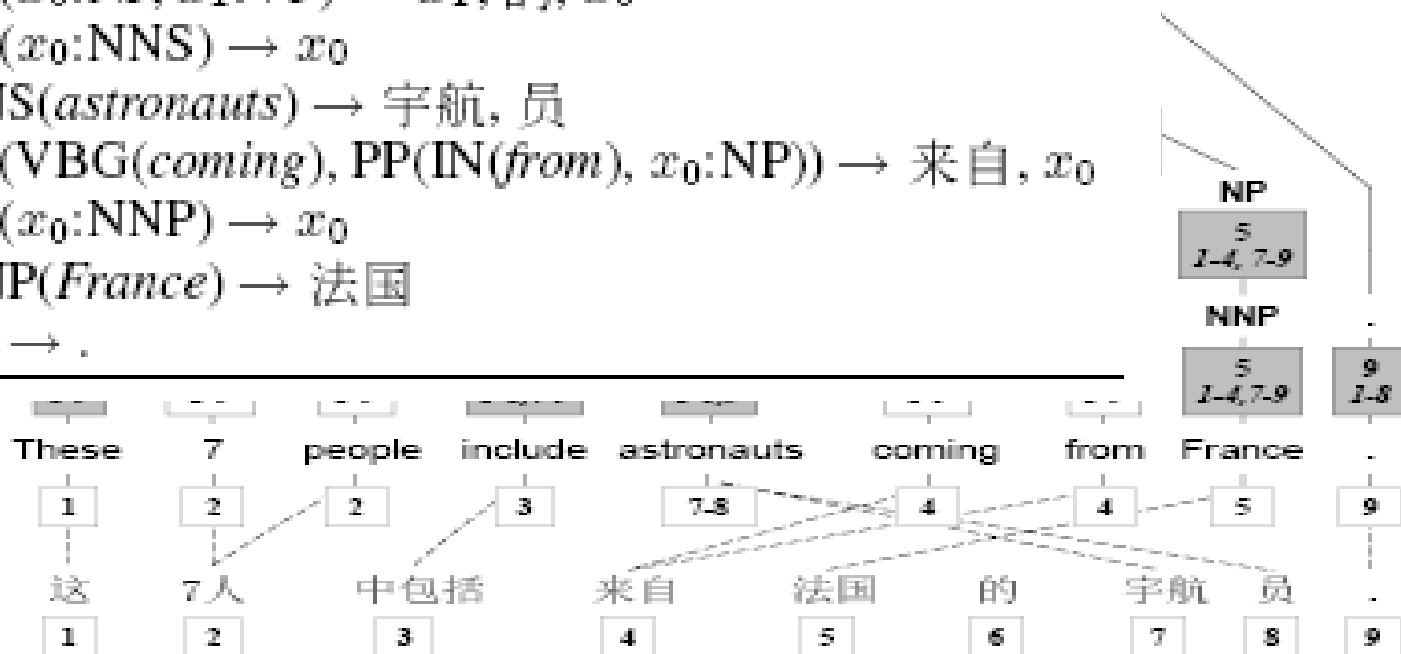


规则与推导



规则与推导

- (a) $S(x_0:NP, x_1:VP, x_2:.) \rightarrow x_0, x_1, x_2$
- (b) $NP(x_0:DT, CD(7), NNS(people)) \rightarrow x_0, 7 \text{ 人}$
- (c) $DT(these) \rightarrow \text{这}$
- (d) $VP(x_0:VBP, x_1:NP) \rightarrow x_0, x_1$
- (e) $VBP(include) \rightarrow \text{中包括}$
- (f) $NP(x_0:NP, x_1:VP) \rightarrow x_1, \text{的}, x_0$
- (g) $NP(x_0:NNS) \rightarrow x_0$
- (h) $NNS(astronauts) \rightarrow \text{宇航, 员}$
- (i) $VP(VBG(coming), PP(IN(from), x_0:NP)) \rightarrow \text{来自}, x_0$
- (j) $NP(x_0:NNP) \rightarrow x_0$
- (k) $NNP(France) \rightarrow \text{法国}$
- (l) $.(.) \rightarrow .$



最小规则与组合规则

- 最小规则
 - 定义:
 - 不能再分解成更小规则的规则
 - 例子:
 - $\text{NP}(\text{x0:DT}, \text{CD}(7), \text{NNS}(\text{people})) \rightarrow \text{x0}, 7 \text{ 人}$
 - $\text{DT}(\text{these}) \rightarrow \text{这}$
- 组合规则
 - 定义:
 - 由两个或者多个最小规则组合成的规则
 - 例子:
 - $\text{NP}(\text{DT}(\text{these}), \text{CD}(7), \text{NNS}(\text{people})) \rightarrow \text{这}, 7 \text{ 人}$
 - $\text{NP}(\text{x0:DT}, \text{CD}(7), \text{NNS}(\text{people})) \rightarrow \text{x0}, 7, \text{ 人}$

句法翻译概率表

lhs_1 : NP-C(x_0 :NPB PP(IN(of) x_1 :NP-C))				(NP-of-NP)			
lhs_2 : PP(IN(of) NP-C(x_0 :NPB PP(IN(of) NP-C(x_1 :NPB x_2 :VP))))				(of-NP-of-NP-VP)			
lhs_3 : VP(VBD($said$) SBAR-C(IN($that$) x_0 :S-C))				(said-that-S)			
lhs_4 : SBAR(WHADVP(WRB($when$)) S-C(x_0 :NP-C VP(VBP(are) x_1 :VP-C)))				(when-NP-are-VP)			
rhs_{1i}	$p(rhs_{1i} lhs_1)$	rhs_{2i}	$p(rhs_{2i} lhs_2)$	rhs_{3i}	$p(rhs_{3i} lhs_3)$	rhs_{4i}	$p(rhs_{4i} lhs_4)$
$x_1 x_0$.54	x_2 的 x_1 的 x_0	.6754	说, x_0	.6062	在 $x_1 x_0$ 时	.6618
$x_0 x_1$.2351	在 x_2 的 x_1 的 x_0	.035	说 x_0	.1073	当 $x_1 x_0$ 时	.0724
x_1 的 x_0	.0334	x_2 的 x_1 的 x_0 ,	.0263	表示, x_0	.0591	在 $x_1 x_0$ 时,	.0579
$x_1 x_0$ 的	.026	x_2 的 x_1 的 x_0 有	.0116	他说, x_0	.0234	, 在 $x_1 x_0$ 时	.0289

Table 4: Translation probabilities promote linguistically motivated constituent re-orderings (for lhs_1 and lhs_2), and enable non-constituent (lhs_3) and non-contiguous (lhs_4) phrasal translations.

翻译模型定义

- Galley

- 条件概率模型：将给定的目标语言树转换成源语言词串的概率（ F 是源语言句子， π 是目标语言句法树， Θ 是从 π 生成 F 的所有推导的集合， θ 是一个具体的推导）

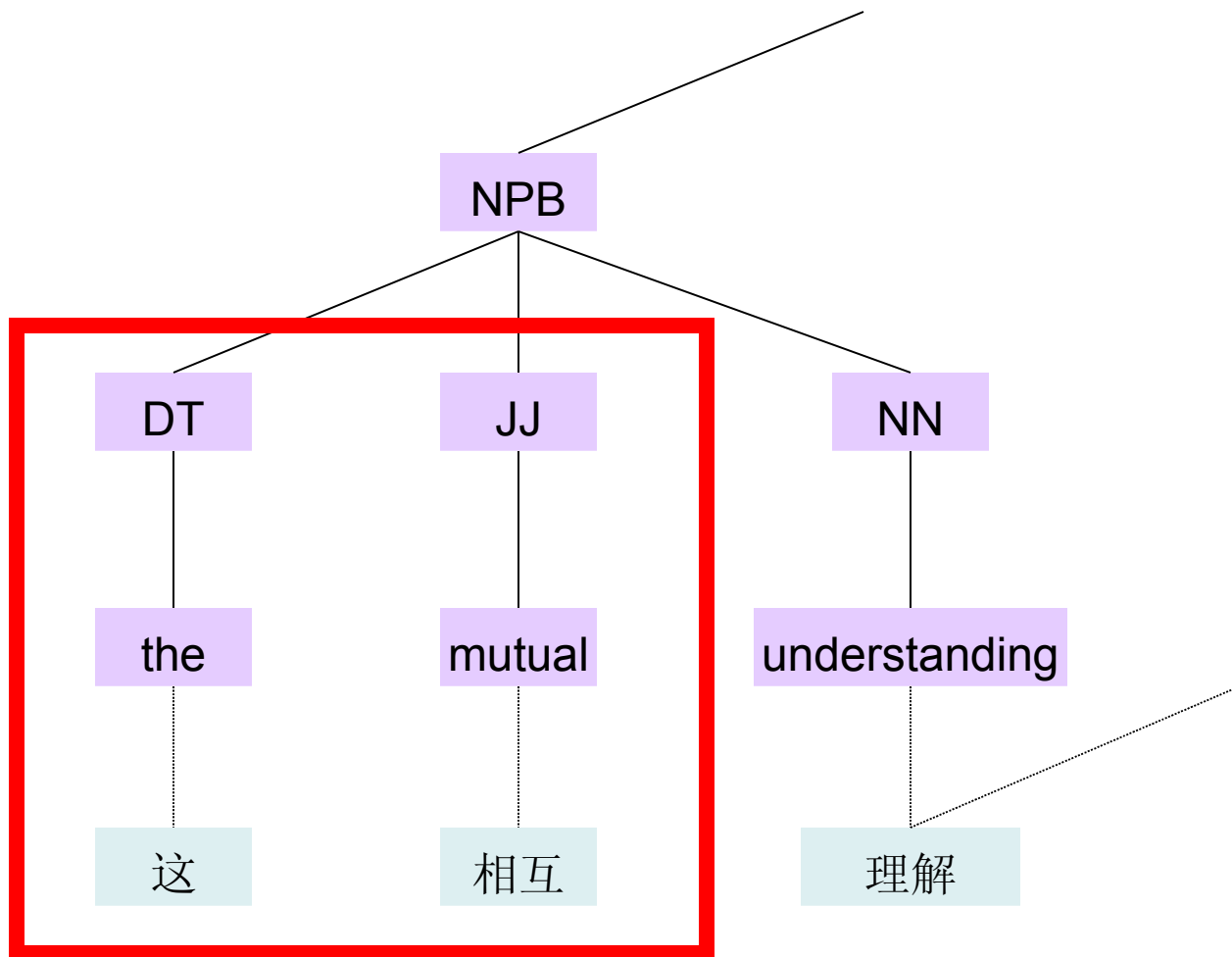
$$Pr(F|\pi) = \frac{1}{|\Lambda|} \sum_{\theta_i \in \Theta} \prod_{r_j \in \theta_i} p(rhs(r_j)|lhs(r_j))$$

- Marcu

- 联合概率模型：同步生成源语言词串、目标语言树和对齐的概率（ F 是源语言句子， π 是目标语言句法树， A 是 F 和 π 的对齐， Θ 是同时生成 π, F, A 的所有推导的集合， θ 是一个具体的推导）

$$Pr(\pi, F, A) = \sum_{\theta_i \in \Theta, c(\theta_i) = (\pi, F, A)} \prod_{r_j \in \theta_i} p(r_j)$$

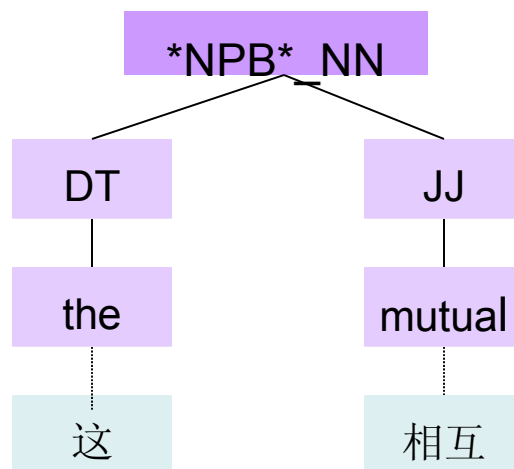
与短语模型的兼容性：非句法短语



非句法短语的处理办法

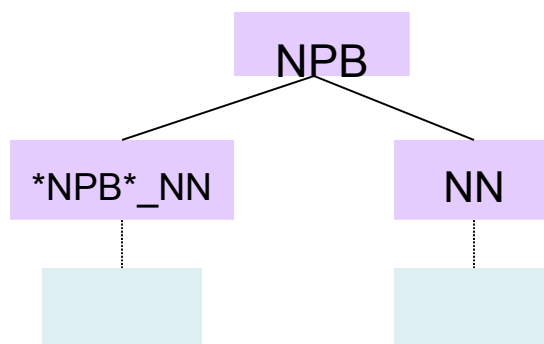
- Marcu 的办法
 - 忽略非句法短语
 - 损失：在汉语-英语双语语料库中提取的短语中，28%都是非句法短语
 - 沿目标语言句法树向上找一个可以覆盖该短语并满足对齐约束的结点
 - 问题：可能需要引入很大范围的上下文
 - 为非句法短语构造新的规则：
Compatible Rules 兼容规则

构造兼容规则 (1)



- 构造一条规则：
 - 根结点是一个“伪”非终结符结点
 - 覆盖若干棵目标语言句法子树及其对应的源语言词串

构造兼容规则 (2)



- 构造另一条对应的规则：描述该“伪”结点如何与周围的“真”句法结点组合成“真”句法树。

构造兼容规则 (3)

- 兼容规则的引入，以比较小的代价实现了与短语模型的兼容性，提高了系统的性能

模型特征

- Galley (1)
 - EM-trained root-normalized SBTM
- Marcu (11)
 - $p_{\text{root}}(r)$ root normalized conditional probability of all rules
 - $p_{\text{cfg}}(r)$ CFG-like probability of non-lexicalized rules
 - $\text{is_lexicalized}(r)$ indicator 0/1
 - $\text{is_composed}(r)$ indicator 0/1
 - $\text{is_lowcount}(r)$ indicator $\text{count} < 3 ? 1 : 0$
 - $\text{lex_pef}(r)$ direct phrase-based conditional probability
 - $\text{lex_pfe}(r)$ inverse phrase-based conditional probability
 - $m1(r)$ IBM model 1 probability
 - $m1\text{inv}(r)$ IBM model 1 inverse probability
 - $\text{lm}(e)$ language model
 - $\text{wp}(e)$ word penalty

训练

- 规则抽取
 - Input: word-aligned, target side parsed bilingual corpus
 - Output: rules
- 概率估计
 - How to estimate the probability distribution of rules?

Galley 的规则抽取方法

- 首先计算边沿结点集合
- 自顶向下，以每一个边沿结点为根结点：
 - 抽取最小规则，得到最小推导或者：
 - 对于该结点覆盖的未对齐源语言结点，考虑其不同的附着方式，抽取所有组合规则，得到推导森林

将树到串对齐表示为图

- 为了对规则抽取的过程进行形式化描述，我们将（树，串，对齐）三元组表示为一个有向图（边都是向下的），其中并不对树中的边和表示对齐关系的边加以区别。

Some Notions

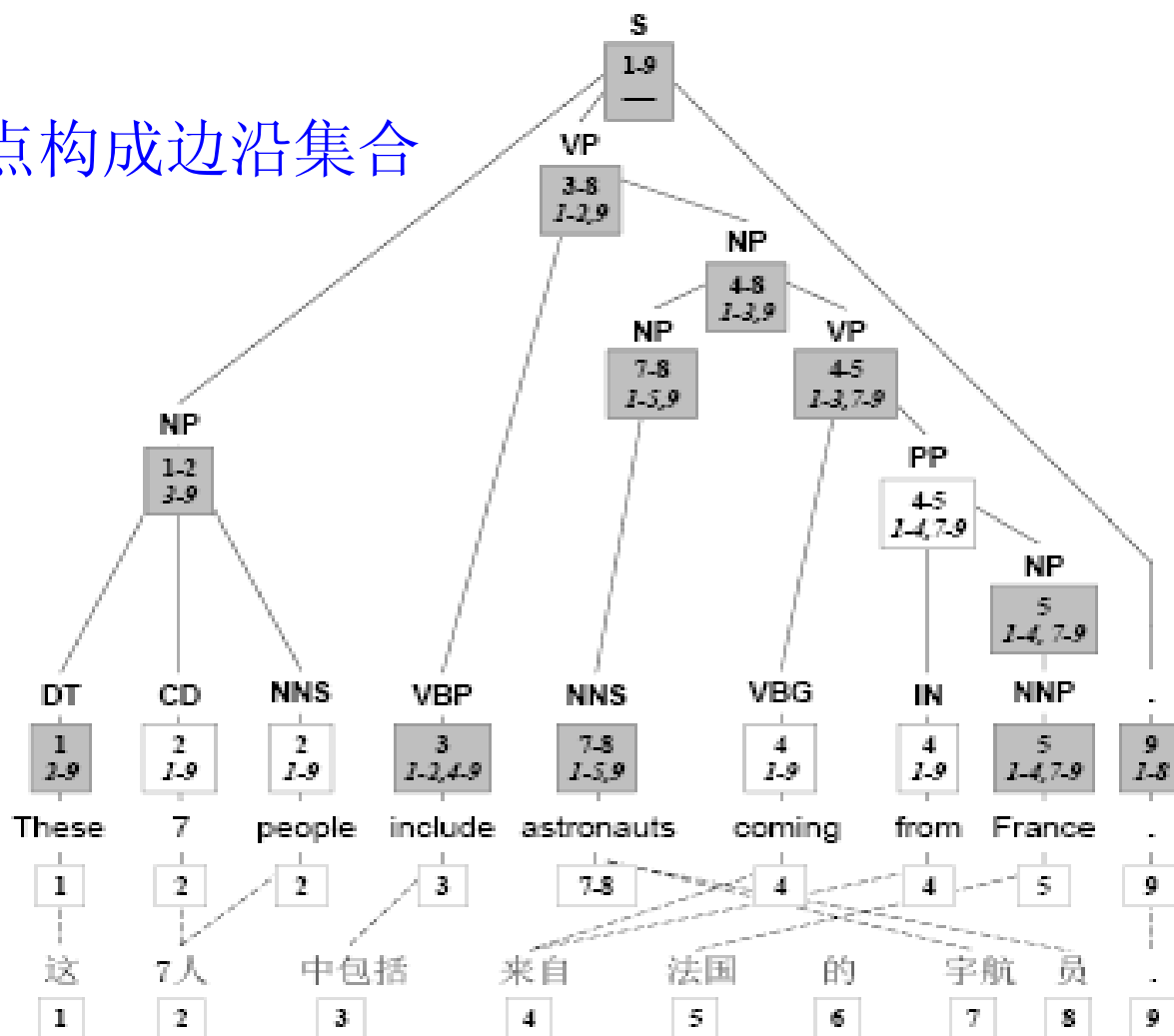
- The **span** of a node n is defined by the indices of the first and last word in the source string that are reachable from n
- The **complement span** of a node n is the union of the spans of all nodes n' in G that are neither descendants nor ancestors of n
- Nodes of G whose spans and complement spans are non-overlapping form the **frontier set** $F \in G$
- A **frontier graph fragment** is a graph fragment that root and all sinks are in the frontier set
- A **minimal frontier graph fragment** is the one that is a subgraph of every other frontier graph fragment with the same root.

一些概念定义

- 一个结点 n 的**区间 (span)** 是该结点所对应的第一个和最后一个源语言单词所指定的范围。
- 一个结点 n 的**补区间 (complement span)** 是图 G 中所有既非 n 的子孙结点也非 n 的祖先结点的那些结点 n' 的区间 (span) 所构成的并集
- 图 G 的**边沿集合 (frontier set)** 是由图 G 中那些其区间与补区间不重叠的结点所构成的集合 F ($F \subseteq G$)
- 图 G 的一个**边沿图片段 (frontier graph fragment)** 是图 G 的一个片段, 其根结点及其 Sink 结点都位于图 G 的边沿集合中
- 图 G 的一个**最小边沿图片段 (minimal frontier graph fragment)** 是图 G 的一个边沿图片段, 而且它是所有其他具有相同根结点的边沿图片段的子图

An Example

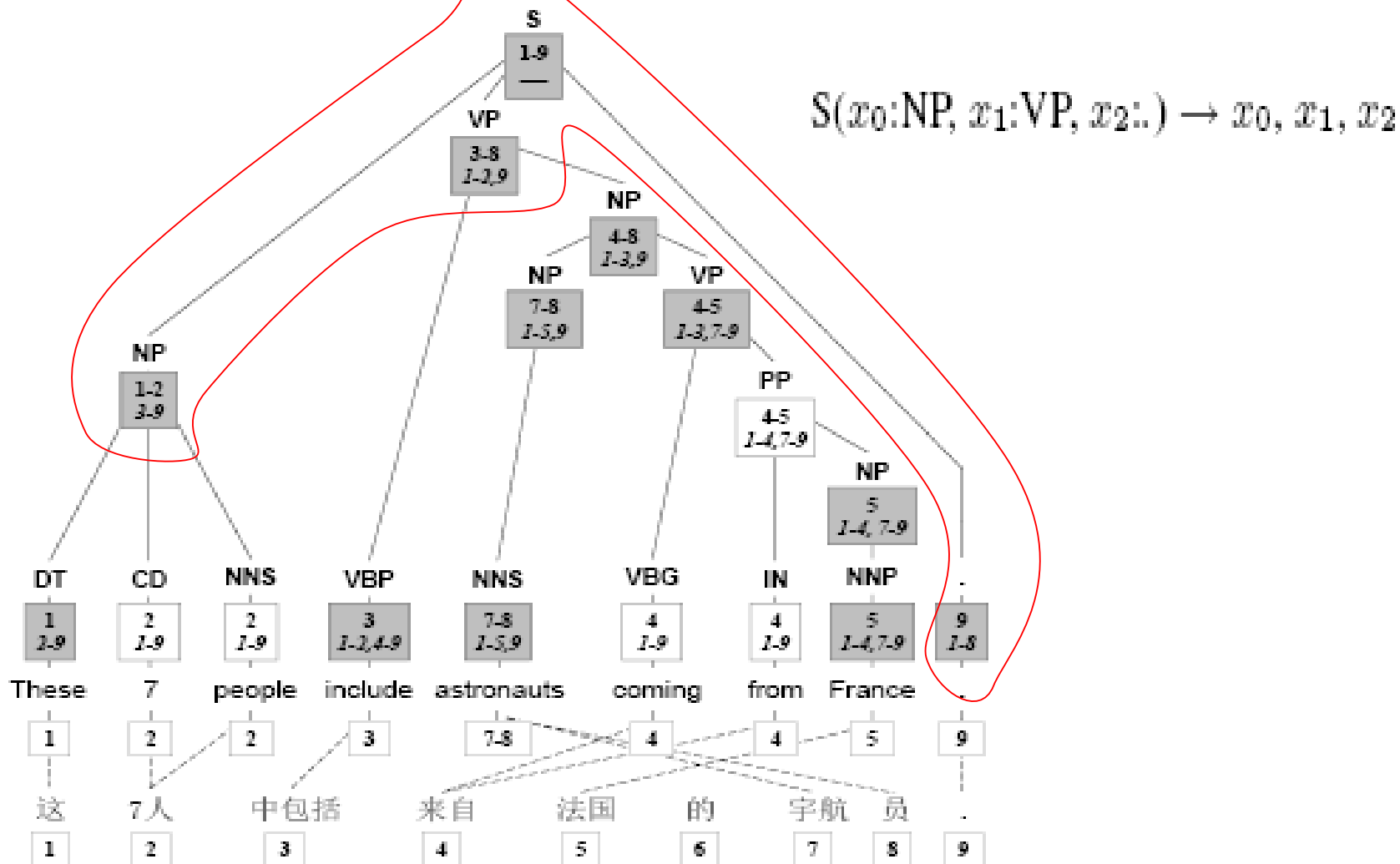
灰色结点构成边沿集合



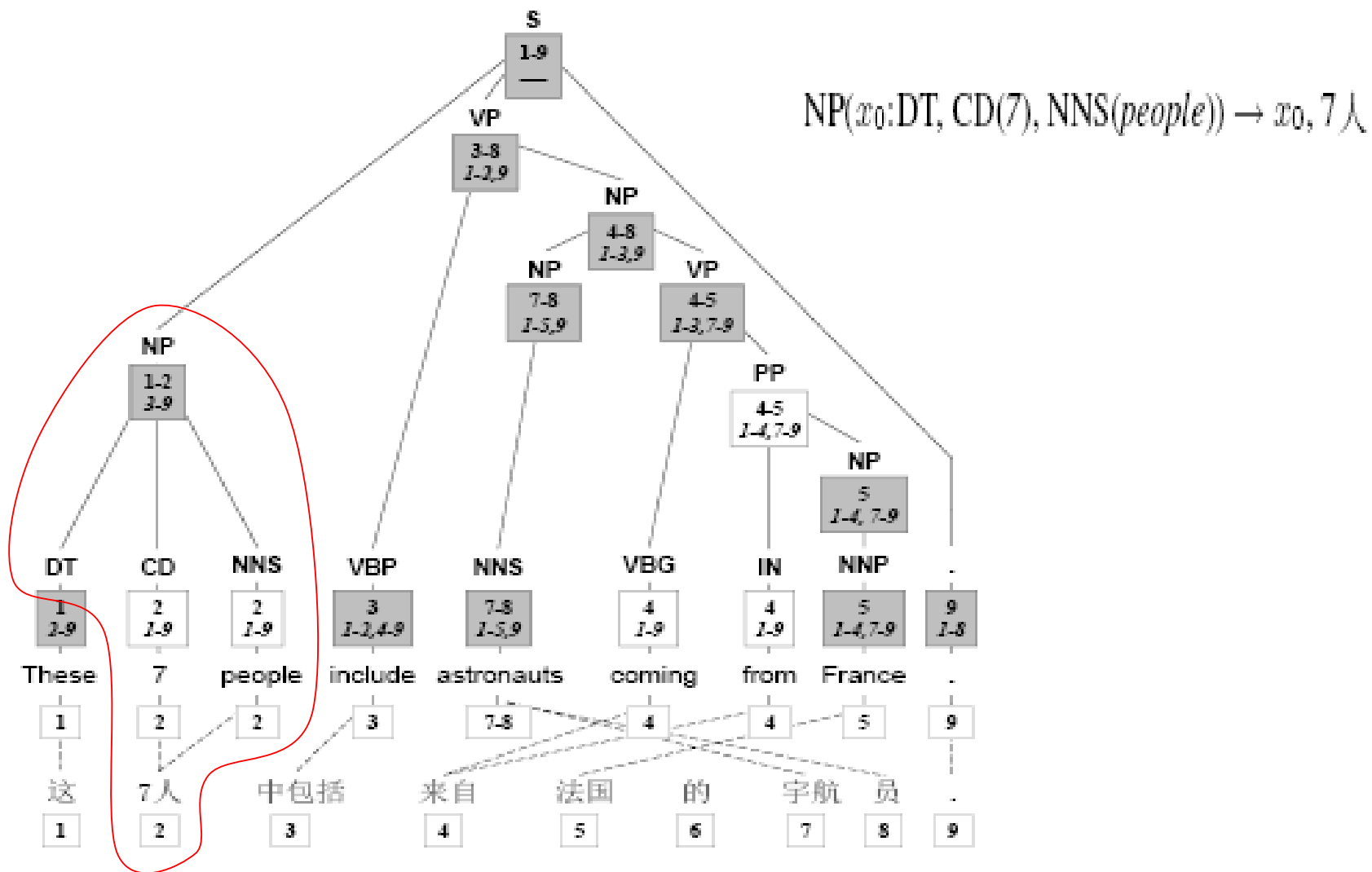
抽取规则算法

- Step 1: 计算图的边沿集合
- Step 2: 对边沿集合中的每个结点，计算以其为根结点的最小边沿图片片段
- Step3: 从该最小边沿图片片段中导出规则

抽取规则 1

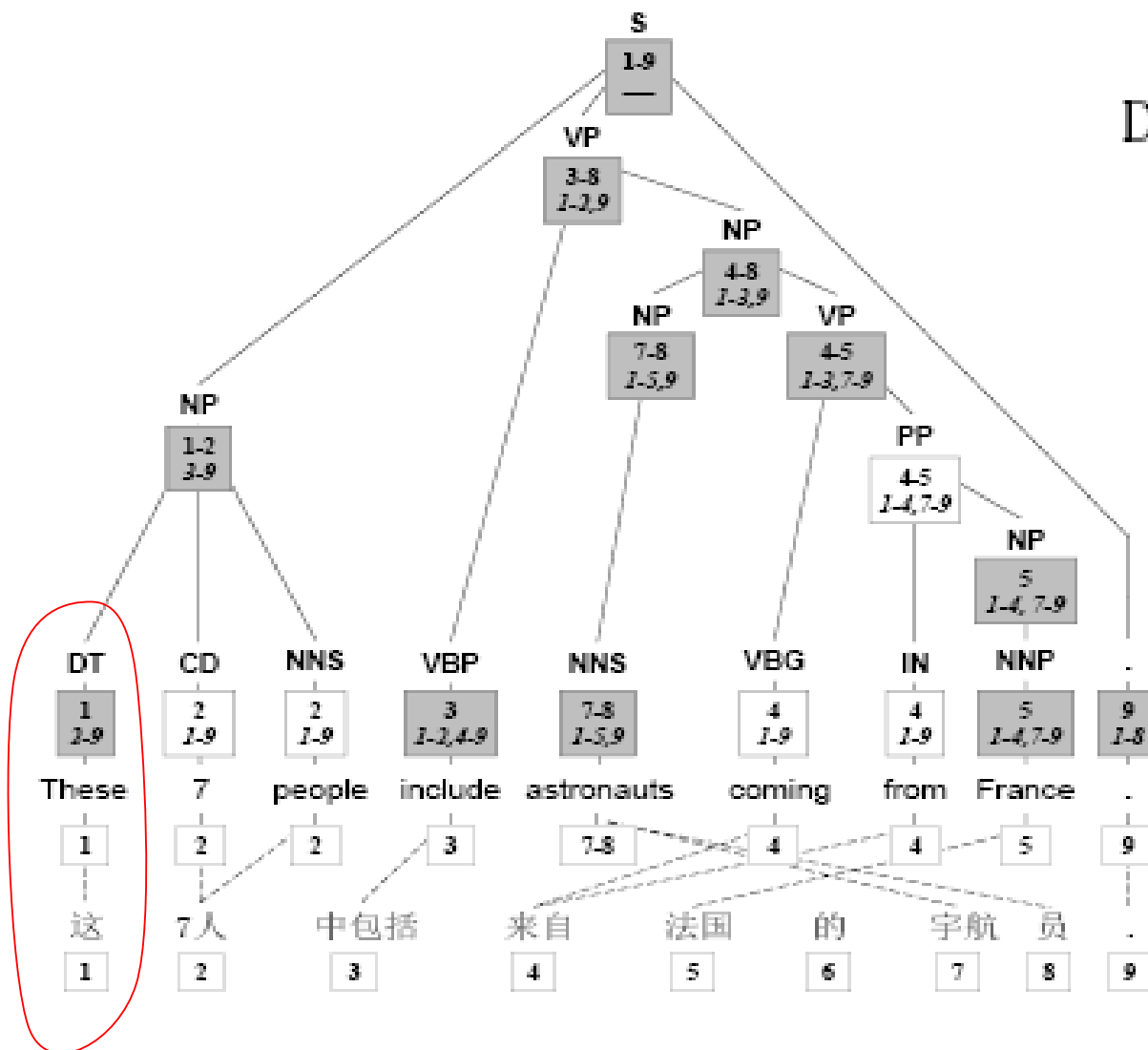


抽取规则 2

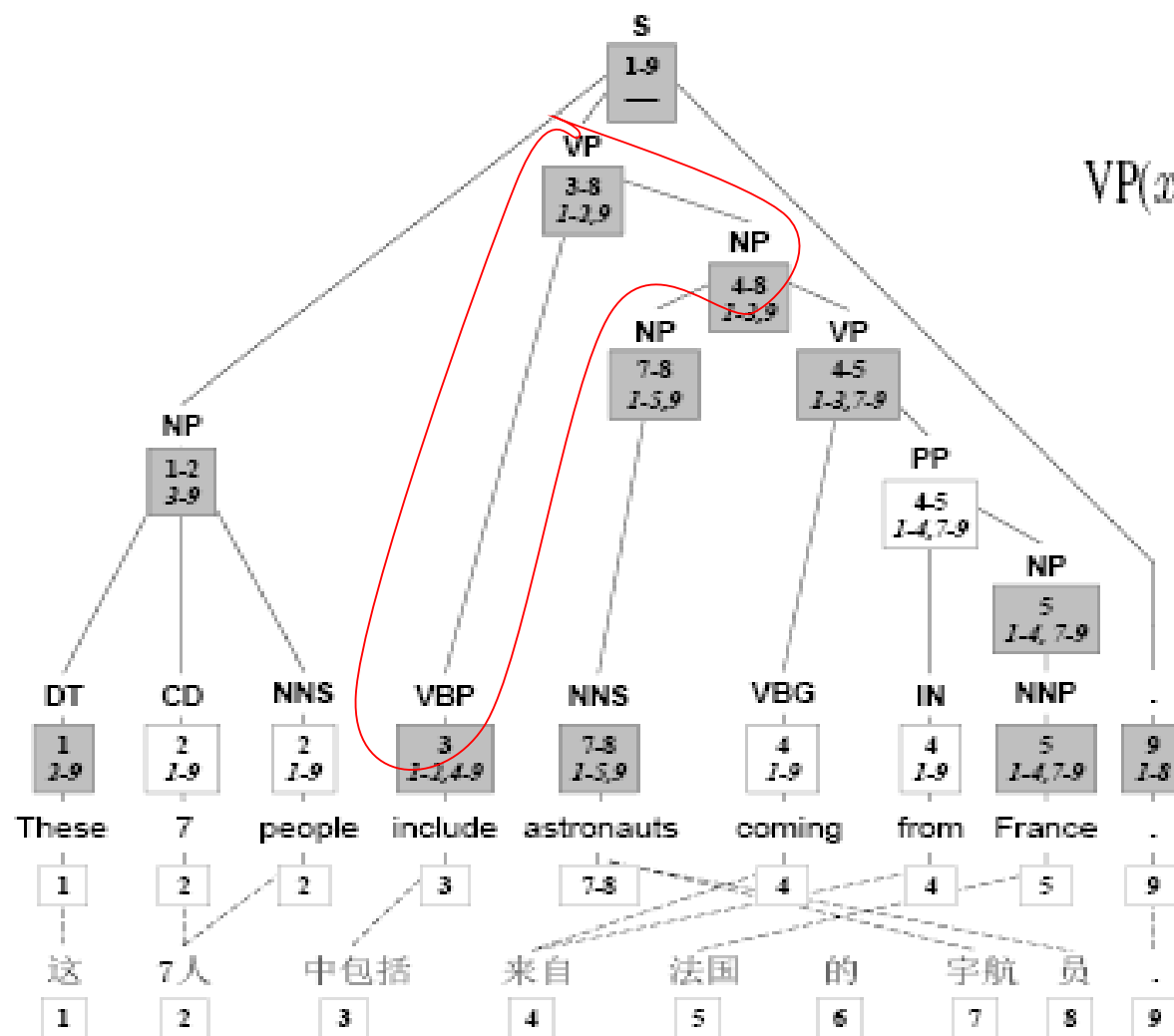


抽取规则 3

DT(*these*) → 这

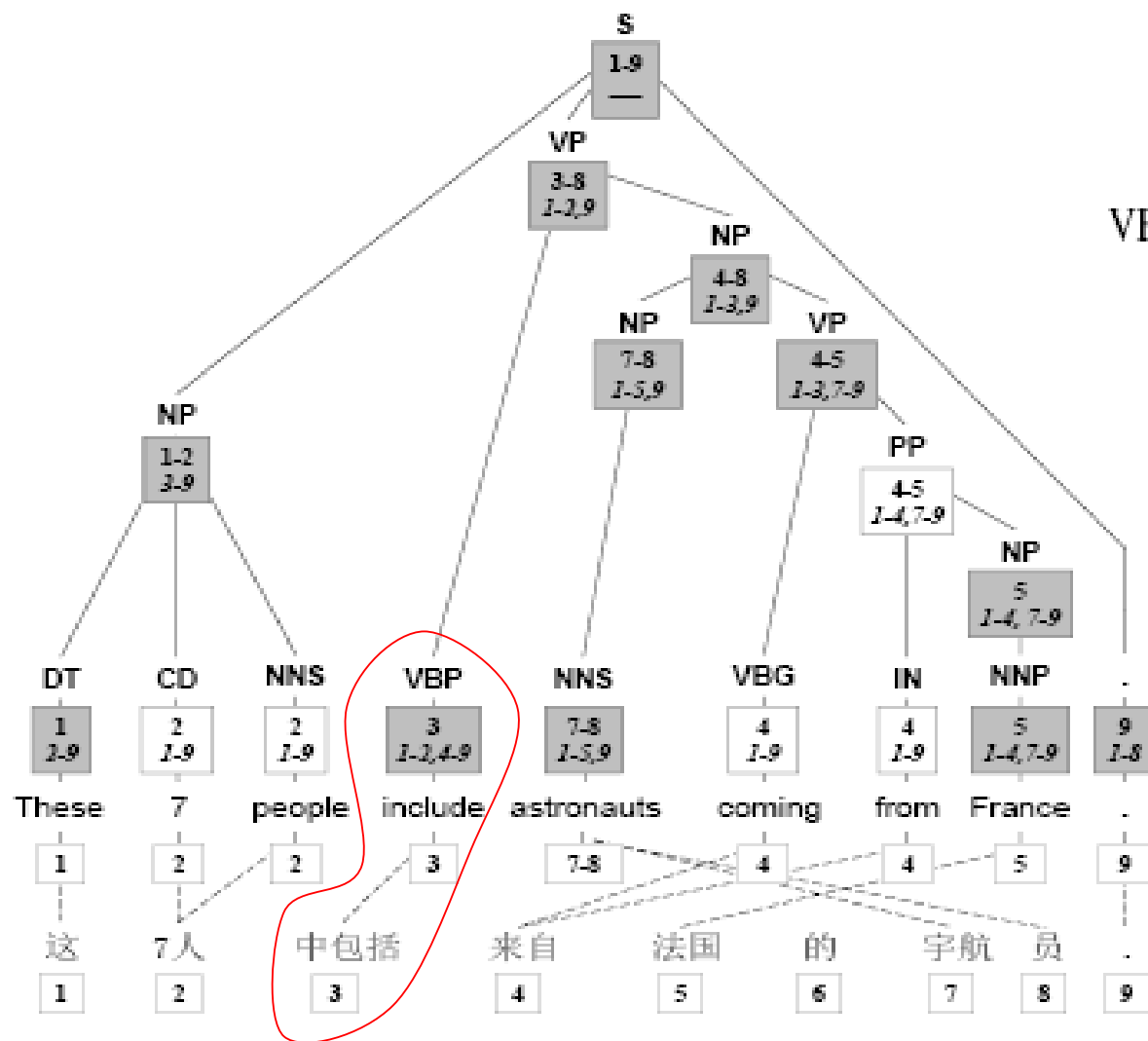


抽取规则 4



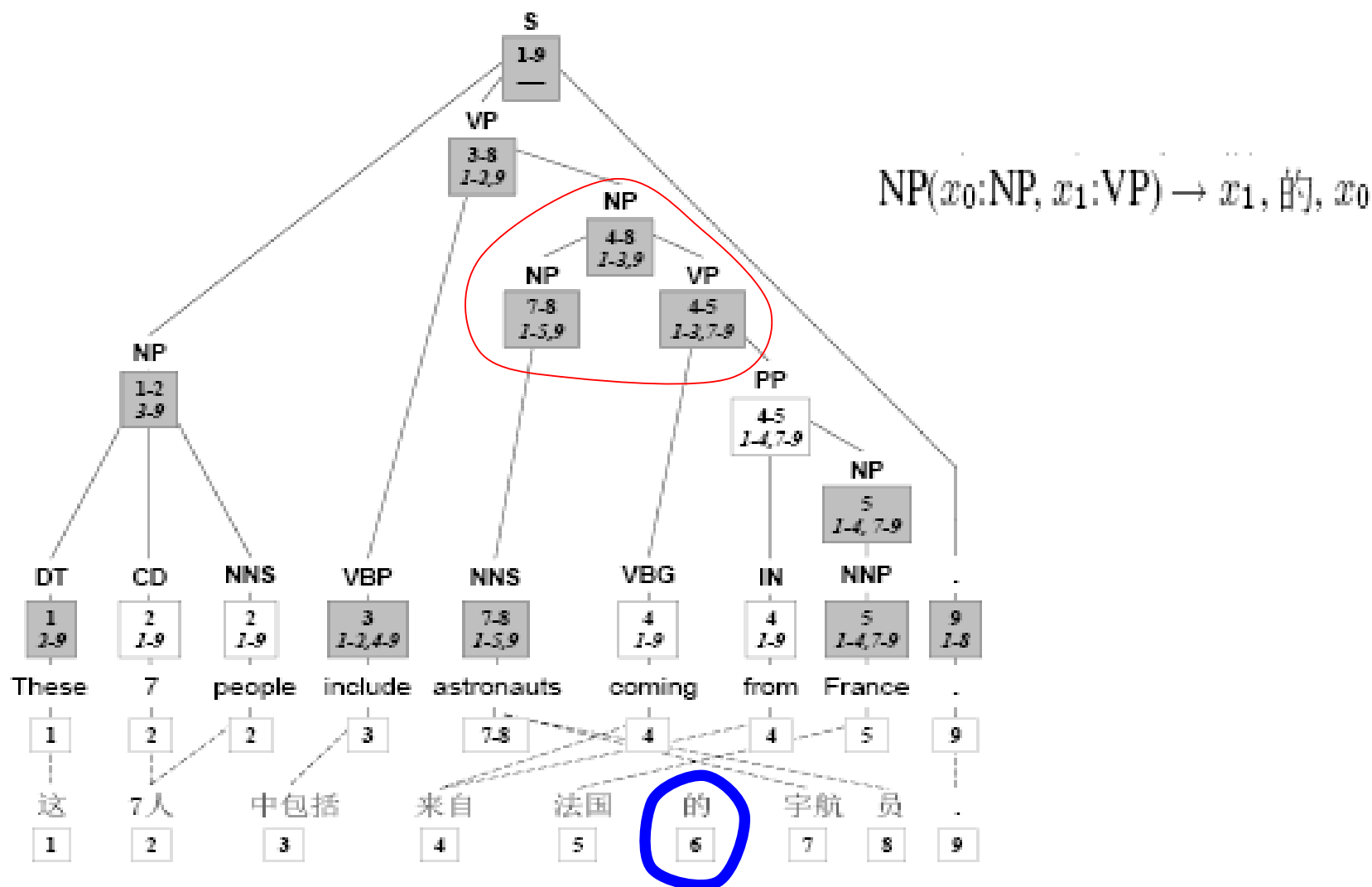
$VP(x_0:VBP, x_1:NP) \rightarrow x_0, x_1$

抽取规则 5

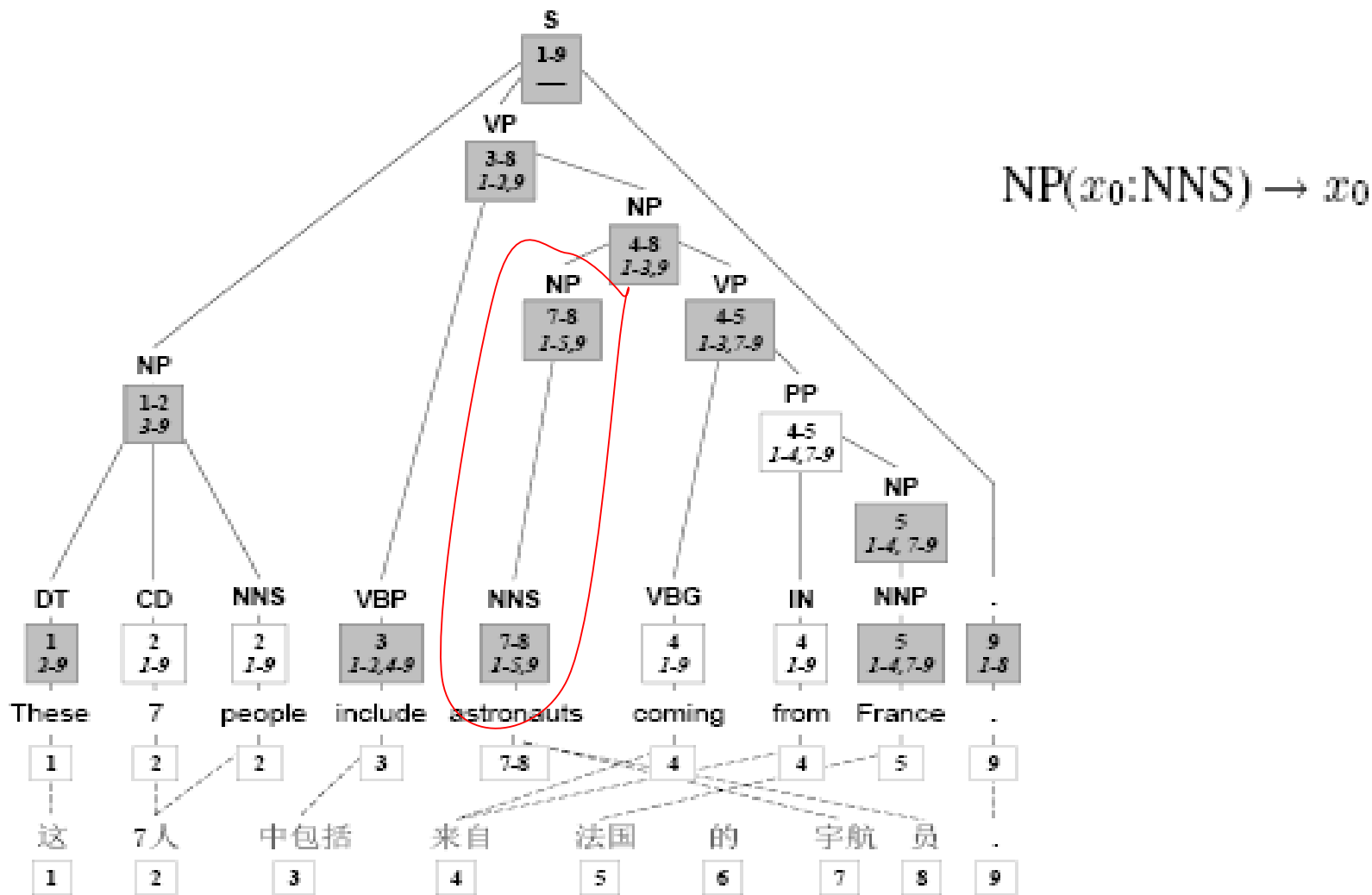


VBP(include) → 中包括

抽取规则 6

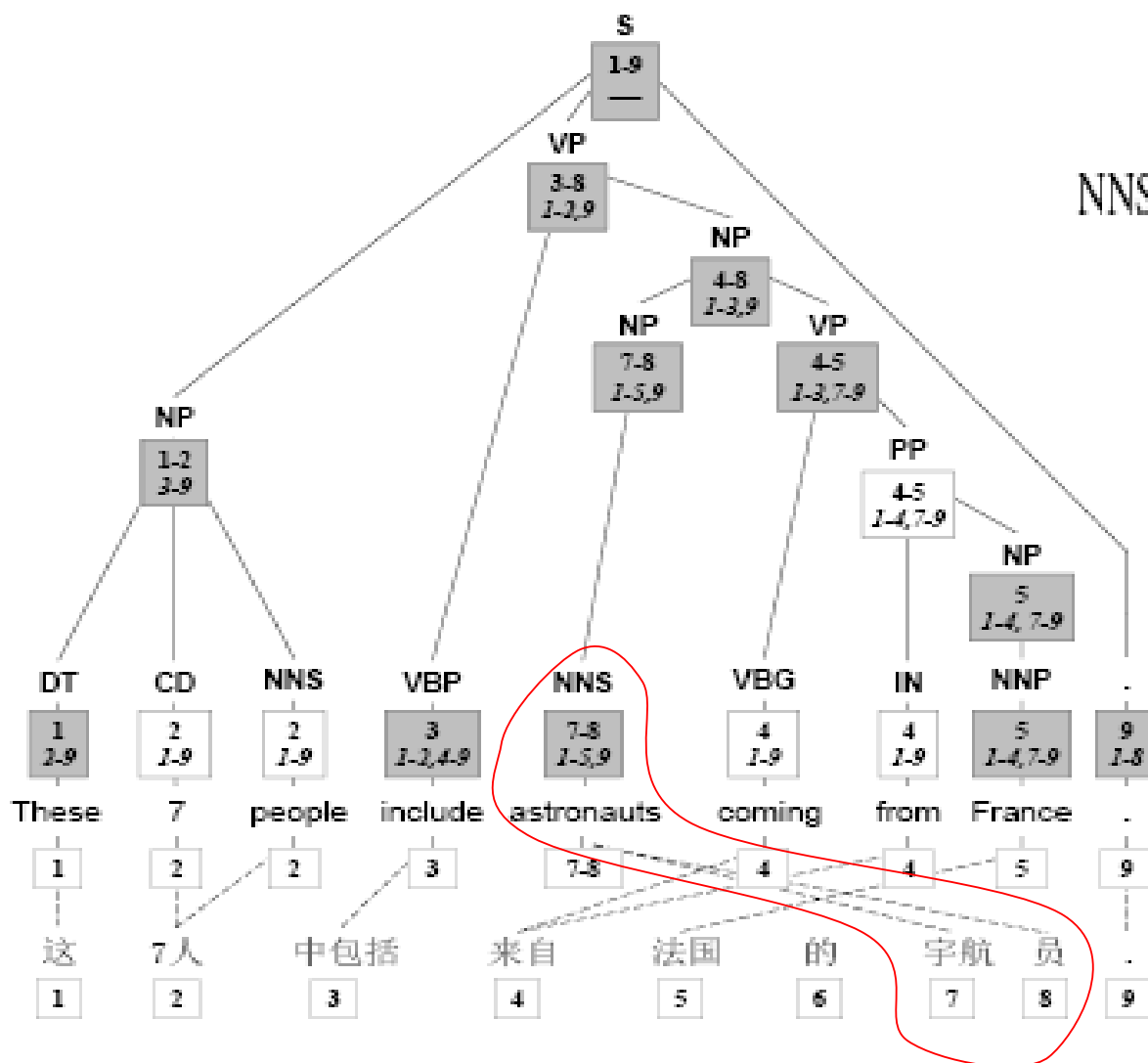


抽取规则 7

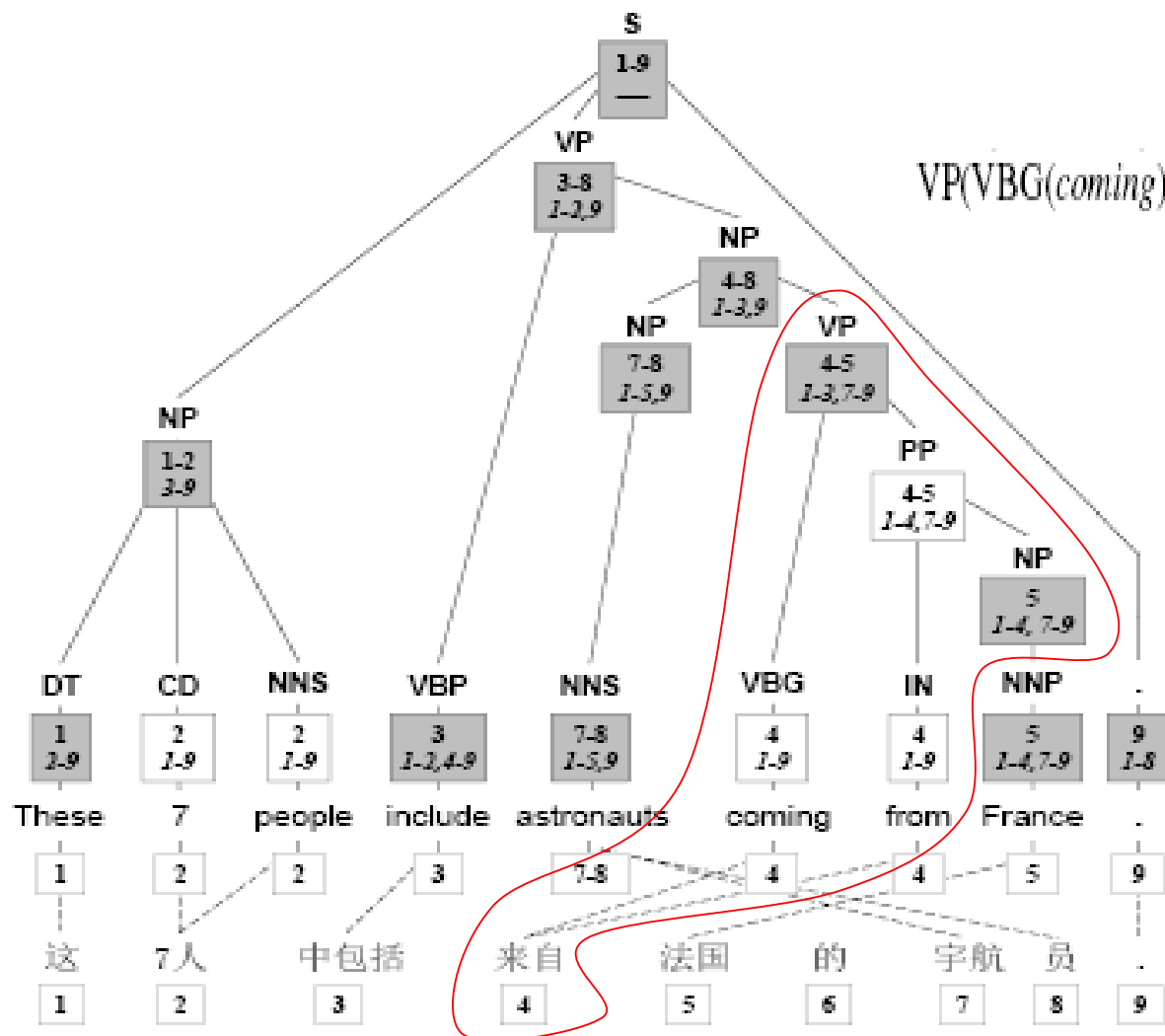


抽取规则 8

NNS(*astronauts*) → 宇航, 员

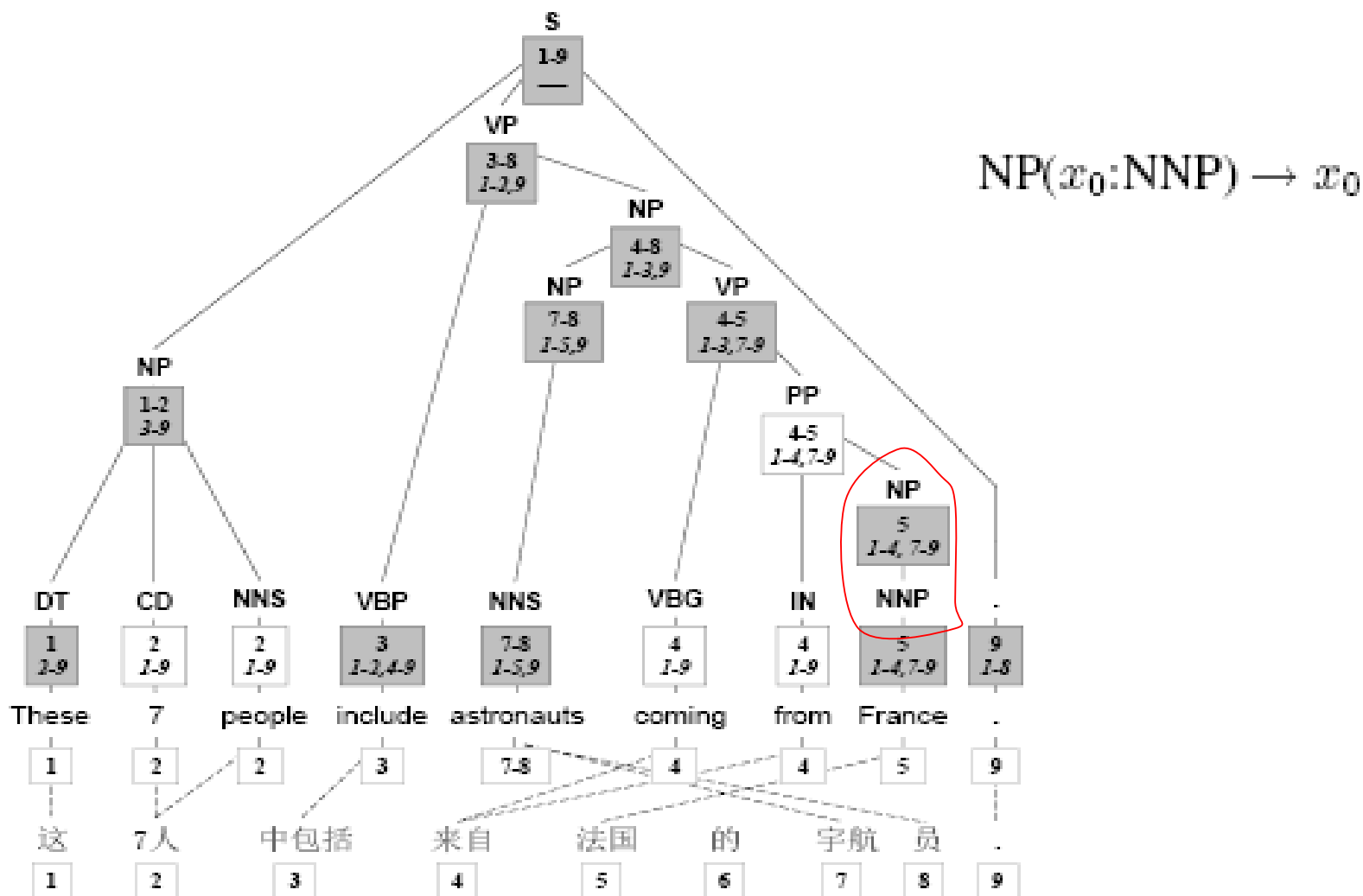


抽取规则 9

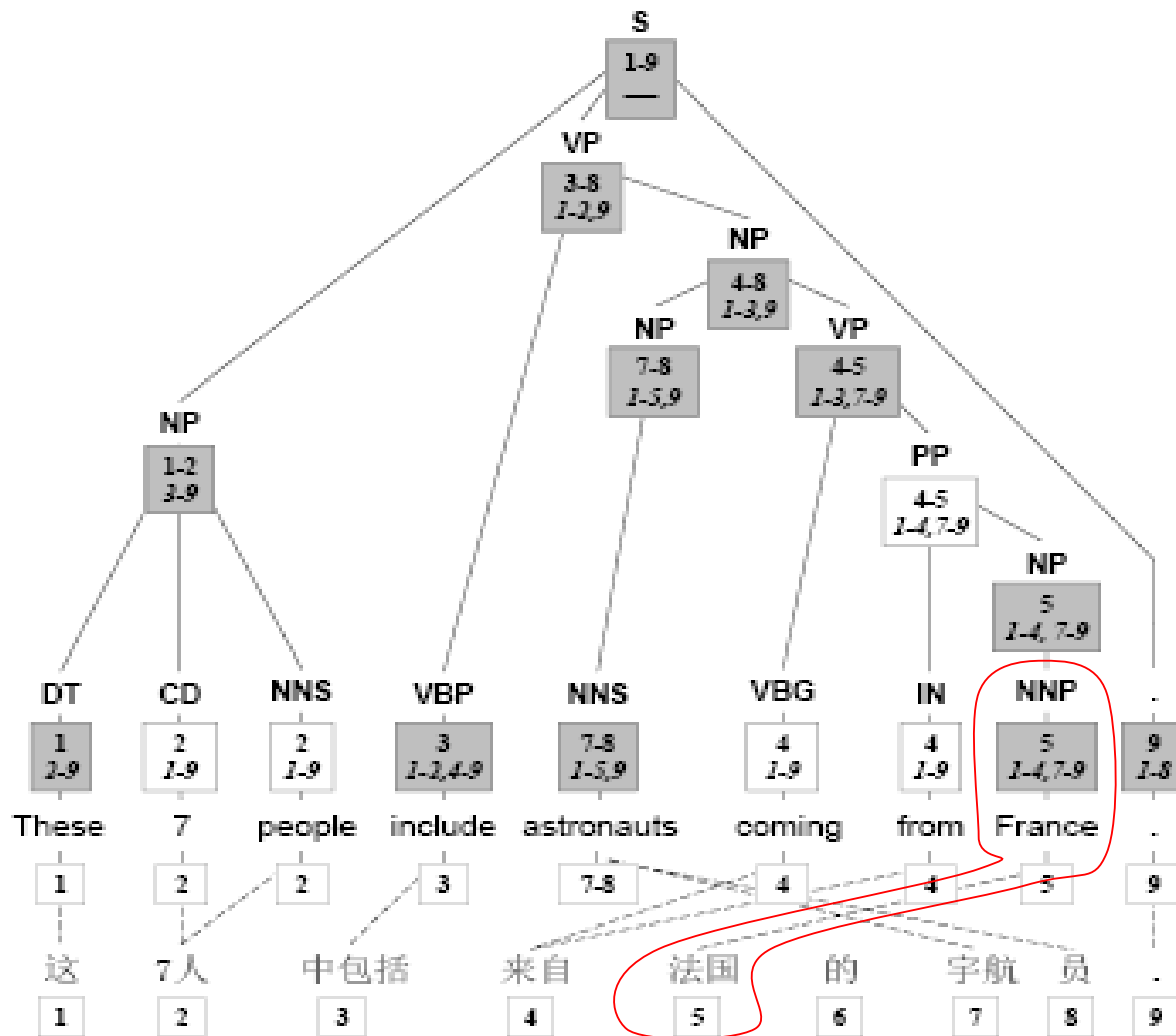


$VP(VBG(coming), PP(IN(from), x_0:NP)) \rightarrow \text{来自}, x_0$

抽取规则 10

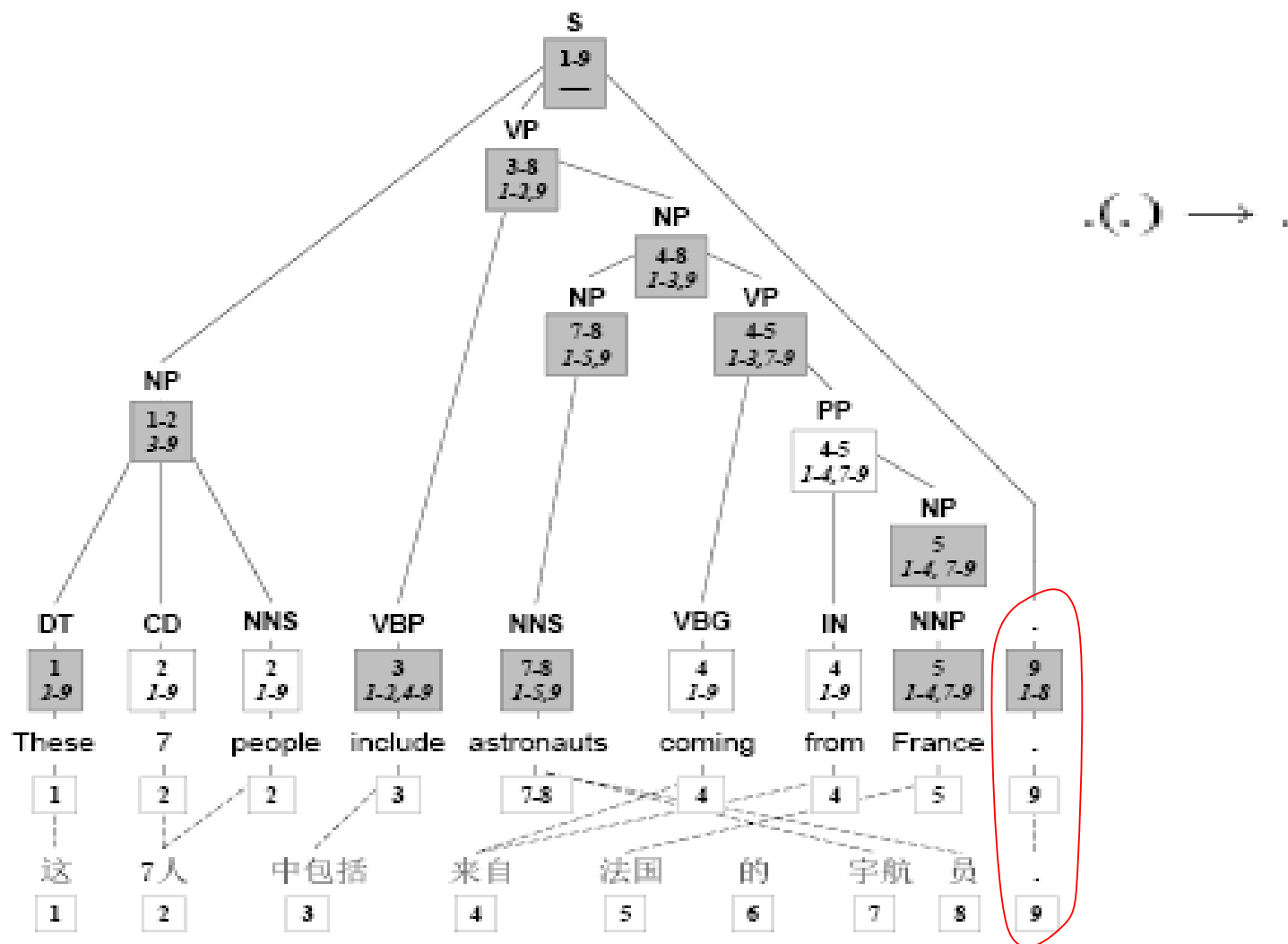


抽取规则 11



NNP(*France*) → 法国

抽取规则 12

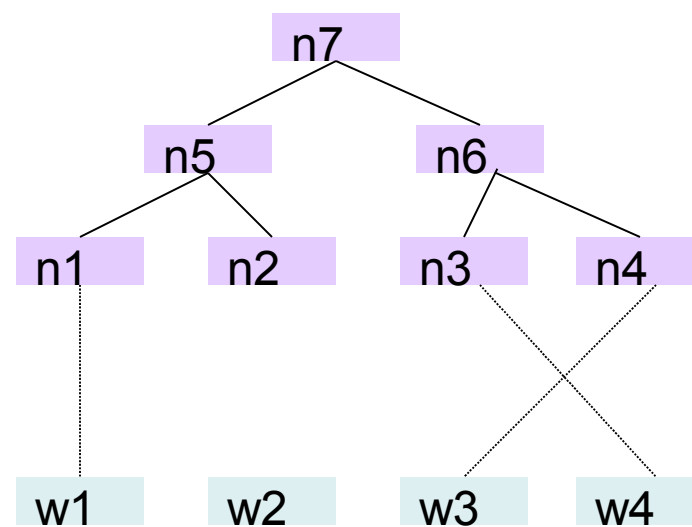


唯一最小推导

-
- (a) $S(x_0:NP, x_1:VP, x_2:.) \rightarrow x_0, x_1, x_2$
 - (b) $NP(x_0:DT, CD(7), NNS(people)) \rightarrow x_0, 7 \text{ 人}$
 - (c) $DT(these) \rightarrow \text{这}$
 - (d) $VP(x_0:VBP, x_1:NP) \rightarrow x_0, x_1$
 - (e) $VBP(include) \rightarrow \text{中包括}$
 - (f) $NP(x_0:NP, x_1:VP) \rightarrow x_1, \text{的}, x_0$
 - (g) $NP(x_0:NNS) \rightarrow x_0$
 - (h) $NNS(astronauts) \rightarrow \text{宇航, 员}$
 - (i) $VP(VBG(coming), PP(IN(from), x_0:NP)) \rightarrow \text{来自}, x_0$
 - (j) $NP(x_0:NNP) \rightarrow x_0$
 - (k) $NNP(France) \rightarrow \text{法国}$
 - (l) $.(.) \rightarrow .$
-

未对齐词导致的问题

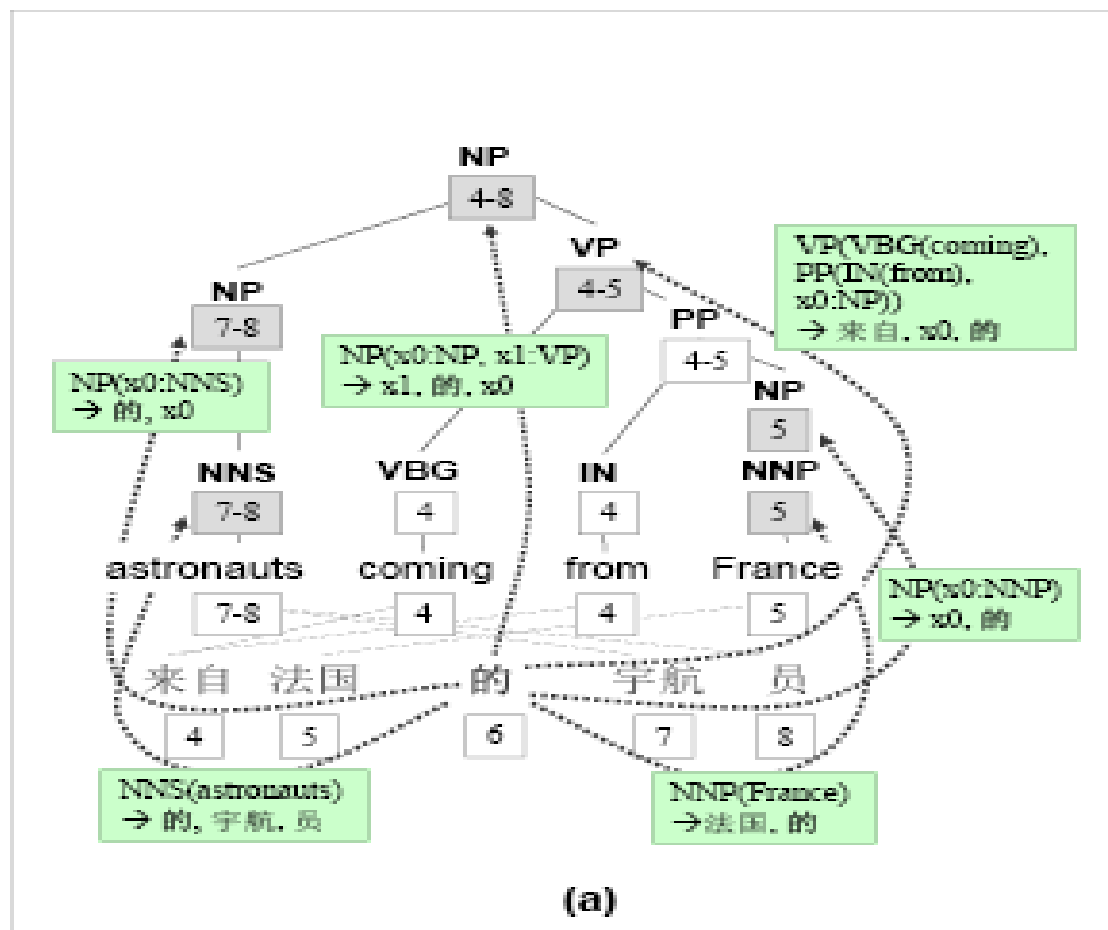
- 真实双语语料库中，源语言和目标语言中都有有一些未对齐的词，这些词会导致以下问题：
 - 未对齐的目标语言词使得其祖先结点区间无法确定
 - 未对齐的源语言词会导致抽取的规则数量组合爆炸



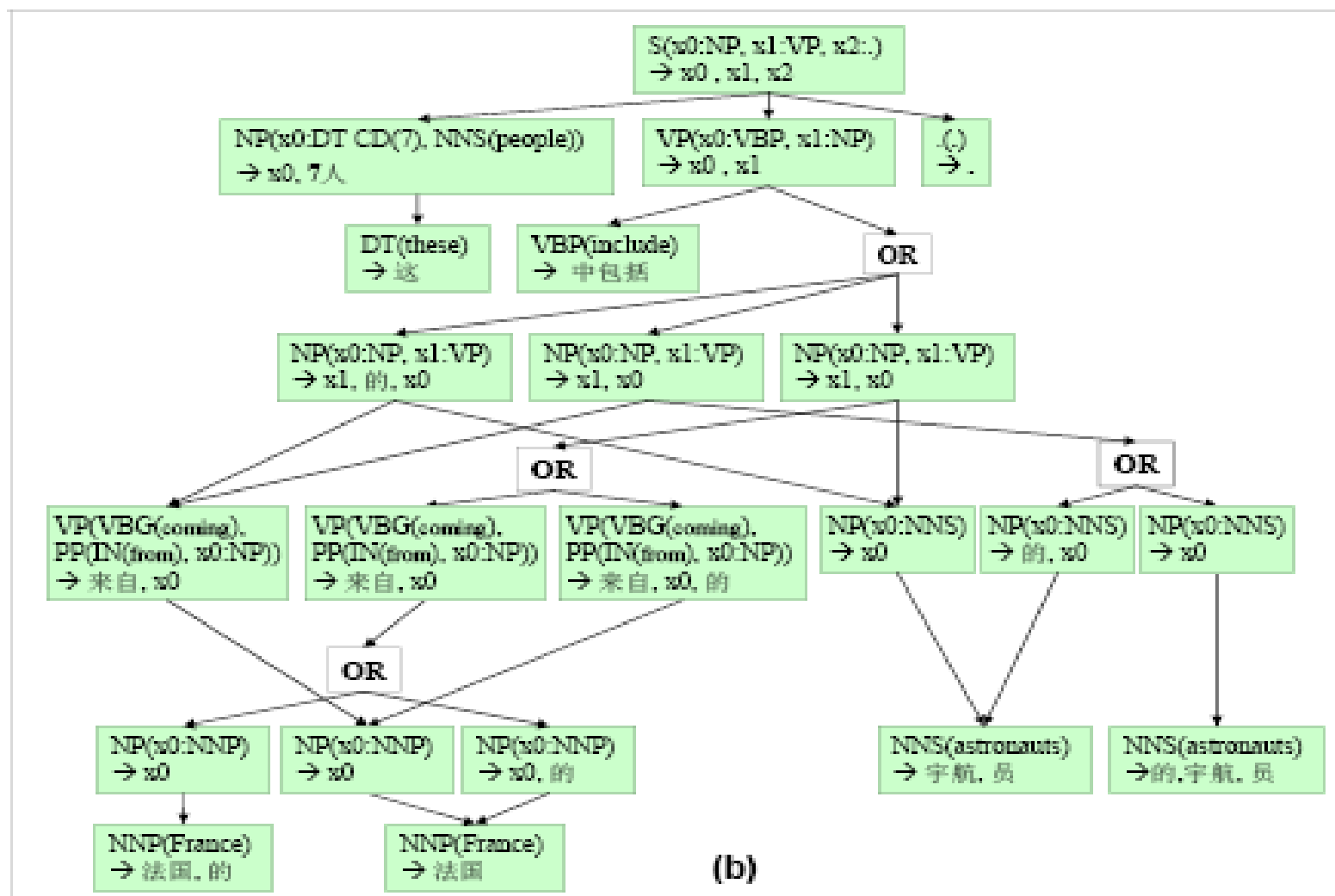
未对齐词的处理办法

- 单边附着
 - 将未对齐单词附着到覆盖它的区间最小的结点
- 多边附着
 - 不做任何将未对齐词“正确”附着的先验性假设，而是返回所有与图 G 相容的推导
 - 利用语料库的统计信息来优先选取与整个语料库一致性最好的未对齐附着方式

源语言未对齐词的多边附着



推导森林



新的规则抽取算法

- 与老算法类似，新算法也是自顶向下遍历图 G ，区别在于，对于每个结点 $n \in F$ 执行以下操作：

搜索所有以 n 为根结点的子树，找到对源语言未对齐单词进行附着的方式，并构造使用这些附着方式的图 G 的有效推导

- 比较

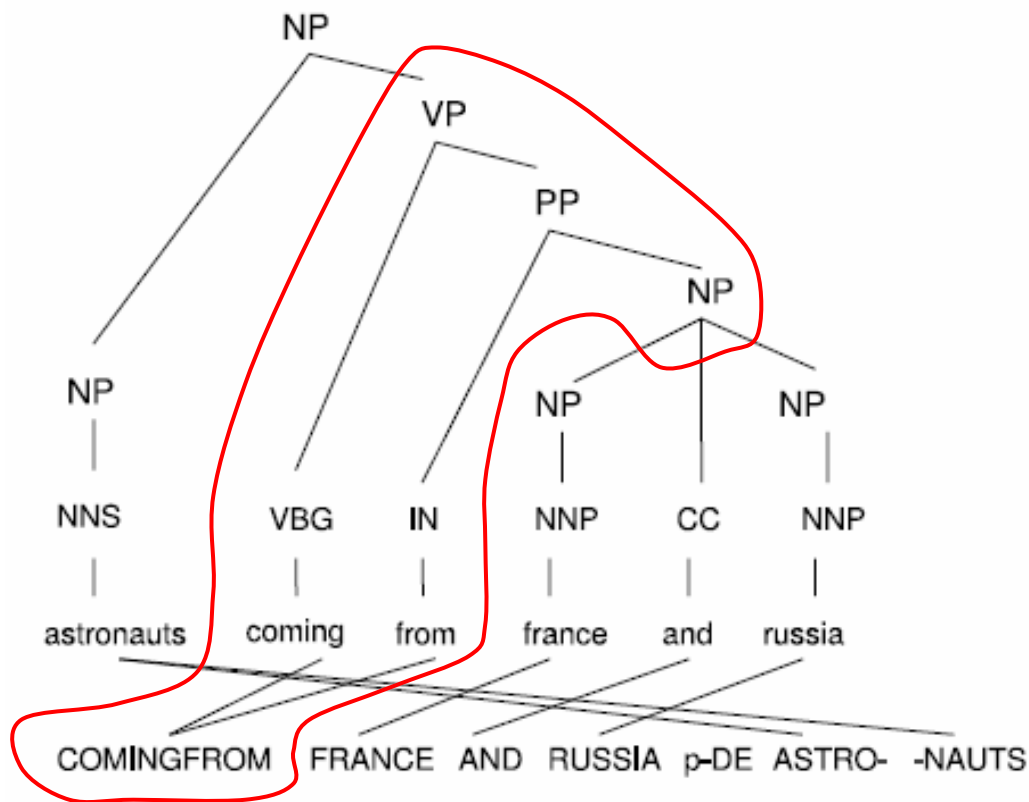
	老算法	新算法
规则	最小	最小、组合
附着方式	单边	多边

Marcu 的规则抽取算法

- 对于每一个源语言短语：
 - 首先抽取覆盖该短语的最小规则；
 - 从最小规则的根结点，往上找到第一个带有多子节点的父节点，从该父结点开始抽取一个包含该短语的组合规则

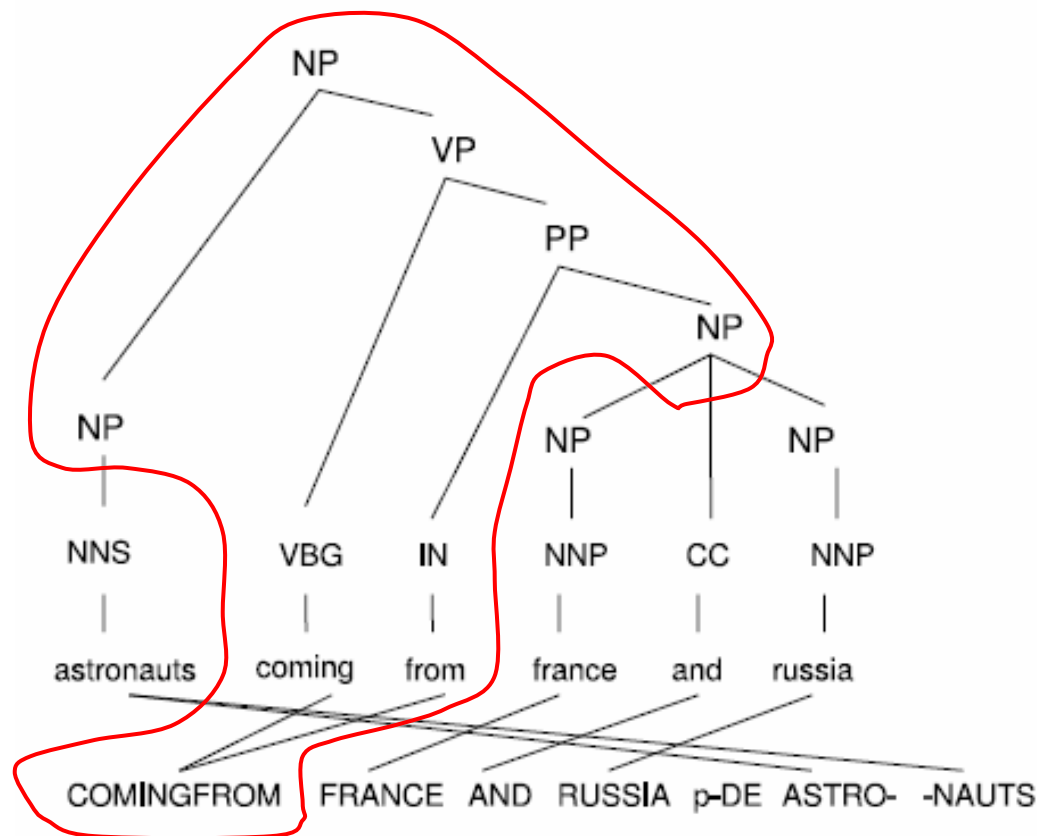
Marcu 的规则抽取算法

最小规则



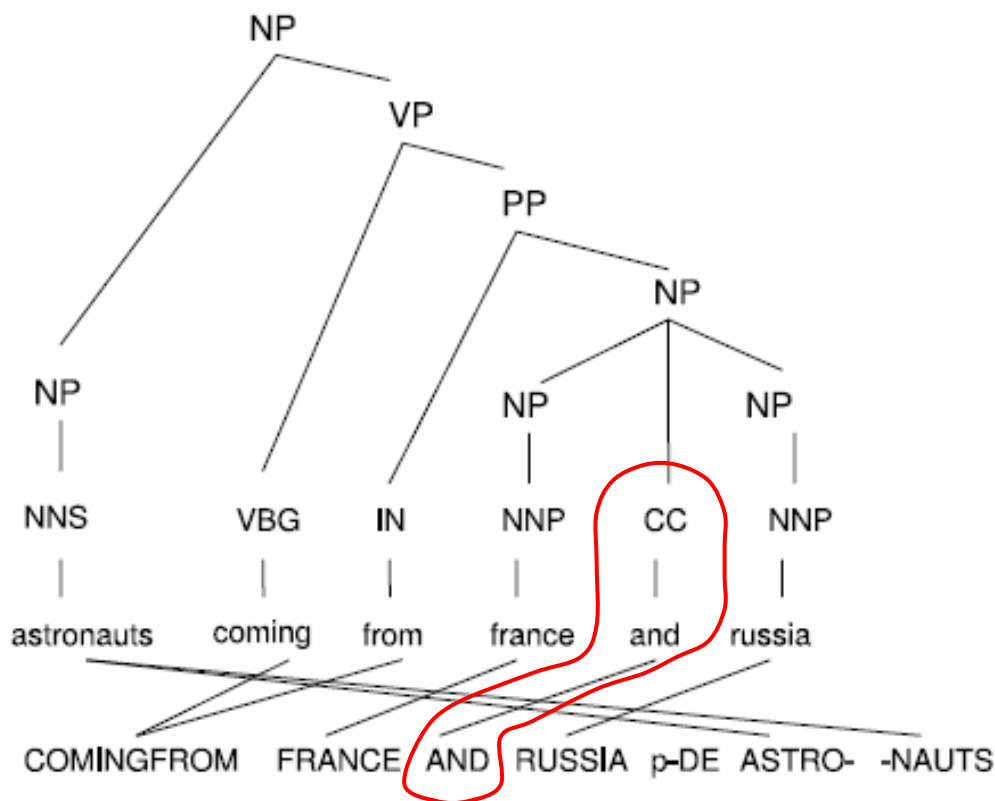
Marcu 的规则抽取算法

组合规则



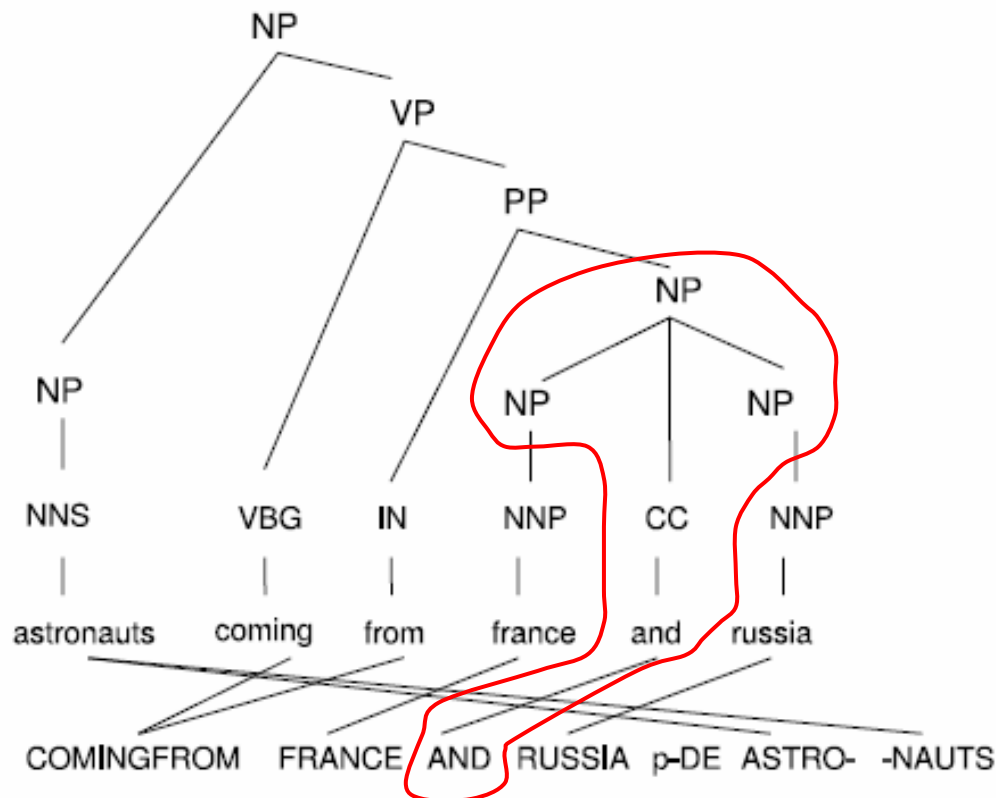
Marcu 的规则抽取算法

最小规则



Marcu 的规则抽取算法

组合规则



估计规则概率

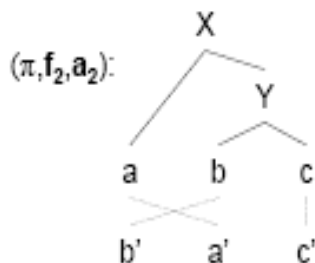
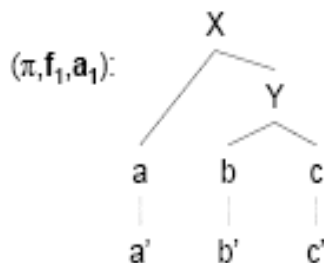
按规则左部完整树进行归一化：

$$p(rhs(r)|lhs(r)) = \frac{f(r)}{\sum_{r': lhs(r')=lhs(r)} f(r')}$$

按规则左部根结点进行归一化：

$$p(r|root(r)) = \frac{f(r)}{\sum_{r': root(r')=root(r)} f(r')}$$

Example



$$r_1: X(a, Y(b, c)) \rightarrow a', b', c'$$

$$r_2: X(a, Y(b, c)) \rightarrow b', a', c'$$

$$r_3: X(a, x_0:Y) \rightarrow a', x_0$$

$$r_4: Y(b, c) \rightarrow b', c'$$

假设左部的对齐出现了99次，右边的对齐仅出现了1次

按树归一化:

$$p_1 = 99/100 = 0.99$$

$$p_2 = 1/100 = 0.01$$

$$p_3 = 99/99 = 1.0$$

$$p_4 = 99/99 = 1.0$$

preferred by Liu

按根结点归一化:

$$p_1 = 99/199 = 0.4975$$

$$p_2 = 1/199 = 0.0050$$

$$p_3 = 99/199 = 0.4976$$

$$p_4 = 99/99 = 1.0$$

preferred by Galley

EM 训练算法

可以采用 **EM** 算法来训练上述模型，各规则的初始概率设置为均匀分布：

- 计算所有推导 θ_i 的概率，为推导过程所采用的所有规则概率的乘积；
- 对同一个句子的所有推导 θ_i 的概率进行归一化，使其概率之和为1；
- 对于每一条规则，对齐出现在的所有推导 θ_i 求和，作为该规则的新的概率 p_i
- 对 p_i 进行归一化

重复上述步骤，语料库的似然率将逐步提高。

高效的 EM 训练算法

- 由于不可能对大量的推导进行穷举，上述算法实际上是不可行的。
- Graehl and Knight (2004) 提出了一种高效的训练算法：
 - 对于每个训练实例构造一个推导森林；
 - 在该推导森林上运行 EM 算法，其复杂度是森林规模的多项式函数；

解码

- 采用自底向上的 CKY 形式的算法，在源语言句子的基础上，构造目标语言的短语结构句法树

Example

枪手

被

警方

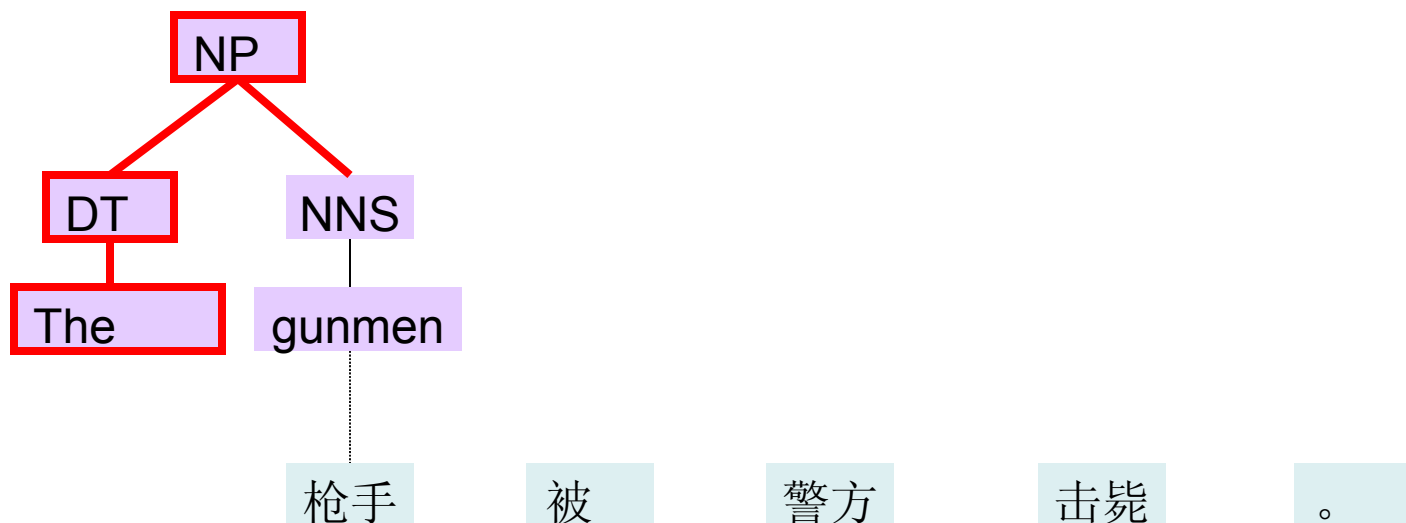
击毙

。

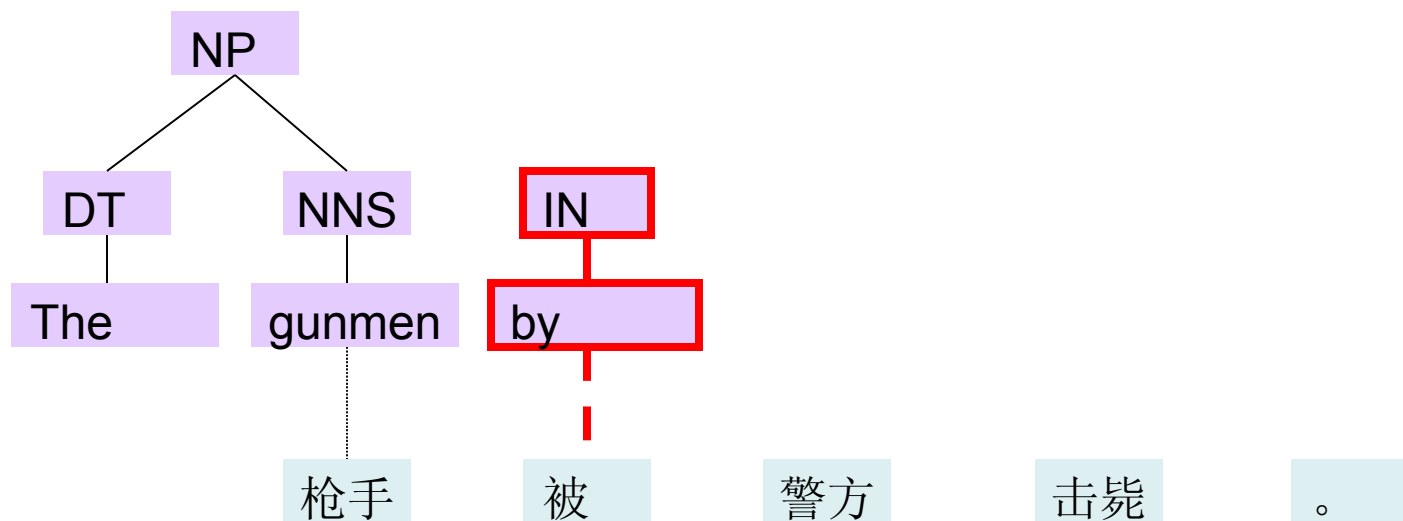
Example



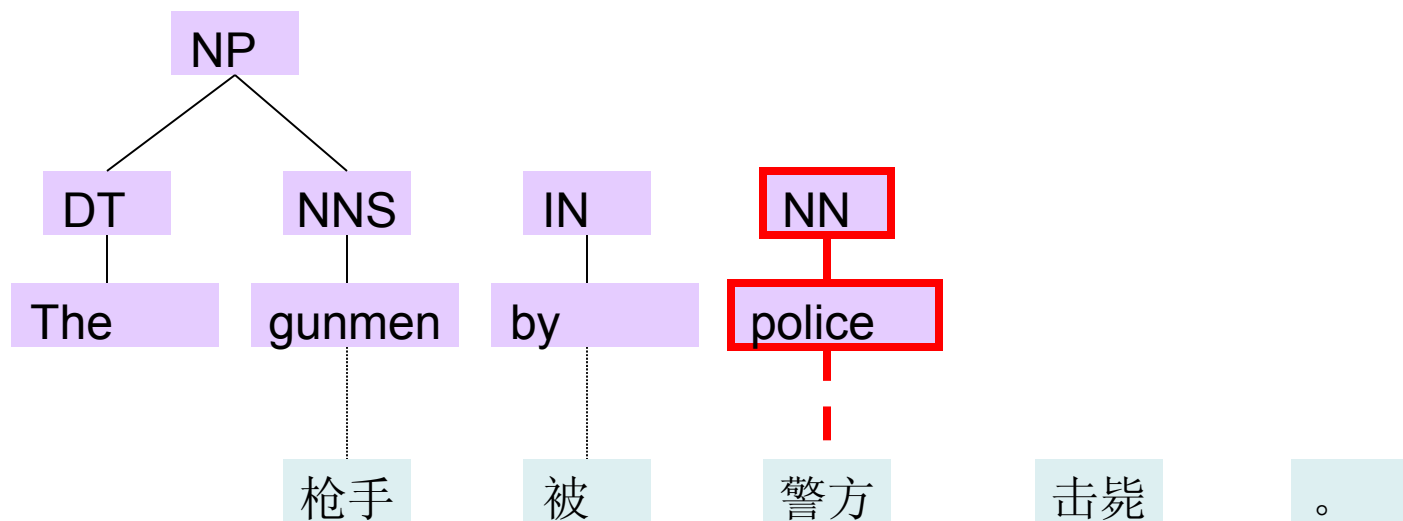
Example



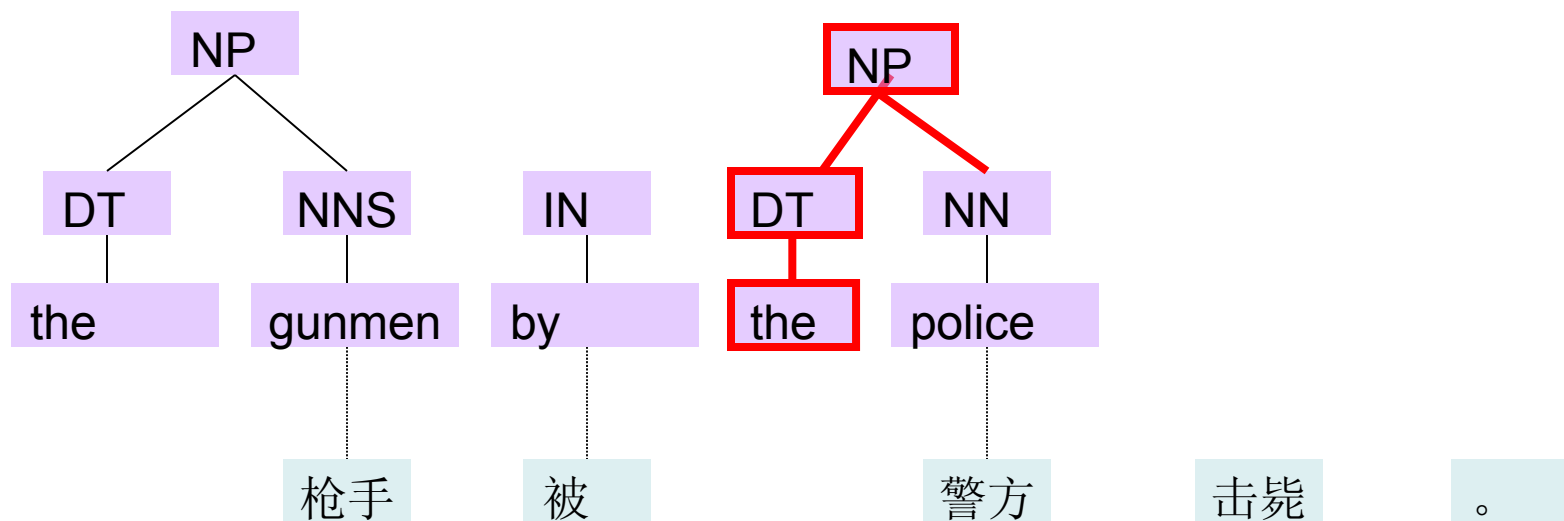
Example



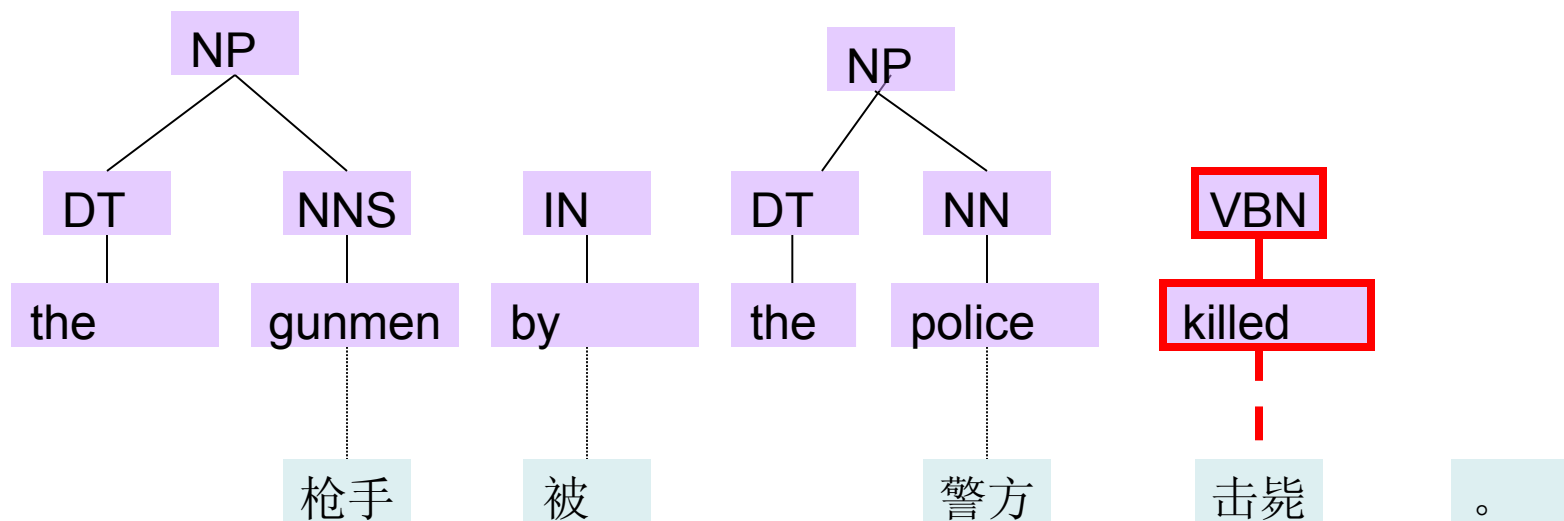
Example



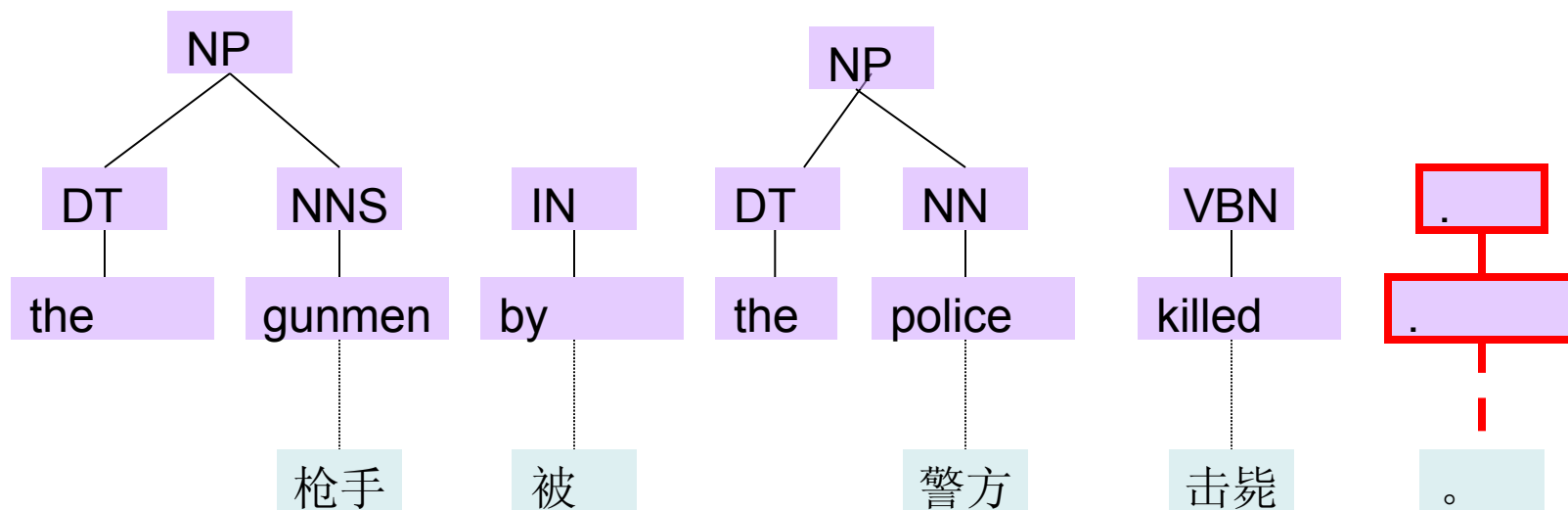
Example



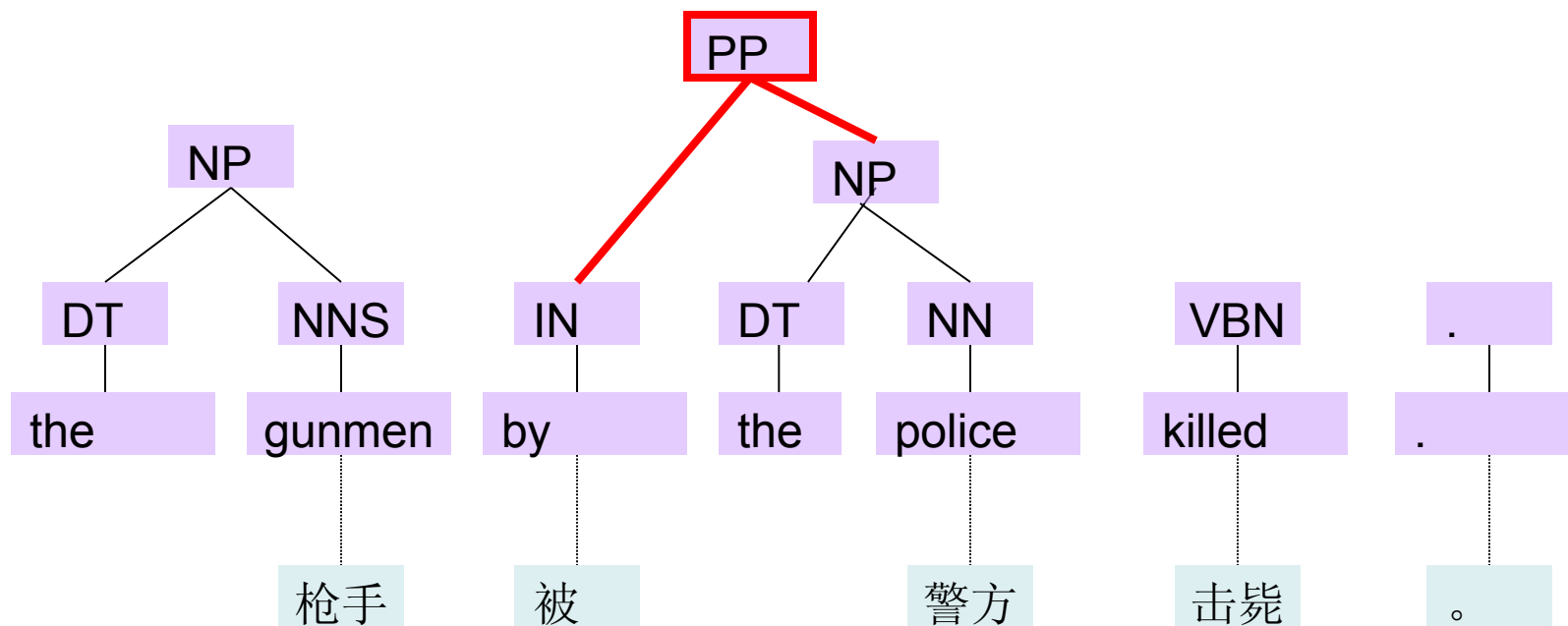
Example



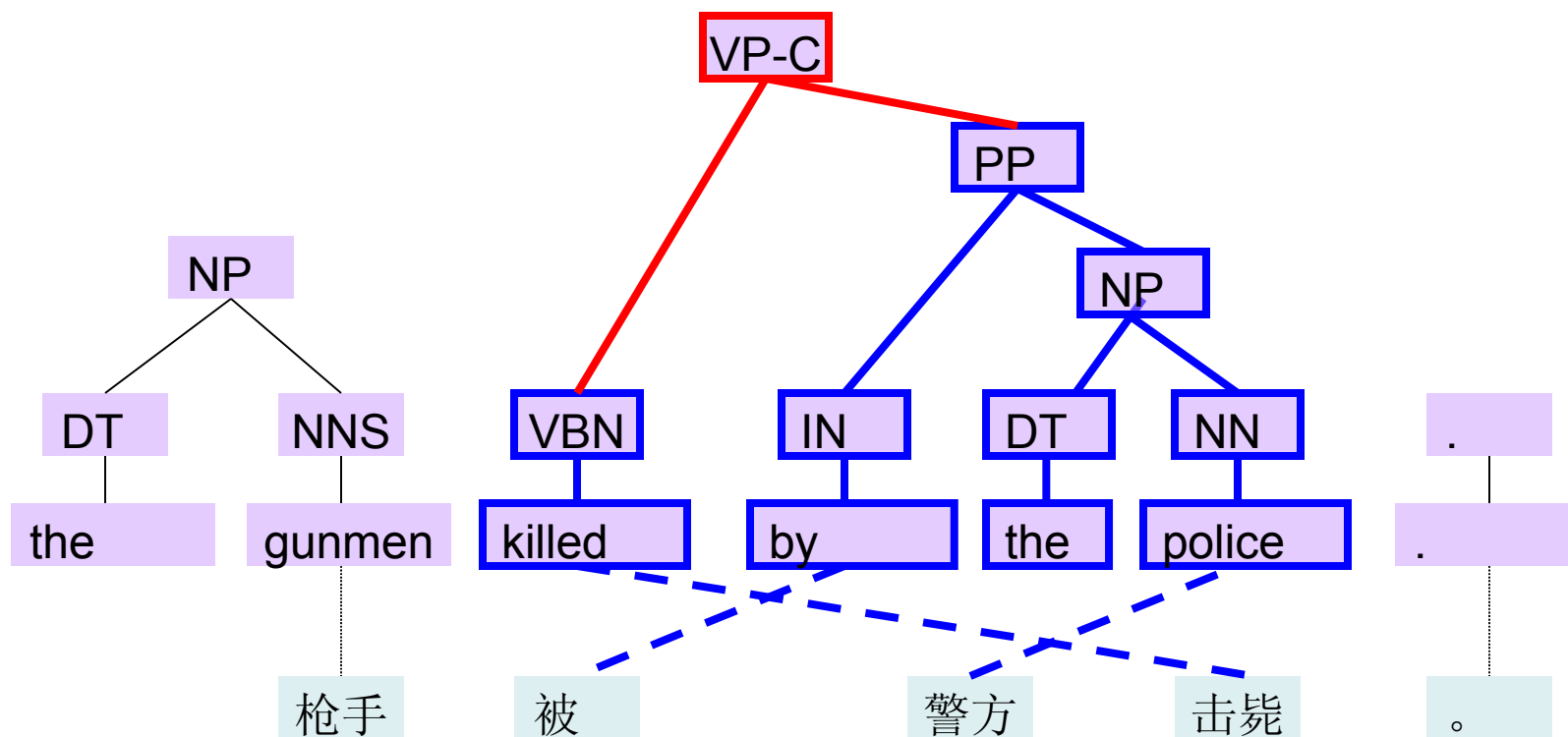
Example



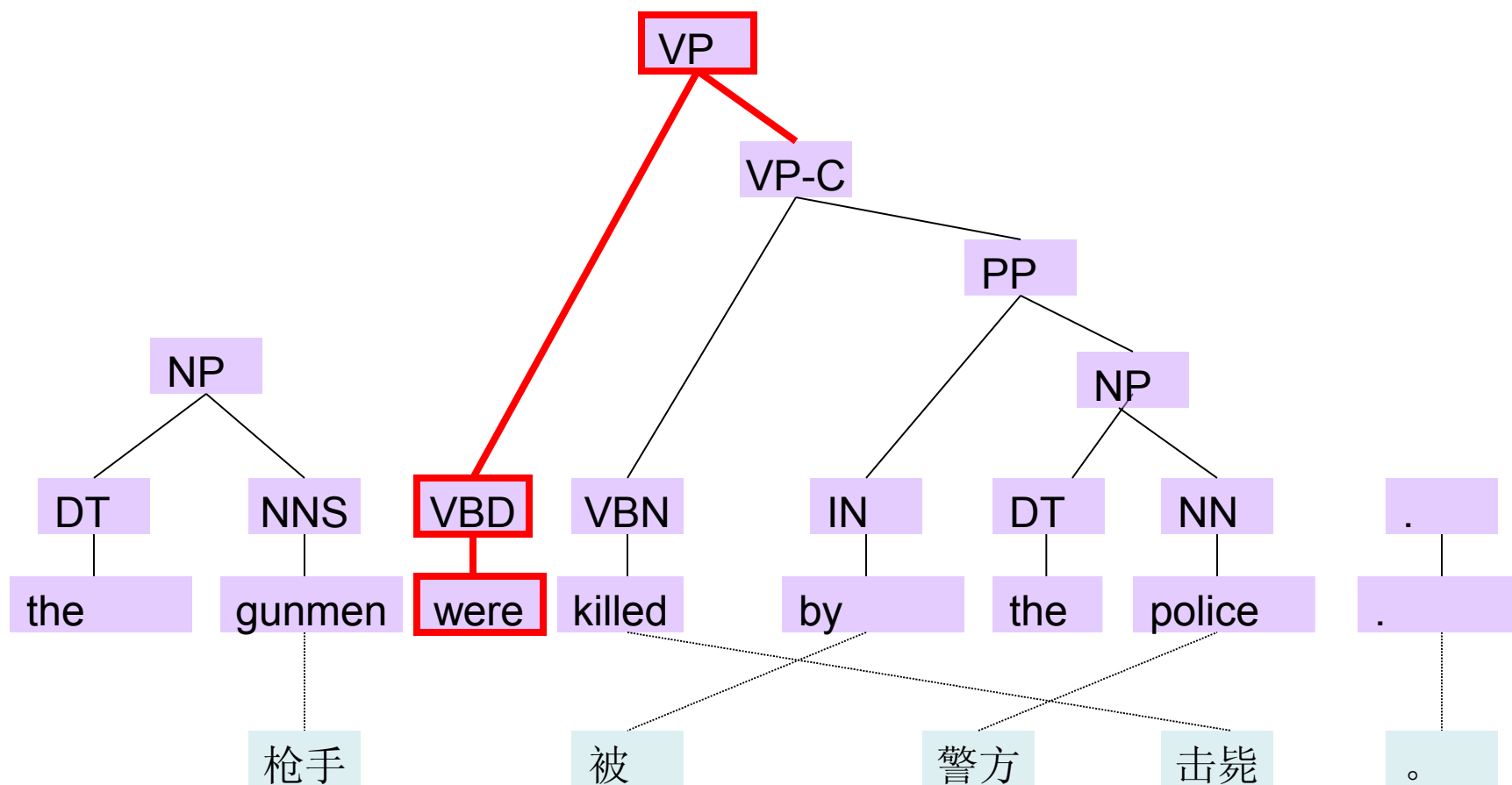
Example



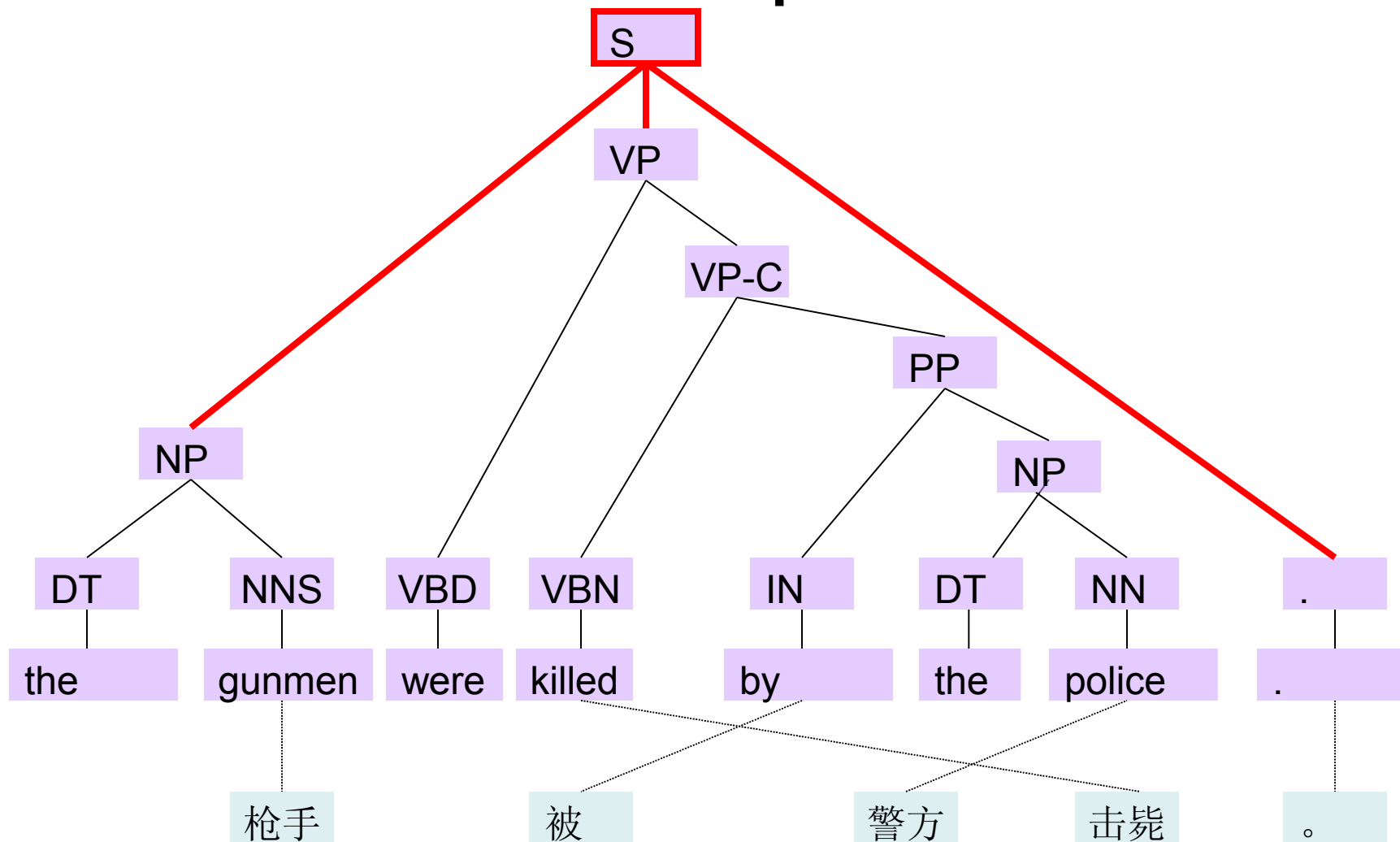
Example



Example



Example



规则的二义化（ Binarization）

- 现有规则不是二义的：规则右部可以有多个结点；
- 多义的规则导致解码器编程比较复杂；
- 解决办法：对规则进行二义化。通过增加非终结符将所有规则变成二义的规则。

规则的二义化（ Binarization）

- 不是所有规则都可以二义化
- 有多大比例的规则可以二义化？
 - 根据张浩的论文，在一个汉英双语语料库中提取的50,879,242条规则中，99.7%的规则是可以二义化的，而且剩下的0.3%的规则，根据人类专家的判断，绝大部分都是对齐错误导致的
- 规则二义化有什么作用？
 - 对翻译质量没有损失
 - 简化解码器的编程复杂度
 - 允许更有效的剪枝

线性的二义化算法

- 一条规则通常有很多种二义化的方法
- 张浩、黄亮等人提出了一种有效的二义化算法
 - 一种移进-规约算法
 - 只需扫描一遍，在线性时间内找到一种规则二义化等价形式

Galley 的实验： 不同规则集

rule set	nb. of rules	nb. of nodes	deriv-time	EM-time
C_m	4M	192M	2 h.	4 h.
C_3	142M	1255M	52 h.	34 h.
C_4	254M	2274M	134 h.	60 h.

Table 2: Rules and derivation nodes for a 54M-word, 1.95M sentence pair English-Chinese corpus, and time to build derivations (on 10 cluster nodes) and run 50 EM iterations.

C_m : 只抽取最小规则

C_3 : 抽取最小规则和组合规则，规则最多包含三个内部结点

C_4 : 抽取最小规则和组合规则，规则最多包含四个内部结点

Galley 所采用的特征

- 实验系统的解码器搜索过程中仅采用了一个经过 **EM** 训练的、**Root** 归一化的基于句法的翻译模型（**SBTM**）特征，甚至没有采用语言模型
- **Och** 的基于对齐模板的系统 **AlTemp** 作为对比系统，该系统采用了两个基于短语（**PB TM**）的翻译模型特征和12个其他特征

Marcu 的实验: SPMT Models

- SPMT Model 1 最小规则
- SPMT Model 1 Composed 组合规则
- SPMT Model 2 最小规则+兼容规则
- SPMT Model 2 Composed 组合规则+兼容规则

抽取的时候，限制源语言端短语的长度不超过四个词

组合多个 SPMT 模型的输出结果

- 每一个 SPMT 模型对于开发集中的所有句子都生成一个 nbest 列表，并给出每一个候选译文的所有特征值
- 将同一个句子的所有候选译文合并，根据其特征值重新进行最小错误率训练，得到一组新的特征参数
- 用这组特征参数，对于测试集上生成的 nbest 输出进行重新评分（rerank）

Marcu 采用的特征

- $proot(r_i)$: 对 root 归一化的概率
- $pcfg(r_i)$: 类 cfg 概率
- $is_lexicalized(r_i)$: 是否词汇化规则
- $is_composed(r_i)$: 是否组合规则
- $is_lowcount(r_i)$: 是否出现三次以下
- $lex_pof(r_i)$: 短语翻译概率
- $lex_pfe(r_i)$: 反向短语翻译概率
- $m1(r_i)$: IBM model 1
- $m1inv(r_i)$: 反向 IBM Model 1
- $lm(e)$: 语言模型
- $wp(e)$: 单词数惩罚

实验结果: Galley Vs. Och

Och 的对齐模板模型

	Syntactic	AlTemp
Arabic-to-English	40.2	46.6
Chinese-to-English	24.3	30.7

Table 5: BLEU-4 scores for the 2005 NIST test set.

	C_m	C_3	C_4
Chinese-to-English	24.47	27.42	28.1

Table 6: BLEU-4 scores for the 2002 NIST test set, with rules of increasing sizes.

实验结果： Marcu Vs. Och

System	# of rules (in millions)	Bleu score on Dev (4 refs) < 20 words	Bleu score on Test (4 refs) < 20 words	Bleu score on Test (4 refs)
PBMT	125.8	34.56	34.83	31.46
SPMT-M1	34.2	37.60	38.18	33.15
SPMT-M1C	75.7	37.30	38.10	32.39
SPMT-M2	70.4	37.77	38.74	33.39
SPMT-M2C	111.1	37.48	38.59	33.16
SPMT-Comb	111.1	39.44	39.56	34.10

Table 1: Automatic evaluation results.

实验结果： Marcu Vs. Och

System	Bleu score on Dev (3 refs) < 20 words	Judge 1	Judge 2	Judge 3	Judge avg
PBMT	31.00	3.00	3.34	2.95	3.10
SPMT-M1	33.79	3.28	3.49	3.04	3.27
SPMT-M1C	33.66	3.23	3.43	3.26	3.31
SPMT-M2	34.05	3.24	3.45	3.10	3.26
SPMT-M2C	33.42	3.24	3.48	3.13	3.28
SPMT-Combined	35.33	3.31	3.59	3.25	3.38
Human Ref	40.84	4.64	4.62	4.75	4.67

Table 2: Human-based evaluation results.

Analysis

- Galley
 - Good results with very poor feature functions
 - Very promising
 - It's reasonable to find that $C_4 > C_3 > C_m$ due to their difference in expressive power
- Marcu
 - Outperform Och's !
 - The results are really confusing. I suppose that:
 - $m_{2c} > m_2 > m_{1c} > m_1$
 - **Marcu suspect the decoder still makes many search errors**

内容提要

- 概述
- 同步语法概念
- 反向转录语法和括号转录语法
- 基于最大熵括号转录语法的翻译模型
- 同步上下文无关语法和同步树替换语法
- 层次短语模型
- 树到串翻译模型
- 串到树翻译模型
- 总结

总结

- 同步语法
- 形式化基于句法的翻译模型
 - 最大熵括号转录语法模型
 - 层次化短语模型
- 语言学基于句法的翻译模型
 - 树到串对齐模板模型
 - **ISI** 的串到树模型

Views on String-to-Tree Vs. Tree-to-String

- Galley
 - The target language (i.e. English) has syntactic resources (parsers and treebanks) that are considerably more available than for the source language
 - There is less benefit in modeling the syntax of the source language, since the input sentence is fixed during decoding and is generally already grammatical
- Liu
 - Source analysis may be important
 - Ill-formed source trees make the Tree-to-String decoder difficult to seek the “true” translation. We can never expect good syntactically-motivated reordering under ill-formed source tree structures. In contrast, the input of String-to-Tree decoding is source string. The decoder may build a reasonable target tree with the help of language model even though the rules are learned from ill-formed target trees in training data.
 - Tree-to-String decoding is useful for translating resource-rich languages (e.g. English) into resource-poor languages (e.g. Inuktitut)

ISI Vs. ICT

	ISI	ICT
model	string-to-tree	tree-to-string
phrasal compatibility	full	partial
features	11	7
extraction algorithm	top-down	bottom-up
unaligned words attachment	multiple	single
decoding algorithm	bottom-up CKY	bottom-up beam search
rule binarization	yes	no
nbest derivation generation	yes	no
nbest list generation	yes	no
treat BPs as special rules	no	yes
improve fluency using BPs	no	yes

Comparison

	Chiang	Galley	Marcu	Liu
model	formal syntax	string-to-tree	string-to-tree	tree-to-string
tree annotations	single-level	multi-level	multi-level	multi-level
syntactically-motivated	no	yes	yes	yes
phrasal compatibility	yes	no	yes	no
enable discontinuous source phrases	yes	yes	no	yes
feature functions	8	1	11	7

讨论